PI-Grau
(Internet Protocols)

José M. Barceló Ordinas
Departamento de Arquitectura de Computadores (UPC)

- **Topic 2: Corporate Networks: Switching Blocks**
  - Objectives
    - Introduce basic **switching** concepts
    - Understand **Corporate Network design** principles
    - Understand **L2/L3 reliability** concepts and protocols
    - Learn **CPD (Data Processing Center)** design techniques

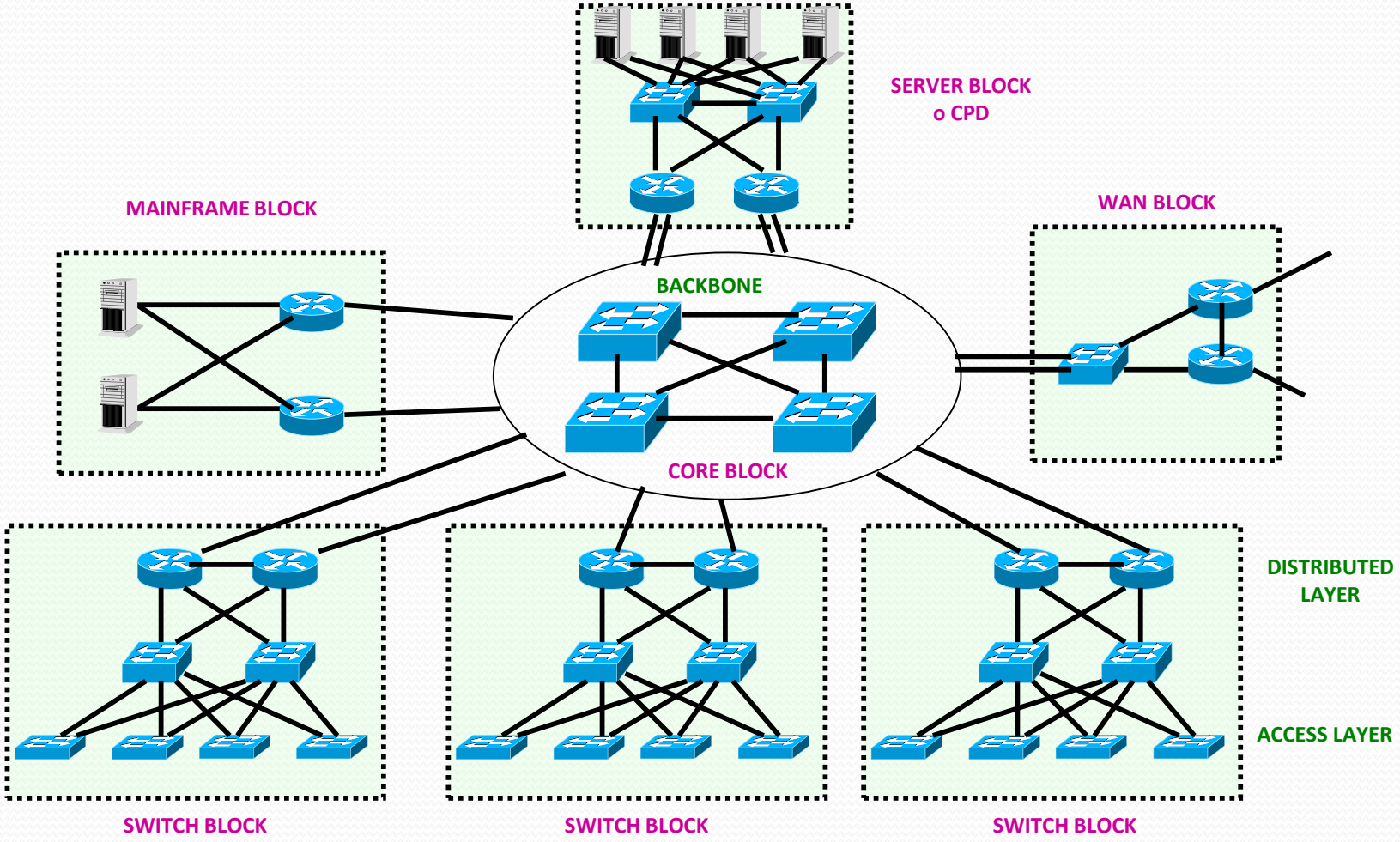
Corporative Networks (IP-Net-Clients)

- Companies with ***end users*** and ***end services***
- As any end user, they are connected to other end users and other corporative networks via an ISP
- A corporative network can be something ranging between:
  - Small company with few users, to a large company with thousands of users
- A corporative network may:
  - Manage their services in a CPD (Centre Processing Data) located in the Main Site
  - Manage their services via others (e.g. either another corporative network or an ISP) that provides the service (e.g. hosting, housing, virtualization, …)

- **Switching Block Architectures:**
  - **Corporate Network Design:** use switching blocks interconnected by a fast switching backbone
    - **Switching blocks:** users connect to **access switches**. These ones are connected to **aggregation switches** that aggregate user traffic to the routers that get them out of the switching block.
    - **Data Processing Center (CPD):** specific switching block in which the access switches give service to **servers** instead of end users.
    - **Backbone block:** group of **core switches** that interconnect switching blocks
    - **WAN block:** block that give **access to Internet** and to VPN connectivity.

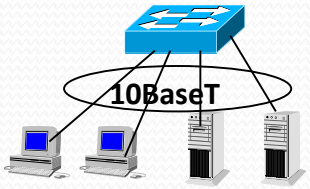
Topic 2: Corporate Networks: Switching Blocks

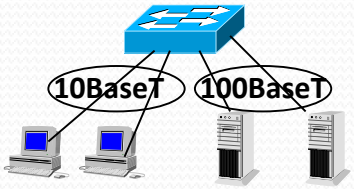
- **Switching Block Architectures:**

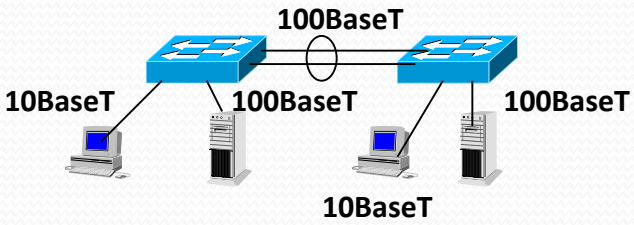


- **Ethernet Technologies:**
  - Symmetric ports: all with the same rate
  - Asymmetric ports: different rates
  - **Half Duplex** ports (typically 10BaseT) and **Full Duplex** (typically Fast Eth. and GigaBit Eth.)
    - FD ports deactivates the collision CSMA/CD mechanism
  - **MAC Tables:** static/dynamic entries. The MAC table allows switching frames from one port to other as a function of @MAC-dst

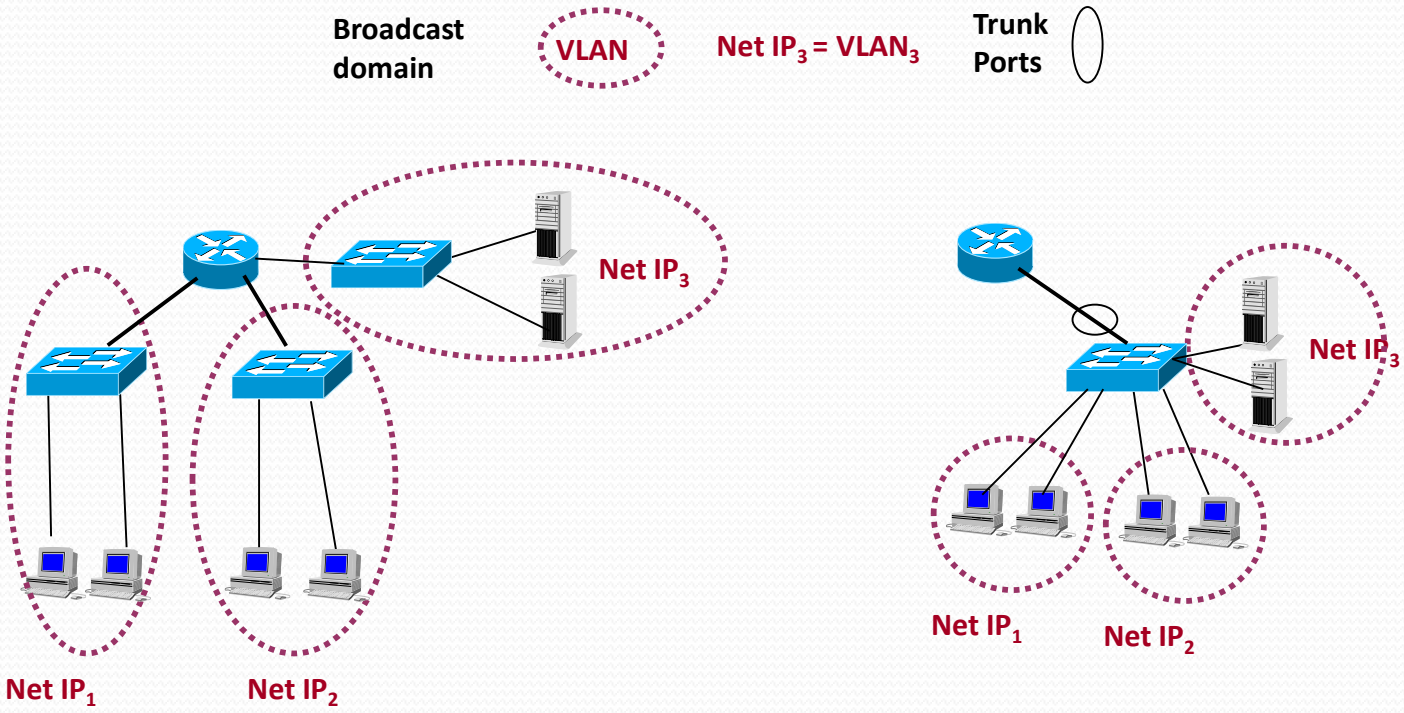
PORT (IFACE)	MAC	AGE
Eth0	01:01:01:ab:fe:f2	10'
Eth0	01:01:01:cb:5e:27	10'
Eth1	01:01:01:bb:a4:31	10'



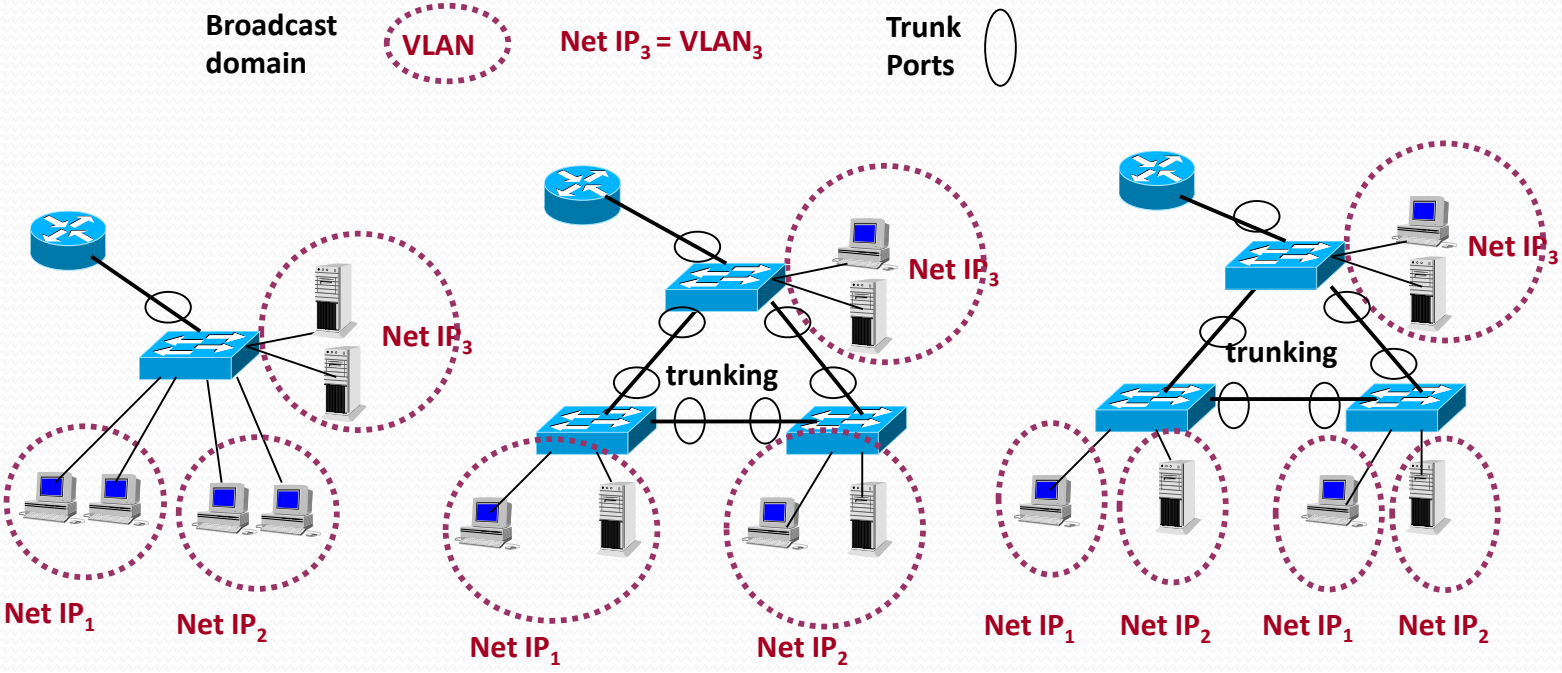




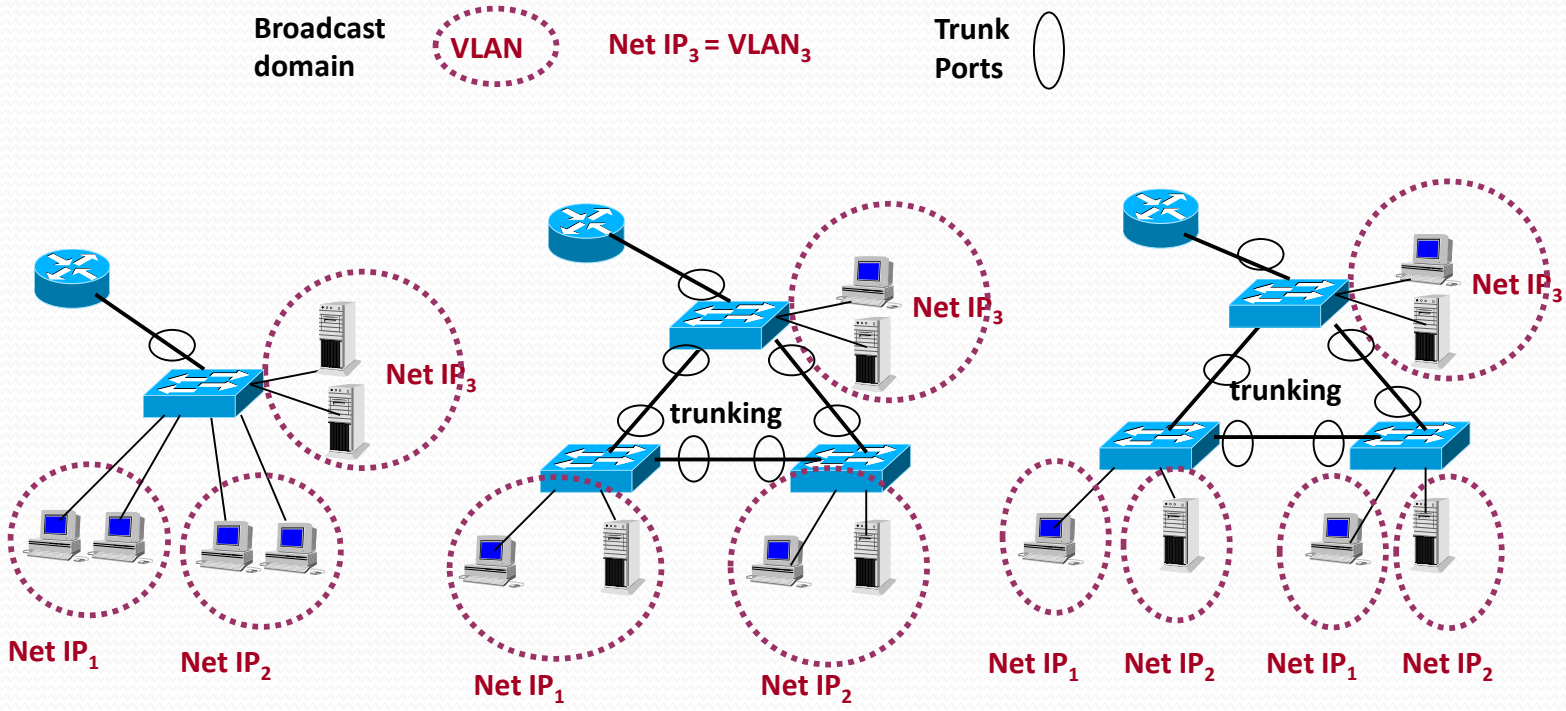
- **VLANs**
  - Split broadcast domains (IP networks) in a switch instead than do it in the router,
  - Saves router ports using dedicated software in the router and the switch,
  - Each VLAN is a broadcast domain (IP network) and as we can see in the following slide can be deployed across several switches.



- **VLANs**
  - Split broadcast domains in a switch instead than do it in the router (saves router ports) using dedicates software
  - **Trunking:** links that support multiple VLANs using "tagging" techniques: 802.1Q, CISCO ISL (Inter-switch Link)



- **Static VLANs**
  - Group of ports belonging to a switch/switches that are assigned to the same broadcast domain (IP network)



- **VLANs: CISCO IOS (Sw 2950)**

Setting a static VLAN

!!! Setting vlan in the database with name vlan2

Sw# **vlan database**

Sw(vlan)# **vlan** 2 **name** vlan2

!!!! Assigning the port Ge0 to vlan 2

Sw(conf)# **interface** Ge0

Sw(config-if)# **switchport mode access**

Sw(config-if)# **switchport access vlan** vlan2

!!! Port Ge1 activated as port trunk

Sw(conf)# **interface** Ge1

Sw(config-if)# **switchport mode trunk**

Security in Switches: port security features

The **port security** functionality provides the ability to limit what addresses will be allowed to send traffic on individual switchports through the switch.

!!! Setting port security
Sw(conf)# **interface** Ge0
Sw(config-if)# **switchport mode access**
Sw(config-if)# **switchport port-security maximum** *value*
Sw(config-if)# **switchport port-security violation** *{protect | restrict | shutdown}*

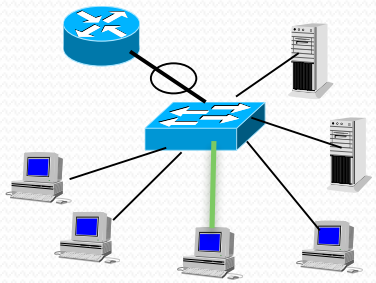
Sw# **show port-security interface** Ge0

In this case, "value" indicates the maximum number of MAC@ allowed in the port the switch allows a single MAC@ to be learnt (dynamically) and the action taken. Default is value=1 and shutdown using the single command switchport port-security (without specifying value and violation action).

Protect: only allows traffic from permitted MAC@ but does not notify if there is a violation of MAC;

Restrict: same that protect but sends an SNMP (simple network management protocol) to the admin;

Shutdown: disable the port if violation is seen. It has to be activated (no shutdown) manually by the admin



5D:FF:68:DE:22:0A ALLOWED
Other MAC@ not allowed in this port

Security in Switches: port security features

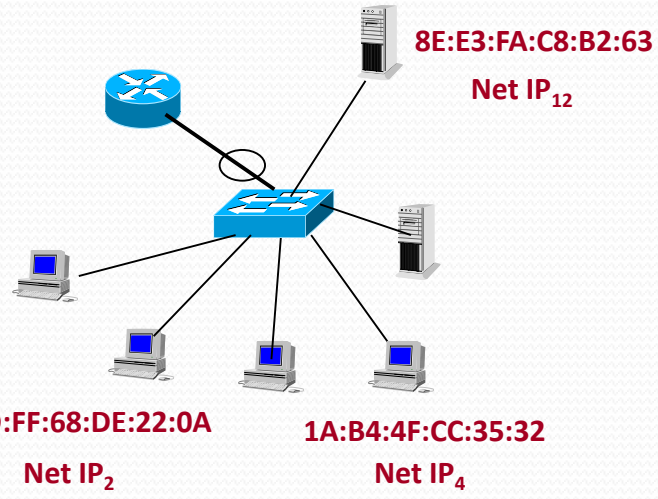
It is possible to specify specific MAC@ using the MAC@'s or the **sticky** figure

!!! Setting port security
Sw(conf)# **interface** Ge0
Sw(config-if)# **switchport mode access**
Sw(config-if)# **switchport port-security maximum** *1*
Sw(config-if)# **switchport port-security violation** shutdown
Sw(config-if)# **switchport port-security mac-address** 5D:FF:68:DE:22:0A

!!! Setting port security
Sw(conf)# **interface** Ge0
Sw(config-if)# **switchport mode access**
Sw(config-if)# **switchport port-security maximum** *1*
Sw(config-if)# **switchport port-security violation** shutdown
Sw(config-if)# **switchport port-security mac-address** sticky

A **sticky MAC address** is a hybrid between a static and dynamic MAC address. When it is dynamically learned, the MAC address is automatically entered into the running configuration as a static MAC address; the address is then kept in the running configuration until a reboot.

- **Dynamic VLANs**
  - a switch automatically assigns the port to a **VLAN** using the MAC address or IP address. When a device is connected to a switch port the switch queries a database to establish **VLAN** membership,
  - **VMPS (VLAN Member Policy Server)** provides a centralized server for selecting the VLAN for a port dynamically based on the MAC device address.



A VMPS Server Essentially Maps VLANS to MAC's

Entry	VLAN Membership	MAC Address
1	2	5D:FF:68:DE:22:0A
2	4	5A:09:DF:FF:41:12
3	4	1A:B4:4F:CC:35:32
4	12	8E:E3:FA:C8:B2:63
5	4	F2:3D:A9:00:37:42
6	4	C4:72:36:FF:A2:61
7	12	5B:90:03:BB:BC:25
8	12	B9:42:27:A3:7F:1F
9	2	DD:0D:26:52:78:35
10	2	C4:42:25:1F:DA:94

The VMPS server contains a database with all VLAN to MAC address mapping, allowing the "dynamic" VLAN configuration of these hosts, no matter where they are located within the network.

VLANs: Setting dynamic VLANs

Las líneas que empiezan con el carácter '!' son comentarios. A continuación hay una breve descripción:

- Hay que definir un ***VMPS domain*** (línea 6), este dominio debe coincidir con el **dominio VTP** del switch. VTP (Virtual Trunking Protocol) es un protocolo propietario de CISCO que permite propagar la configuración de las VLANs a todos los switches de un mismo dominio. Por ejemplo, al crear una VLAN en un servidor VTP, la VLAN se propaga a todo el dominio.

- VMPS puede operar en modo *open* o *secure* (línea 7). En **modo *open***: Si una MAC no está definida, se le asigna una VLAN por defecto (línea 8). En **modo *secure***: Si una MAC no está definida se bloquea el puerto. Para desbloquear un puerto hay que ejecutar los comandos *shutdown / no shutdown*.

- En la sección ***MAC Addresses*** (línea 11) se asignan las VLANs a las que pertenecen las direcciones MAC. Puede usarse **--NONE--** para denegar explícitamente el acceso a cualquier VLAN. Notar que para identificar las VLANs se usa el **VLAN-name**, no el VLAN-id. En el switch deben haberse creado las VLANs con el mismo nombre que el indicado en esta sección del fichero de configuración.

VLANs: Setting dynamic VLANs

1. **!vmps domain <domain-name> - The VMPS domain must be defined.**
2. **!vmps mode { open | secure } - The default mode is open.**
3. **!vmps fallback <vlan-name>**
4. **!vmps no-domain-req { allow | deny } - The default value is allow.**
5. **!**
6. vmps domain mydomain
7. vmps mode open
8. vmps fallback --NONE--
9. vmps no-domain-req deny
10. **!!**
11. **!MAC Addresses**
12. **! address <addr> vlan-name <vlan_name>**
13. **!**
14. vmps-mac-addrs
15. **!**
16. address 0010.a49f.30e1 vlan-name --DEFAULT--
17. **! disabled - no access**
18. address 0010.a49f.30e2 vlan-name --NONE--
19. **! vlan TEST restricted**
20. address 0010.a49f.30e3 vlan-name TEST
21. **! vlan TEST1 unrestricted**
22. address 0010.a49f.30e4 vlan-name TEST1

VLANs: Setting dynamic VLANs

- **Una VLAN se puede restringir a un switch específico, o a un grupo de puertos de un switch**. Para ello hay que especificar:

1. Los **puertos permitidos** (sección *Port Groups*, línea 24). Por ejemplo, la línea 31 especifica el puerto 2/4 del switch 10.0.0.1, y la línea 32 especifica todos los puertos del switch 10.0.0.2,

2. **Las VLANs a las que se les aplicará alguna restricción** (sección *VLAN groups*, la línea 34),

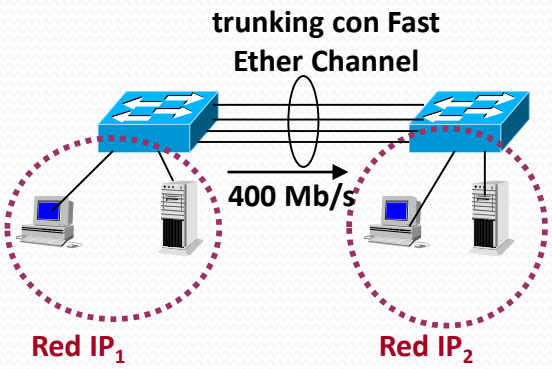
3. **La asociación entre las definiciones anteriores** (sección *VLAN port Policies*, línea 43).

VLANs: Setting dynamic VLANs

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!Port Groups
!vmps-port-group <group-name>
! default-vlan <vlan-name>
! fallback-vlan <vlan-name>
! device <device-id> { port <port-name> | all-ports }
!
vmps-port-group myswitch
device 10.0.0.1 port 2/4
device 10.0.0.2 all-ports
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!VLAN groups
!vmps-vlan-group <group-name>
! vlan-name <vlan-name>
!
vmps-vlan-group myvlans
vlan-name TEST
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!VLAN port Policies
!vmps-port-policies {vlan-name <vlan_name> | vlan-group <group-name> }
! { port-group <group-name> | device <device-id> port <port-name> }
!
vmps-port-policies vlan-group myvlans
port-group myswitch
```

Link Aggregation in L2

- **Link aggregation**: technique consisting in using several (around N=2-4) Ethernet links in order to increase the capacity to NxC Mb/s (for each Full Duplex direction)
  - Load balancing policies: based in L2 (MACs), L3 (IPs) or L4 (ports) or flows
  - Port redundancy (if a link fails, the rest of links still work)
- i.e. **IEEE 802.3ad** for link aggregation
- i.e. CISCO Fast Ether Channel or Giga Ether Channel **(port trunking)**
- i.e. others call it **NIC-team(ing)**



Aggregation implies a reduction in the number of physical ports in the switch that normally are used for other purposes (e.g. hosts).

Options:

- **add switches**
- **put a switch with more ports (normally if more than 50% of the switch ports are used in aggregation.**

- **Port aggregation: CISCO IoS (Sw 2950)**

Setting an aggregated port

!!!! Create the port-channel and assign ports

!!!! The ports have to be in the same VLAN or to be trunk

Sw(conf)# **interface port-channel** 1

Sw(conf)# **interface** Ge1

Sw(config-if)# **channel-group** 1 **mode on**

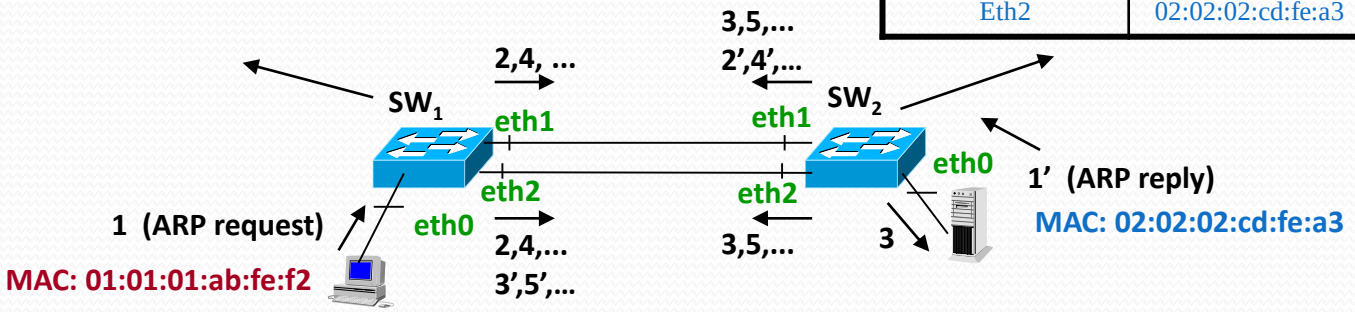
Sw(conf)# **interface** Ge2

Sw(config-if)# **channel-group** 1 **mode on**

Broadcast Storms:

PORT (IFACE)	MAC	AGE
Eth0	01:01:01:ab:fe:f2	10'
Eth1	01:01:01:ab:fe:f2	10'
Eth2	01:01:01:ab:fe:f2	10'
Eth2	02:02:02:cd:fe:a3	10'

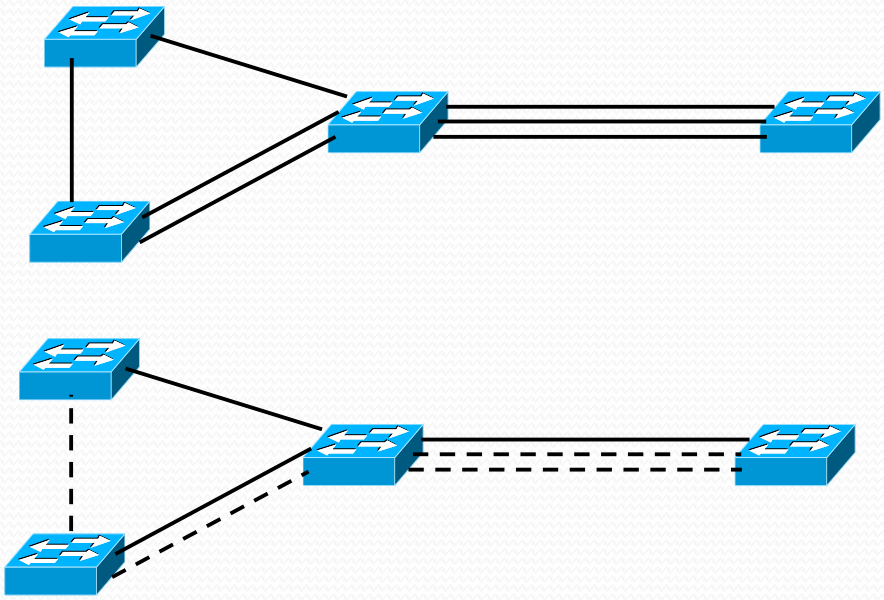
PORT (IFACE)	MAC	AGE
Eth0	02:02:02:cd:fe:a3	10'
Eth1	01:01:01:ab:fe:f2	10'
Eth2	01:01:01:ab:fe:f2	10'
Eth1	02:02:02:cd:fe:a3	10'
Eth2	02:02:02:cd:fe:a3	10'



(1) Station 1 sends a frame in broadcast (e.g. an ARP frame)
(2) Sw1 forwards in eth1 and eth2
(3) Sw2 forwards again (e.g. eth1 to eth2 and eth0)
(4) Sw1 forwards again → infinite loop

- If the broadcast produces a loop (i.e. due to ARP requests that have MAC_{dest} = ff:ff:ff:ff:ff:ff) the growth in number of frames is exponential → the loop will produce a Sw failure (load causes around 80% of the CPU of the Sw but routers also will process these broadcasts) → the whole network fails,
- Does IPv6 produce broadcast storms ? ARP does not exist. NDP uses multicast IPv6 destination addresses that translates to broadcast MAC destination, thus, YES, it also produces broadcast storms.

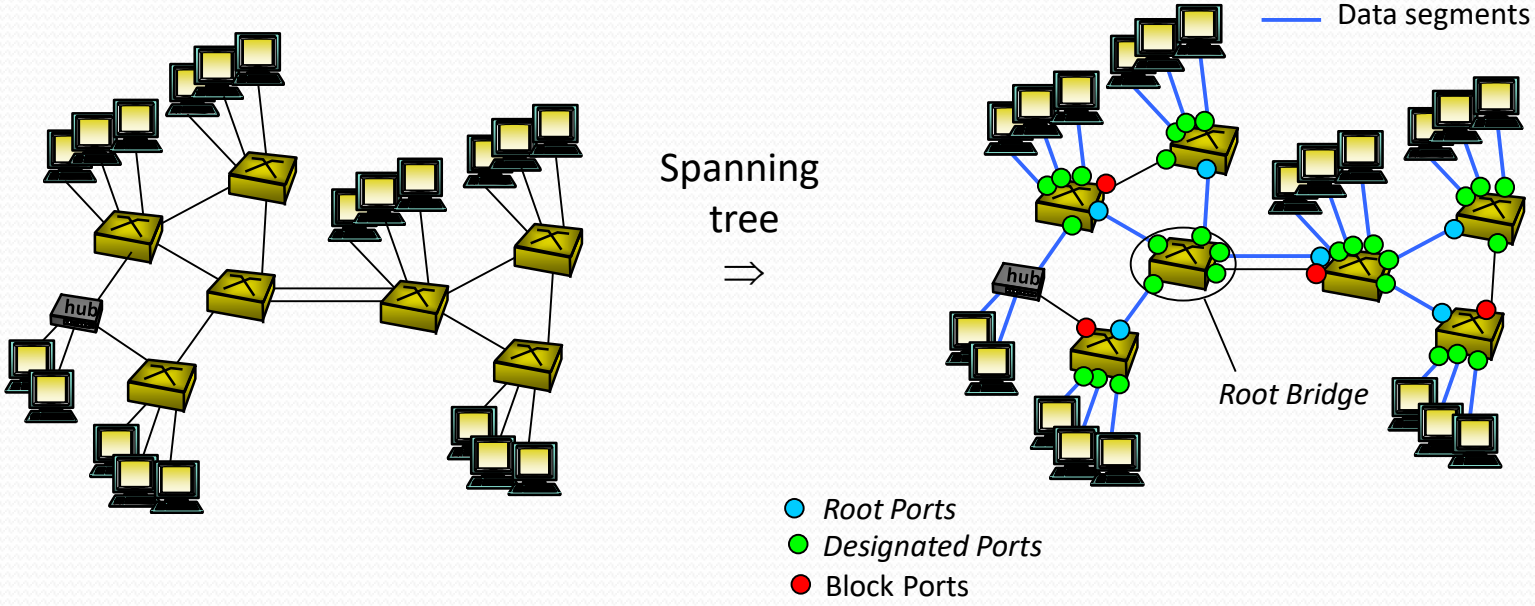
- **Spanning Tree Protocol (STP, IEEE 802.1D):**
  - Main objective:
    - Avoid and eliminate *loops and redundant links* organizing the network in a tree topology (*blocking* those switch ports that would produce a loop)
    - Use costs based on link rates as metrics



Link capacity	Cost
10 Gb/s	2
1 Gb/s	4
100 Mb/s	19
10 Mb/s	100

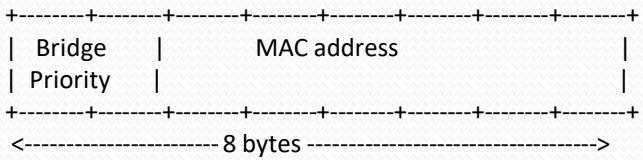
- **In order to build a Spanning Tree, the protocol has to choose:**

1. A ***Root Bridge*** **(RB)** for the whole broadcast domain. Note: s*witch and bridge* has the same meaning,
2. A ***Root Port*** for each switch that is not the RB, this allows sending traffic towards the RB. Root ports guarantee that a tree topology is built,
3. A ***Designated Port*** at each segment. Guarantees that all segments are reachable. A root port can never be a designated port,
4. Those port not elected as *Root Ports* or *Designated Ports*, will be **Blocked Ports**.



For the election of *Root Bridge, Root Port, Designated Port*:

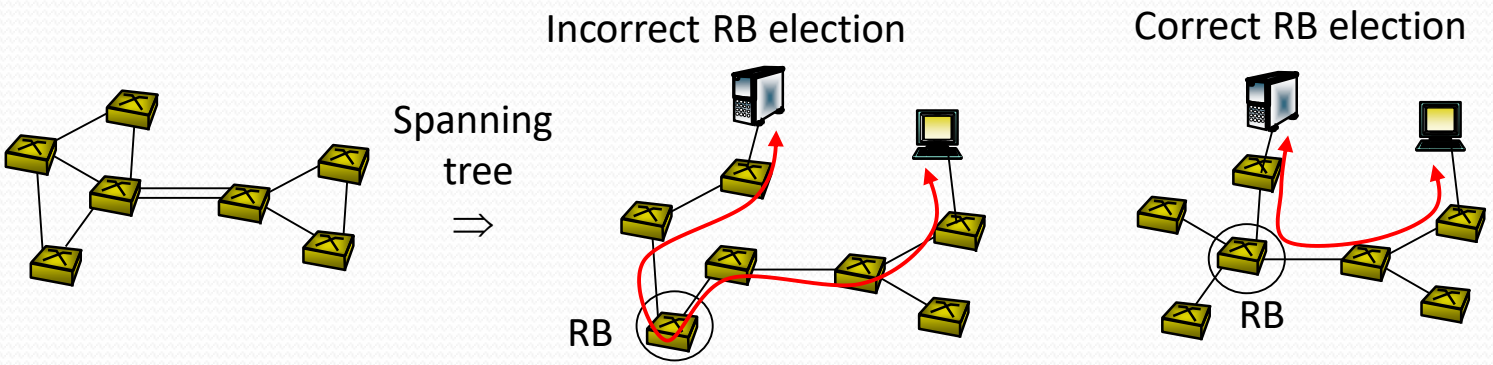
- Switches are identified with a ***Bridge ID (BID)*** formed with a priority field (manually configurable) and one of the MACs addresses of the switch, **BID = Priority (2B) + MAC Address (6B).** The lowest BID the highest priority in the algorithms, default Priority value = $8000_{(hex)}$ = $32768_{(dec)}$:



  - Signaling messages are sent: ***Bridge Protocol Data Unit*** **(BPDU)**.
- Initially the BPDUs are sent every 2 seconds to the **multicast address** 01:80:C2:00:00:00. The BPDUs fields used in the tree calculation are:
  - ***Root BID*** (8 bytes): BID of the Root Bridge
  - ***Root Path Cost*** (4 bytes): Is incremented with the cost of the port where is received.
  - ***Sender BID*** (8 bytes): BID of the switch that sends the BPDU.
  - ***Port ID*** (2 bytes): ID of the port that transmits the BPDU (all the ports of the same switch have different IDs and have a port-priority). Thus, **PORT-ID = Priority (1B) + port# (1B)**. Default Priority value = $128_{(dec)}$, e.g., for a port labeled Fe5, its **PORT-ID = 128:5.** The lowest PORT-ID the highest priority in the algorithms.

Root Bridge (RB) election:

- Initially all switches generate BPDUs with *Root BID* = *Sender BID*.
- If a switch receive a BPDU with a lower *Root BID*, stops sending its own BPDUs and assumes that BID as *Root BID*. In small amount of time **only the RB generates BPDUs**. **The other switches modify the *Root Path Cost, Sender BID* y *Port ID* before re-sending those RB BPDU's**. The RB sends BPDUs for all its ports. **The other switches only send BPDUs received from the *Root Port***.
- The **Bridge/Switch priority** may be manually configurable. Initially values **0x8000 (32768).** Lowest have more priority:
  - Switch# **spanning-tree vlan** *vlan-id* **priority** *priority*
- The election of the RB may impact performance: should be the more centric switch (star topology is better than tree topology).

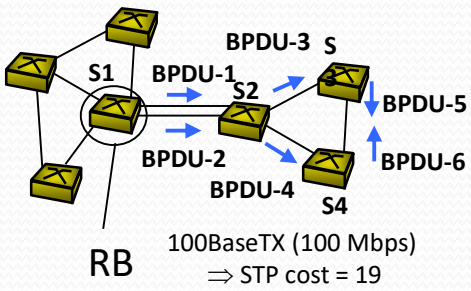


Root Port election:

- **Each switch that is not the RB selects a port as *Root Port*.**
- For the election the information contained in the received BPDUs is compared at each port. **The selected port is the one that has received a BPDU that fulfils the following sequence of conditions**:
  1. Lowest *Root BID* (towards the *Root Bridge*).
  2. Lowest *Root Path Cost* (optimal path towards the *Root Bridge*).
  3. Lowest *Sender BID*
  4. Lowest *Port ID*

  } To assure the selection is unique.

Root BID = BID-S1 = 00:00:00:00:00:11:11:11
BID-S2 = 80:00:00:00:00:22:22:22
BID-S3 = 80:00:00:00:00:33:33:33
BID-S4 = 80:00:00:00:00:44:44:44



⇒

BPDU-1: Root BID = Sender BID = BID-S1
Root Path Cost = 0, Port ID = 1

BPDU-2: Root BID = BID-S1, Sender BID = BID-S1
Root Path Cost = 0, Port ID = 2

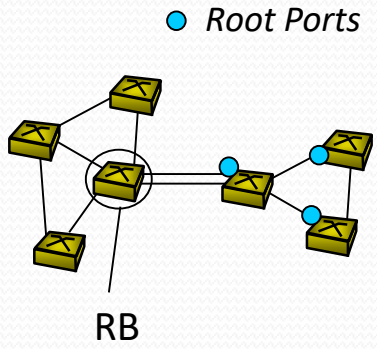
BPDU-3: Root BID = BID-S1, Sender BID = BID-S2
Root Path Cost = 19, Port ID = 1

BPDU-4: Root BID = BID-S1, Sender BID = BID-S2
Root Path Cost = 19, Port ID = 2

BPDU-5: Root BID = BID-S1, Sender BID = BID-S3
Root Path Cost = 38, Port ID = 1

BPDU-6: Root BID = BID-S1, Sender BID = BID-S4
Root Path Cost = 38, Port ID = 1

⇒

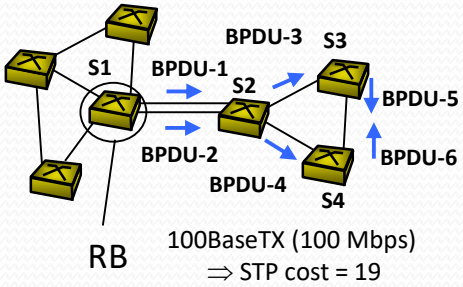


Designated Port Election (collision domain access):

- **All RB ports are *Designated Ports***, except those ones that may form a loop at level 1 (two ports connected to a hub or a crossover cable between two ports).
- **For the other switches:**
- The ports that do not receive BPDUs are *Designated Ports* (Host ports),
- The ports that receive BPDUs and are not *Root Ports*: Compare the information contained by BPDUs received and sent in that port. **The port is *Designated Port* if fulfils the following sequence of conditions**:
  - Lowest *Root BID* (towards the *Root Bridge*).
  - Lowest *Root Path Cost* (optimal path towards the *Root Bridge*).
  - Lowest *Sender BID*
  - Lowest *Port ID*

  } To assure the selection is unique.

Root BID = BID-S1 = 00:00:00:00:00:11:11:11
BID-S2 = 80:00:00:00:00:22:22:22
BID-S3 = 80:00:00:00:00:33:33:33
BID-S4 = 80:00:00:00:00:44:44:44



⇒

BPDU-1: Root BID = Sender BID = BID-S1
Root Path Cost = 0, Port ID = 1

BPDU-2: Root BID = BID-S1, Sender BID = BID-S1
Root Path Cost = 0, Port ID = 2

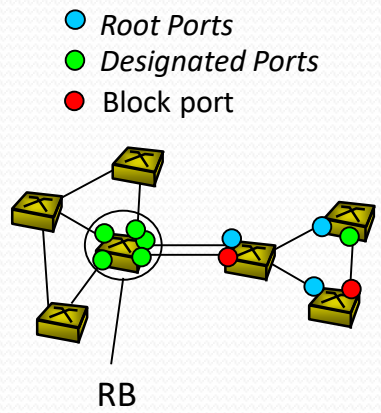
BPDU-3: Root BID = BID-S1, Sender BID = BID-S2
Root Path Cost = 19, Port ID = 1

BPDU-4: Root BID = BID-S1, Sender BID = BID-S2
Root Path Cost = 19, Port ID = 2

BPDU-5: Root BID = BID-S1, Sender BID = BID-S3
Root Path Cost = 38, Port ID = 1

BPDU-6: Root BID = BID-S1, Sender BID = BID-S4
Root Path Cost = 38, Port ID = 1

⇒



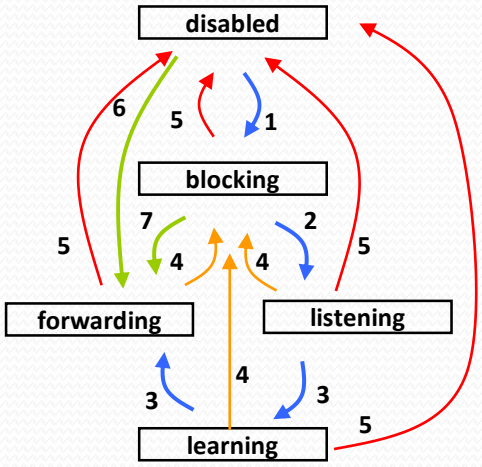
Topic 2: Corporate Networks: Switching Blocks

Port State:

- **Blocking** – No frame Forwarding. BPDUs are listened (at least 20 seconds, **selects RB, root & designated ports**),
- **Listening** - No frame Forwarding. BPDUs are listened/transmitted (15 seconds, **assures the topology does not change**),
- **Learning** – No frame Forwarding. Learning addresses. BPDUs are listened/transmitted (15 seconds, **builds the MAC address table**),
- **Forwarding** - Forwarding of frames and Learning addresses. BPDUs are listened/transmitted,
- **Disabled** - No frame Forwarding and BPDUs are not listened/transmitted.

STP Timers:

- ***Hello***: time between BPDUs sent by a switch *Root Bridge* (2 seconds).
- ***Forward:*** time spent in the *listening/learning states* (15 seconds at each).
- ***Max Age***: maximum time that a BPDU is stored (20 seconds). If no more BPDUs are received, go to the next STP state (*listening*).



State transitions:

1. Initiate or *no shutdown.*
2. *Root* or *Designated* port selected, or *timer Max. Age* expires
3. *Timer forwarding* expires (15 seconds).
4. The ports is no more *Root* or *Designated*. Initially all switches assume they are *Root Bridge* and all their ports are *Designated Ports*.
5. *Shutdown*

CISCO:

6. Port Fast: thought in case a host is connected directly to the switch. If the switch detects a loop, go to *blocking*.
7. UplinkFast: thought for *edge routers*. The switch take into account the redundant links for substituting them rapidly for a *Root Port i*n case fails.

STP+VLAN: CISCO IoS (Sw 2950)

Setting STP in VLANs

!!!! Assign a STP instance to a VLAN

Sw(conf)# **spanning-tree vlan** vlan2

!!!! Select this Switch as Root Bridge for VLAN2

Switch# **spanning-tree vlan** vlan2 **root primary**

!!!! Other way: change the switch priority to some lower value

Switch# **spanning-tree vlan** vlan2 **priority** 0x7000

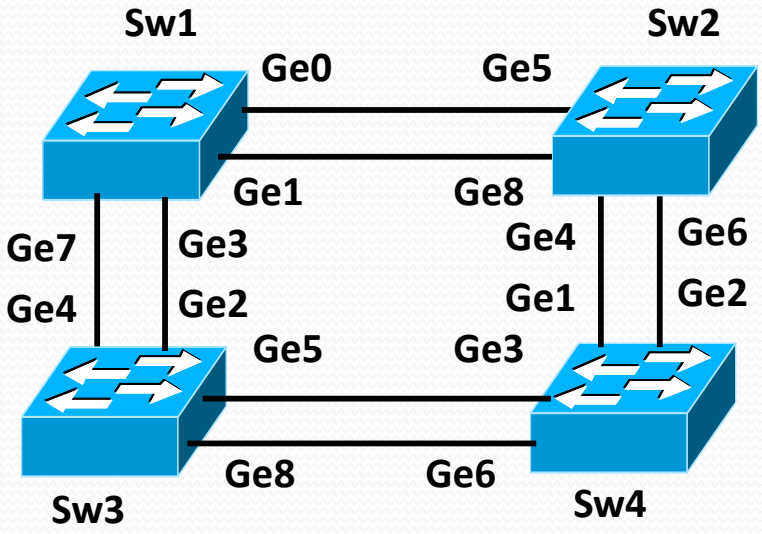
!!!! Modify cost and priority to a port

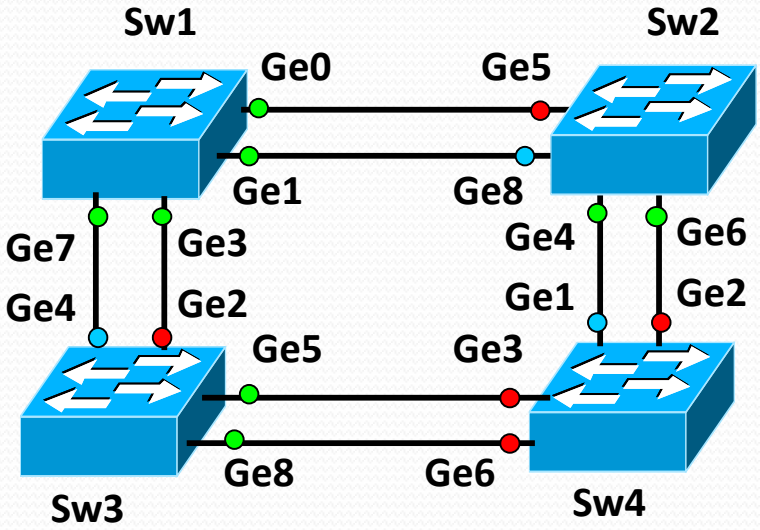
Sw(conf)# **interface** Ge0

Sw(config-if)# **spanning-tree vlan** vlan2 **cost** 5

Sw(config-if)# **spanning-tree vlan** vlan2 **port-priority** 120

- **Spanning Tree Protocol (exercise):**





1. If MAC-sw2 < MAC-sw4 < MAC-sw1 < MAC-sw3, define the resulting topology of the left network,

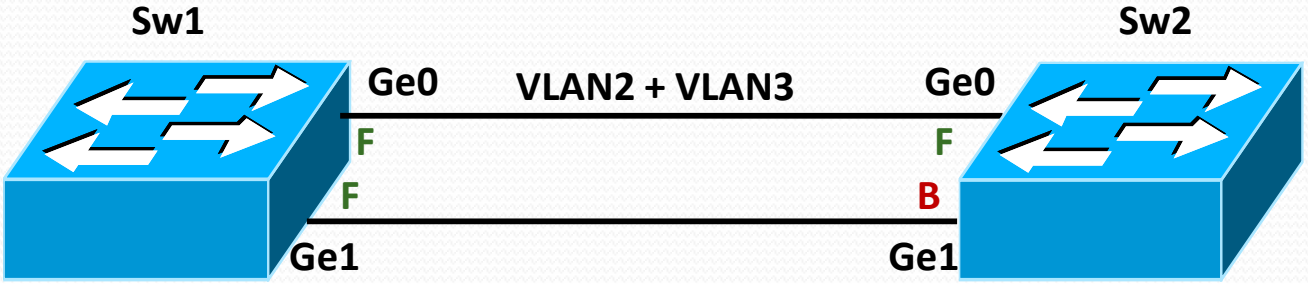
2. Define priorities to obtain the topology of the rigth part assuming that you have the left network with MAC-sw2 < MAC-sw4 < MAC-sw1 < MAC-sw3.

Spanning Tree Protocol and VLANs

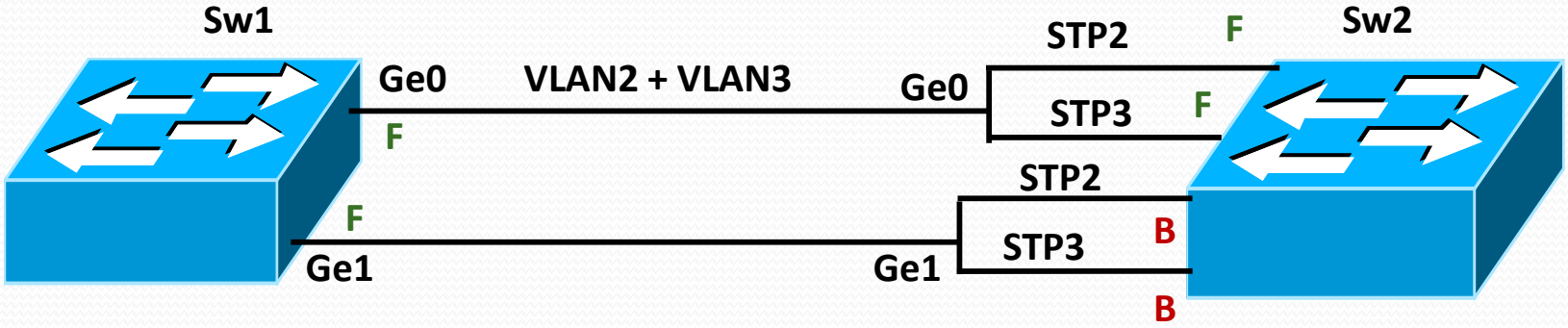
- **STP** (IEEE 802.1D) initially design for **one** VLAN but that works over all VLANs
- VLAN (IEEE 802.1Q): tagging protocol for VLANs over trunk ports,
  - ISL: CISCO VLAN tagging protocol also used over trunk ports,
  - CISCO defines **PVST** (Per VLAN Spanning Tree) that defines *one STP instance per VLAN* → works with ISL and is not compatible with 802.1Q
  - CISCO defines **PVST+** that is compatible with 802.1Q
- IEEE adopts (2003) the concept of one STP instance per VLAN
  - Initially proposes 802.1s or ***Multiple Spanning Tree Protocol*** (MSTP)
    - MSTP allows regions of MST that may run multiple MST instances. These regions are interconnected using a unique common spanning tree (CST).
  - Finally MSTP is included in the 802.1Q
- IEEE 802.1w (**RSTP, Rapid-STP**): reduces the time of convergence to around 3*Time_between_BPDU's = 3*2 = 6 seconds. It is compatible with MSTP.

- **Spanning Tree Protocol and VLANs**

CASO 1 (IEEE 802.1D): STP: VLAN2 + VLAN3



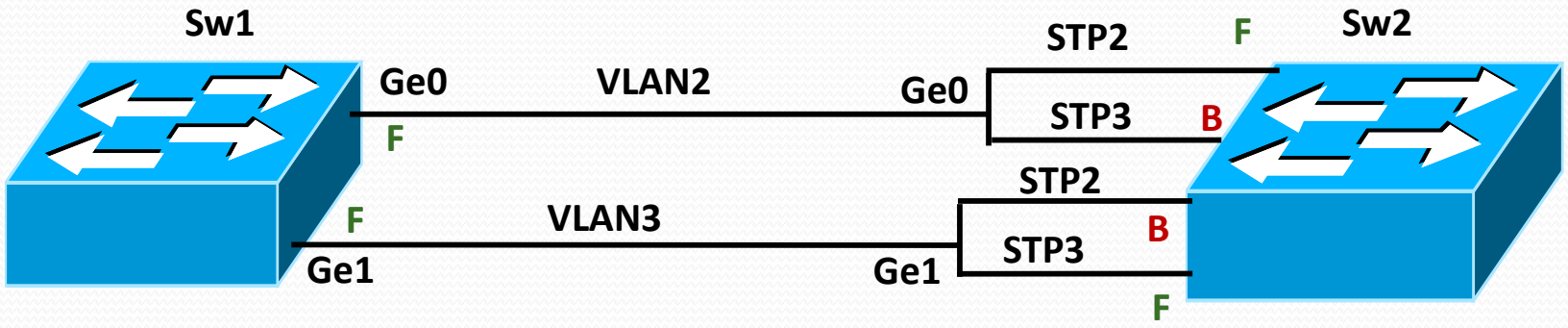
CASO 2 (MSTP): STP2: VLAN2 and STP3: VLAN3



No hay control de prioridades en los puertos (port-priority), el STP no logra balancear la carga de las VLANs ya que STP2 y STP3 bloquean los mismos puertos

- **Spanning Tree Protocol and VLANs**

CASO 3 (MSTP): STP2: VLAN2 and STP3: VLAN3



Hay control de prioridades en los puertos (port-priority, value from 0-255, default 128), el STP logra balancear la carga de las VLANs. Cuidado, el port-priority que hay que manipular es el de Sw1 (el transmisor para que Sw2 decida)

Para ello hay que poner en la Ge0 una prioridad menor en el STP2 (VLAN2) que en la Ge1 del mismo STP2. De manera simétrica, hay que poner en la Ge0 una prioridad mayor en el STP3 (VLAN3) que en la Ge1 del mismo STP3.

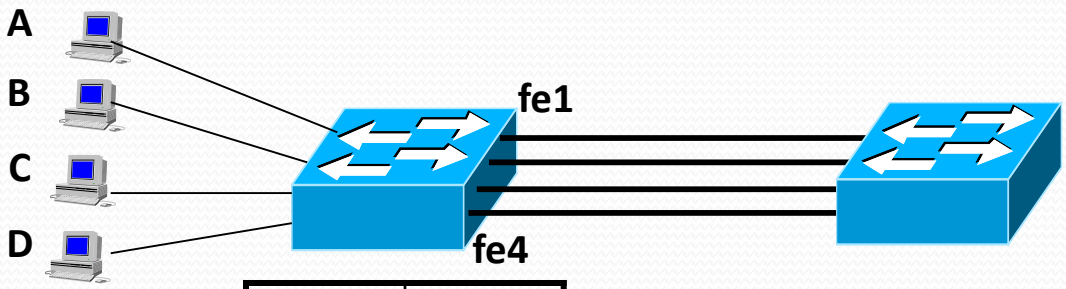
Ahora, los STP2 y STP3 bloquean puertos distintos.

- **MSTP+VLAN: CISCO IoS (Sw 2950)**

Splitting ports per VLAN and STP instance

```
!!!! The port has to be trunk
Sw1(conf)# interface Ge0
Sw1(config-if)# switchport mode trunk
!!!! Play with the port priority at each VLAN and each STP instance. For
!!!! Example, Ge0, highest priority for VLAN2 than in Ge1
Sw1(config)# interface Ge0
Sw1(config-if)# spanning-tree vlan vlan2 port-priority 64
Sw1(config-if)# spanning-tree vlan vlan3 port-priority 128
Sw1(config)# interface Ge1
Sw1(config-if)# spanning-tree vlan vlan2 port-priority 128
Sw1(config-if)# spanning-tree vlan vlan3 port-priority 64
```

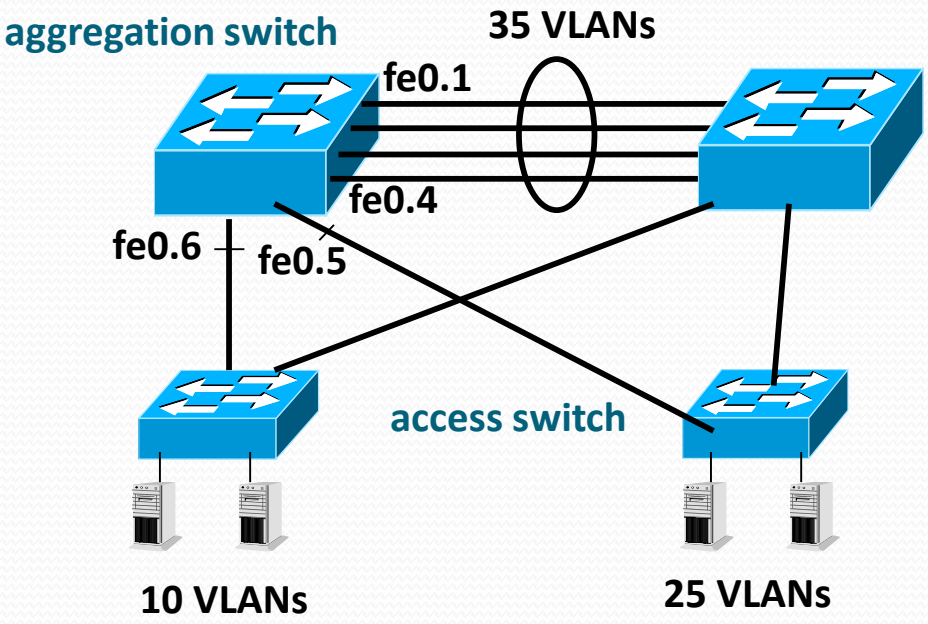
- **Spanning Tree Protocol and Link Aggregation**
  - i.e how does STP works with Fast Etherchannel:
    - Fast EtherChannel with 4 parallel ports usually works with flow load balancing
    - The parallel links are treated by STP as a unique link



Flow	Port
A	fe1
B	fe2
C	fe3
etc	...

- **Limit in the number of STP instances and VLANs in a switched network:**
  - IEEE 802.1Q (VLAN) assumes a tagging scheme of **n**-bits for a VLAN label, e.g., if n=12, then 2^{12} = 4096 VLAN labels can be created,
  - Using MSTP (IEEE 802.1s) allow us to have 1 STP instance for VLAN. The question is whether the Hw (switches) allow this amount of STP instances ?
  - You have to assume that each instance generates its own BPDU's,
  - The limitation is in what is called:
    - The **number of Virtual ports per Line Card**
    - The **number of STP Logical ports active**

- **Number of virtual ports per Line Card**
  - Virtual ports are a per-line card value that reflects the total number of spanning tree processing instances used on a particular line card.
    - Each Line Card may have several interfaces: e.g. Line Card FE0 has FE0/0, FE0/1, ..., FE0/N, and the Line Card FE1 has FE1/0, FE1/1, ..., FE1/M, etc



#Virtual_ports = Sum of trunks * #VLANS

sh vlan virtual-port slot 0

Port	Virtual-ports
fe0.1	35
fe0.2	35
fe0.3	35
fe0.4	35
fe0.5	25
fe0.6	10
Total	175

Fast EtherChannel

Virtual ports: number of VLAN's supported by trunks in a Line Card and then a limit in the number of Spanning tree instances in the Line Card.

- **Number of virtual ports per Line Card**
  - Example: let's imagine a Line Card that supports 1500 Virtual Ports.
    - If we have a 48 switch port with 42 aggregated trunk ports and 6 ports with x VLAN's (the x VLAN's represent the total number of VLAN between these 6 ports), how many VLANs can support these 6 ports?

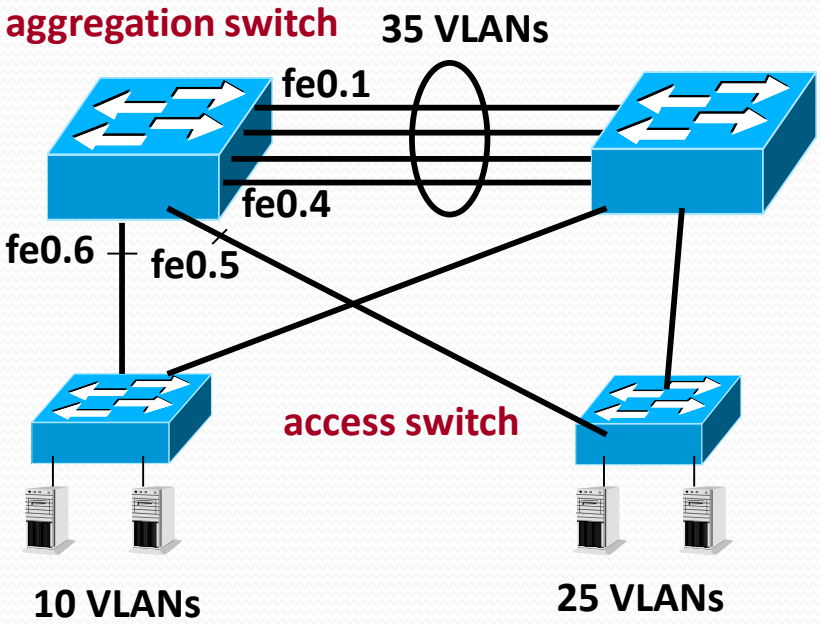
If x is the number of VLAN's that we may allocate. Since, the 6 access ports may upload in total x VLANs and each trunk may aggregate x VLANs each:

$$42*x + x \leq 1500 \rightarrow \qquad 43*x \leq 1500 \rightarrow$$

$$x \leq 34.88 \rightarrow \text{then 34 VLANs between the 6 ports}$$

Virtual ports: number of VLAN's supported by trunks in a Line Card and then a limit in the number of Spanning tree instances in the Line Card.

- **Total number of STP logical ports active**
  - System-wide value that reflects the total number of spanning tree processing instances used in the whole system
    - Calculated for the whole switching module (broadcast network).
  - For an aggregated module:



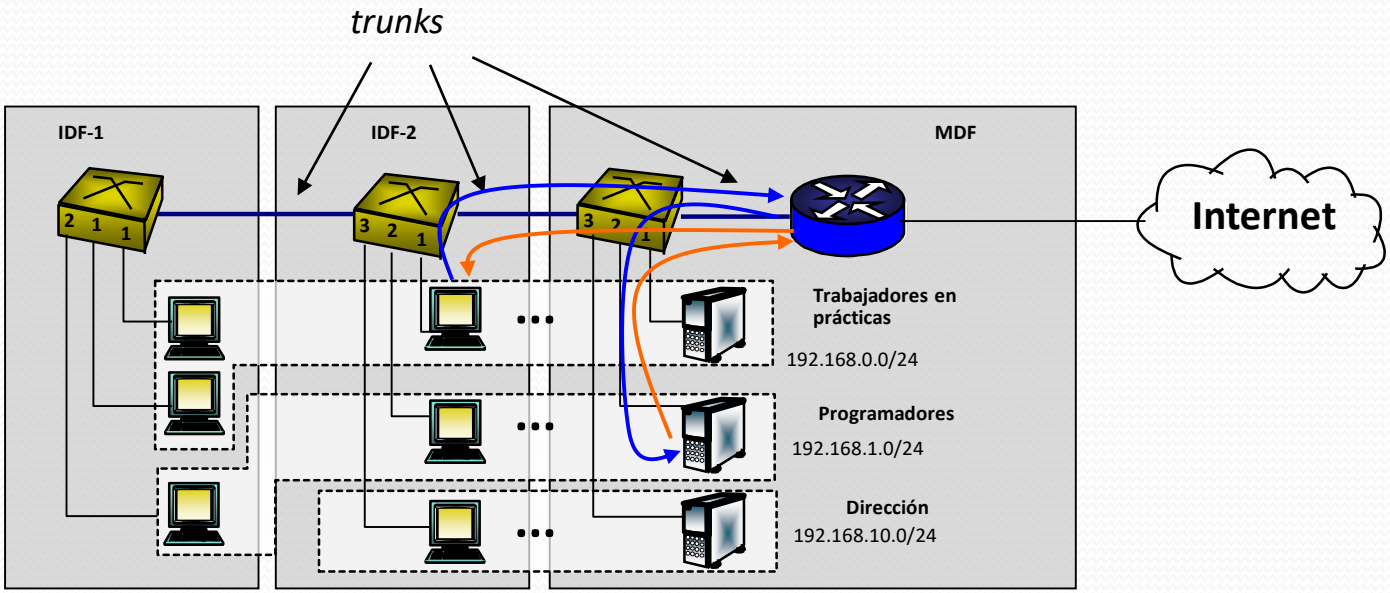
#logical ports = trunks * #VLANS + nontrunks

fe0.1-fe0.4 → 35 VLANS
fe0.5 → 25 VLANS
fe0.6 → 10 VLANS

Total → 70 VLANs

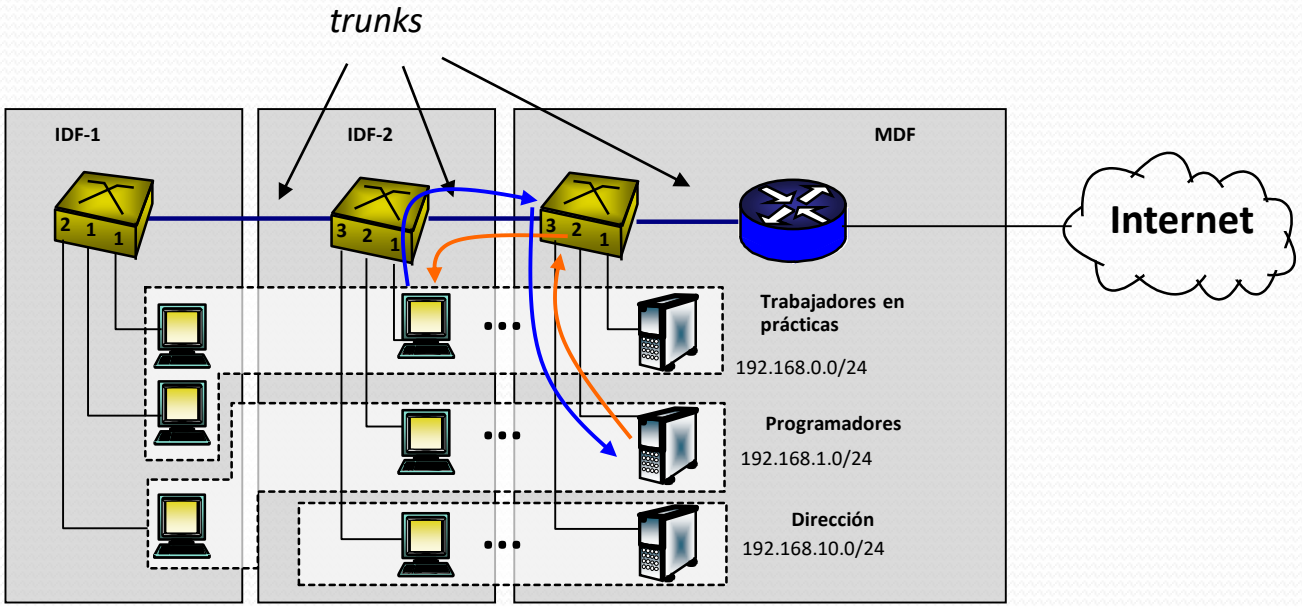
STP Logical Port active: number of STP instances in the whole switching system

- ***MultiLayer Switching* (MLS) or L3 Switches**
  - **Objective:**
    - Assume the following scenario: **a VLAN-1 host access to a VLAN-2 server**.
    - Problem: **frames has to cross the *trunk* link between the switch and router**! ⇒ this *trunk* link will be the bottleneck, since it has to process many packets that go from one VLAN to another VLAN over the same link. The packets have to be processed at L3 (implying looking at the routing table, a Sf process),
    - L3 switches **improve latency** by processing L3 packets faster than routers.



MLS working:

- **When the first IP datagram from a *flow* directed to the router crosses the MLS-switch, this one registers the flow** in one of the following ways (i) IP destination address, (ii) IP source and destination address, (iii) IP addresses and ports, in general using a hash function,
- **When the first IP packet of a flow crosses the switch, activates MLS for that flow using a cache** (first time is necessary to look-up the routing table in order to fill the cache)
- **Any IP packets from that flow arriving to the switch are fast routed towards destination** (the switch works as a router!), after **looking to the cache MLS table**.

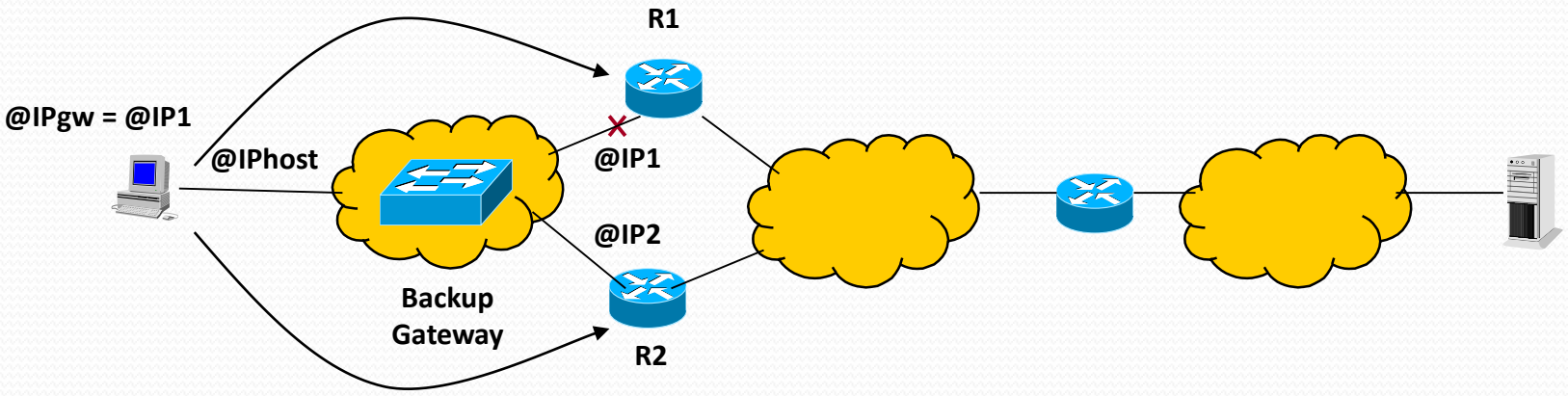


Fail Tolerance in L3 (loss of gateway):

Main Host Objective: obtain a default route to leave the network.

Solution:

- **Between routers**: Dynamic routing allows dynamic default routing configuration → so, the routing protocol is fail tolerant by definition:
  - If a network point fails, the routing protocol is in charge of reconfiguring any routing table,
- **At hosts**: most Hosts and Servers use a default route statically configured or obtained via DHCP
  - If a point of failure appears → hosts/servers loss connectivity.



Virtual Router Redundancy Protocol (VRRP)

- Design to eliminate points of failure related to default routes
- **Terminology:**
  - **VRRP router**: a router running VRRP,
  - **Virtual router:** abstract object (VRRP instance) used by an VRRP router that acts as gateway (default router) for hosts in a LAN:
    - identification of virtual router (VRRP-ID) + set of common @IP in a VLAN
    - A router VRRP may be bound to **several** virtual routers (VRRP instances).
  - **Owner @IP:** the physical @IP of the virtual router. The router uses this @IP to respond for ICMP pings, TCP connections, etc,
  - **Primary @IP:** @IP used to send VRRP advertisements and used as IP gateway,
  - **Virtual router master:** VRRP router responsible of the IP packet forwarding (e.g. the one that answers ARPs frames),
  - **Virtual router backup:** backup router that takes master responsibilities if this one fails.

- **Gratuitous ARP (Request/Reply)**
  - **Gratuitous ARP-request:** ARP request packet where the $@IP_{source}$ and $@IP_{target}$ are both set to the IP of the host sending the packet and the $@MAC_{destination}$=ff:ff:ff:ff:ff:ff (broadcast address).
  - **Gratuitous ARP reply:** a reply to which no request has been made

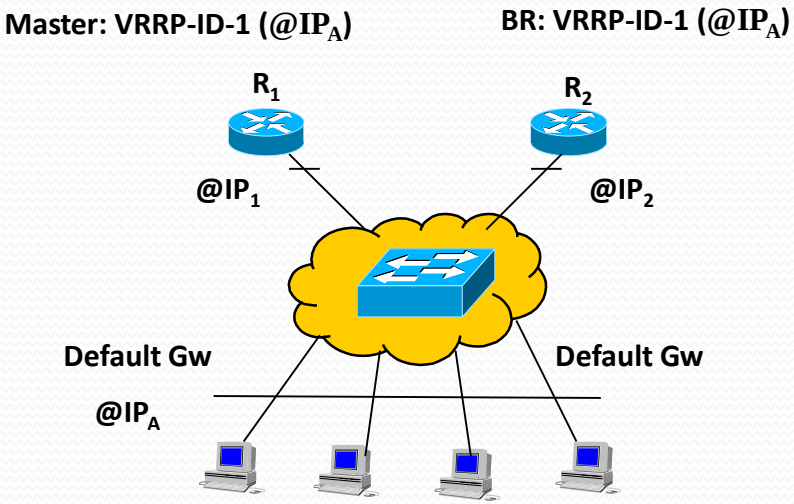
- **Gratuitous ARP objectives:**
  - Detect IP conflicts (@IP duplications)
  - Clear ARP caches
  - Update ARP caches in other hosts (e.g. because we have changed the NIC IP address)
  - Fill MAC Tables in switches
  - …

Examples of use of Gratuitous ARPs:

- They can help **detect IP conflicts**. When a machine receives an ARP request containing a source IP that matches its own, then it knows there is an IP conflict.
- They **assist in the updating of other machines' ARP tables**. Clustering solutions utilize this when they move an IP from one NIC to another, or from one machine to another. Other machines maintain an ARP table that contains the MAC associated with an IP. When the cluster needs to move the IP to a different NIC, be it on the same machine or a different one, it reconfigures the NICs appropriately then broadcasts a gratuitous ARP reply to inform the neighboring machines about the change in MAC for the IP. Machines receiving the ARP packet then update their ARP tables with the new MAC.
- They **inform switches of the MAC address of the machine on a given switch port**, so that the switch knows that it should transmit packets sent to that MAC address on that switch port.
- **Every time an IP interface or link goes up**, the driver for that interface will typically **send a gratuitous ARP to preload the ARP tables of all other local hosts**. Thus, a gratuitous ARP will tell us that that host just has had a link up event, such as a link bounce, a machine just being rebooted or the user/sysadmin on that host just configuring the interface up. If we see multiple gratuitous ARPs from the same host frequently, it can be an indication of bad Ethernet hardware/cabling resulting in frequent link bounces.

Virtual Router Redundancy Protocol

- E.g. with one Virtual Routers in the network (only 1 VLAN)



R$_1$: has @IP$_1$ as owner @IP
R$_2$: has @IP$_2$ as owner @IP
VRRP-ID-1: has @IP$_A$ as primary @IP
4 Hosts has @IP$_A$ as @IP$_{Gw}$

VRRP-ID-1 identifies a Virtual Router and is associated to the primary @IP$_A$,

When R$_1$ activates VRRP is announced as master of VRRP-ID-1 with priority 200 (highest number implies better priority to become master),

When R$_2$ activates VRRP is announced as backup of VRRP-ID-1 with priority 100,

While both routers work well all traffic leave using R$_1$ and R$_2$ is not used (there is not load balancing),

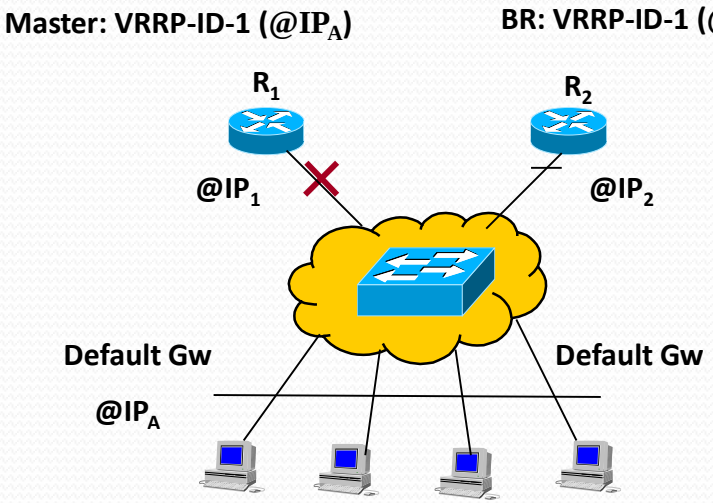
Virtual Router Redundancy Protocol

- While a router works as Master, it has an @IP associated to the Virtual Router.
- **In Master state a router**:
  - SHOULD answer to ARP requests directed to the @IP associated to the virtual router
    - uses the **virtual @MAC:** is not the physical MAC address of the interface, but a MAC address with ID **00:00:5E:00:01:{VRRP-ID}.** **For Example VRRP-ID=1, @MAC = 00:00:5E:00:01:01,**
  - SHOULD forward packets with $@MAC_{dst}$ = virtual @MAC of the virtual router
  - SHOULD NOT accept packets addressed to the @IP associated to the virtual router to which is not @IP owner
  - SHOULD accept packets addressed to the @IP associated to the virtual router

- **Virtual Router Redundancy Protocol**
  - While a router is in Backup state monitors the availability of the Master router
  - In backup state, a router:
    - SHOULD NOT answer to ARP requests directed to @IP associated to the virtual router
    - SHOULD discard packets with $@MAC_{dst}$ = virtual @MAC del router virtual
    - SHOULD NOT accept packets addressed to the @IP associated to the virtual router

Virtual Router Redundancy Protocol

- R_1 fails: the Master VRRP-1 fails



R_1: has @IP$_1$ as owner @IP
R_2: has @IP$_2$ as owner @IP
VRRP-ID-1: has @IP$_A$ as primary @IP
4 Hosts has @IP$_A$ as @IP$_{Gw}$

R_1 fails: R_2 detects the master failure (VRRP messages are not listened) and takes responsibility of the VRRP-ID-1 instance,

R_2 sends ARP gratuitous in order that i) all Hosts "clean" their ARP cache, and ii) switching tables are updated creating new L2 MAC entries pointing towards R_2, it is to say:

ARP gratuitous → If R_1 fails and R_2 sends a ARP request with @IP$_{dst}$= @IP$_{src}$= @IP$_A$ and @MAC$_{dst}$= broadcast and @MAC$_{src}$= @MAC$_{virtual\text{-}ID=1}$ → all hosts will refresh their ARP table with the MAC that R_2 wants to associate to the @IP$_A$

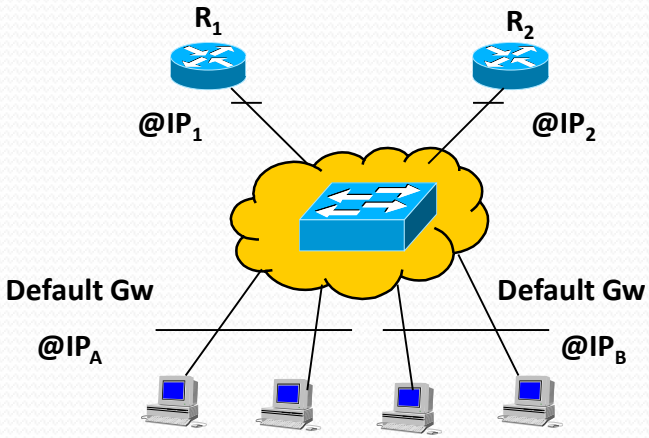
Note that VRRP uses the virtual MAC associated to the VRRP-ID for which the router is master, and does not use the real physical MAC address

Virtual Router Redundancy Protocol

- E.g. **with two Virtual Routers in the network (only 1 VLAN) and load balancing**

Master: VRRP-ID-1 ($@IP_A$)
BR: VRRP-ID-2 ($@IP_B$)

BR: VRRP-ID-1 ($@IP_A$)
Master: VRRP-ID-2 ($@IP_B$)



R_1: has $@IP_1$ as owner @IP
R_2: has $@IP_2$ as owner @IP
VRRP-ID-1: has $@IP_A$ as primary @IP
VRRP-ID-2: has $@IP_B$ as primary @IP
2 Hosts has $@IP_A$ as $@IP_{Gw}$
2 Hosts has $@IP_B$ as $@IP_{Gw}$

VRRP-ID-1 identifies a first instance in the Virtual Router and it is associated to the primary $@IP_A$, and

VRRP-ID-2 identifies a second instance in the Virtual Router and it is associated to the primary $@IP_B$,

When R_1 activates VRRP is announced as master of VRRP-ID-1 with priority 200 (its primary is $@IP_A$) and backup of VRRP-ID-2 with priority 100 (its primary is $@IP_B$),

When R_2 activates VRRP is announced as backup of VRRP-ID-1 with priority 100 (its primary is $@IP_A$) and master of VRRP-ID-2 with priority 200 (its primary is $@IP_B$),

While both routers work well there is load balancing, hosts with gateway $@IP_A$ use R_1 and hosts with gateway $@IP_B$ use R_2,

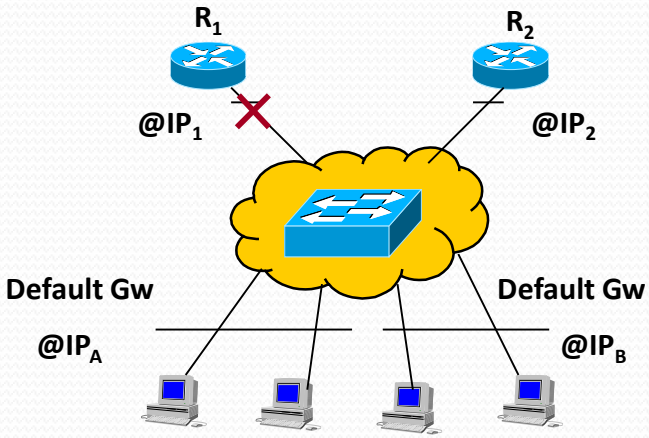
If R_1 fails, R_2 will take responsibility of routing packets as default Gw for that host that have Gw $@IP_A$ as address, and viceversa if R_2 fails.

Virtual Router Redundancy Protocol

- R_1 fails: the Master VRRP-1 fails

Master: VRRP-ID-1 ($@IP_A$)
BR: VRRP-ID-2 ($@IP_B$)

BR: VRRP-ID-1 ($@IP_A$)
Master: VRRP-ID-2 ($@IP_B$)



R_1: has $@IP_1$ as owner @IP
R_2: has $@IP_2$ as owner @IP
VRRP-ID-1: has $@IP_A$ as primary @IP
VRRP-ID-2: has $@IP_B$ as primary @IP
2 Hosts has $@IP_A$ as $@IP_{Gw}$
2 Hosts has $@IP_B$ as $@IP_{Gw}$

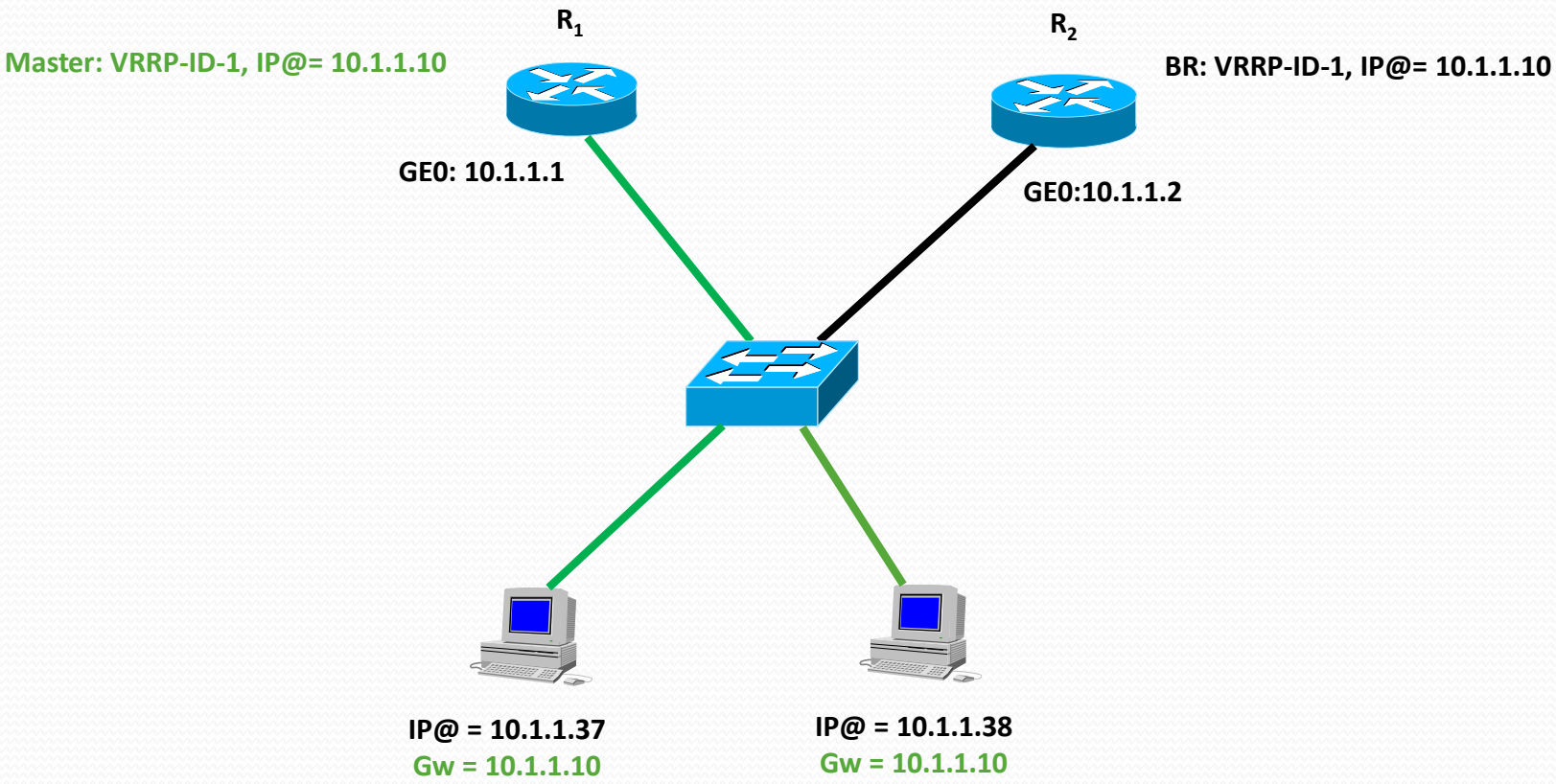
R_1 fails: R_2 detects the master failure of instance VRRP-ID-1 (VRRP messages are not listened) and detects that it acts as backup of VRRP-ID-1,

R_2 sends ARP gratuitous in order that i) all Hosts that have $@IP_A$ as primary "clean" their ARP cache, and ii) switching MAC tables are updated creating new L2 MAC entries pointing towards R_2, it is to say:

ARP gratuitous → If R_1 fails and R_2 sends a ARP request with $@IP_{dst}$= $@IP_{src}$= $@IP_A$, $@MAC_{dst}$ = broadcast and $@MAC_{src}$ = $@MAC_{virtual\text{-}ID=1}$ → all hosts will refresh their ARP table with the MAC that R_2 wants to associate to the $@IP_A$,

Note that VRRP uses the virtual MAC associated to the VRRP-ID for which the router is master, and does not use the real physical MAC address.

- **VRRP + STP + VLAN (CISCO IoS): 1 VLAN without load balancing**



- **VRRP + STP + VLAN (CISCO IoS): 1 VLAN with load balancing**

!!!! Router R1

R1(conf)# **interface** Ge0

R1(config-if)# **ip address** 10.1.1.1 255.255.255.0

R1(config-if)# **vrrp** 1 **priority** 200

R1(config-if)# **vrrp** 1 **ip** 10.1.1.10

R1(config-if)# **no shutdown**

!!!! Router R2

R2(conf)# **interface** Ge0

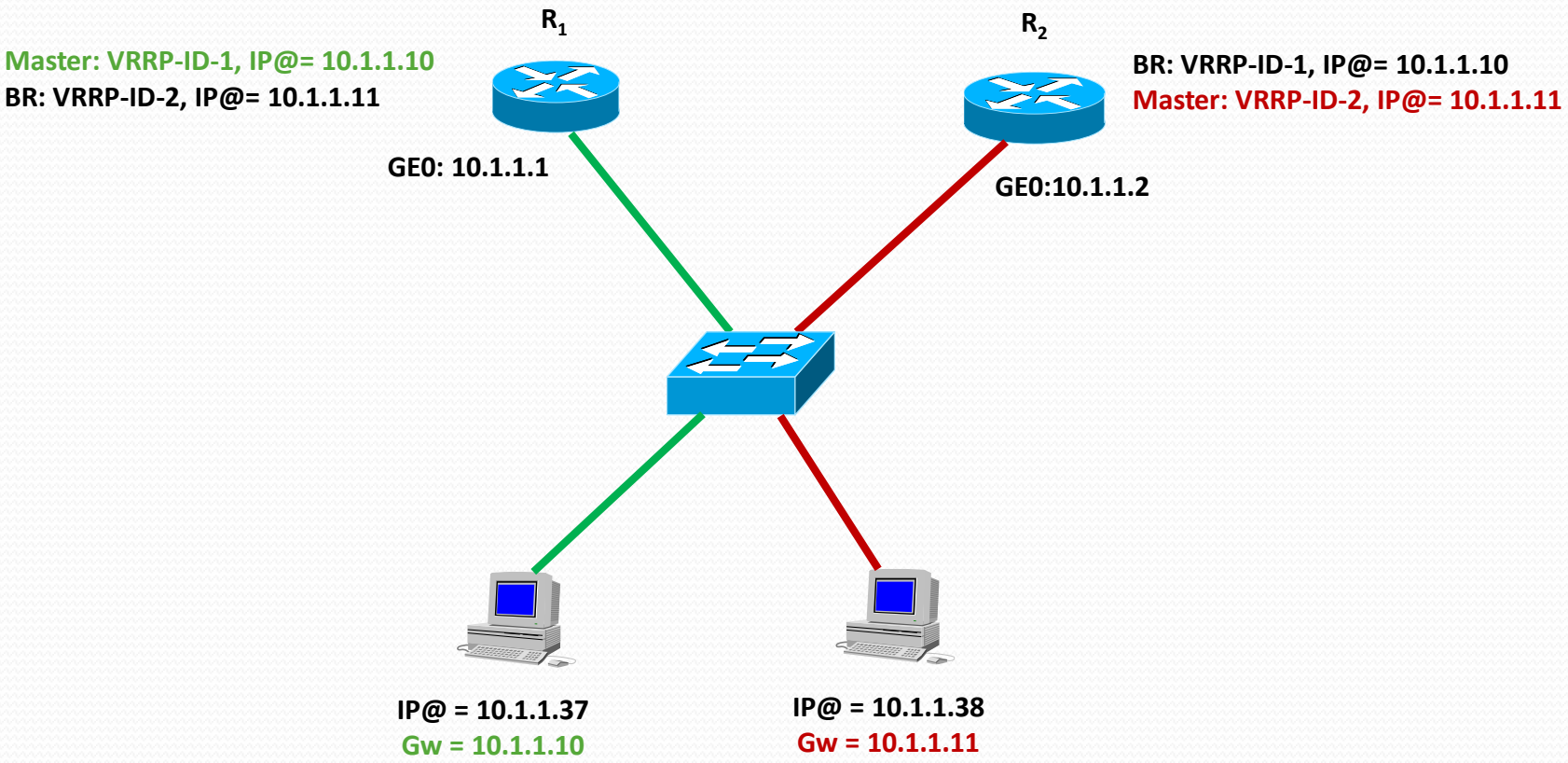
R2(config-if)# **ip address** 10.1.1.2 255.255.255.0

R2(config-if)# **vrrp** 1 **priority** 100

R2(config-if)# **vrrp** 1 **ip** 10.1.1.10

R2(config-if)# **no shutdown**

- **VRRP + STP + VLAN (CISCO IoS): 1 VLAN with load balancing**



- **VRRP + STP + VLAN (CISCO IoS): 1 VLAN with load balancing**

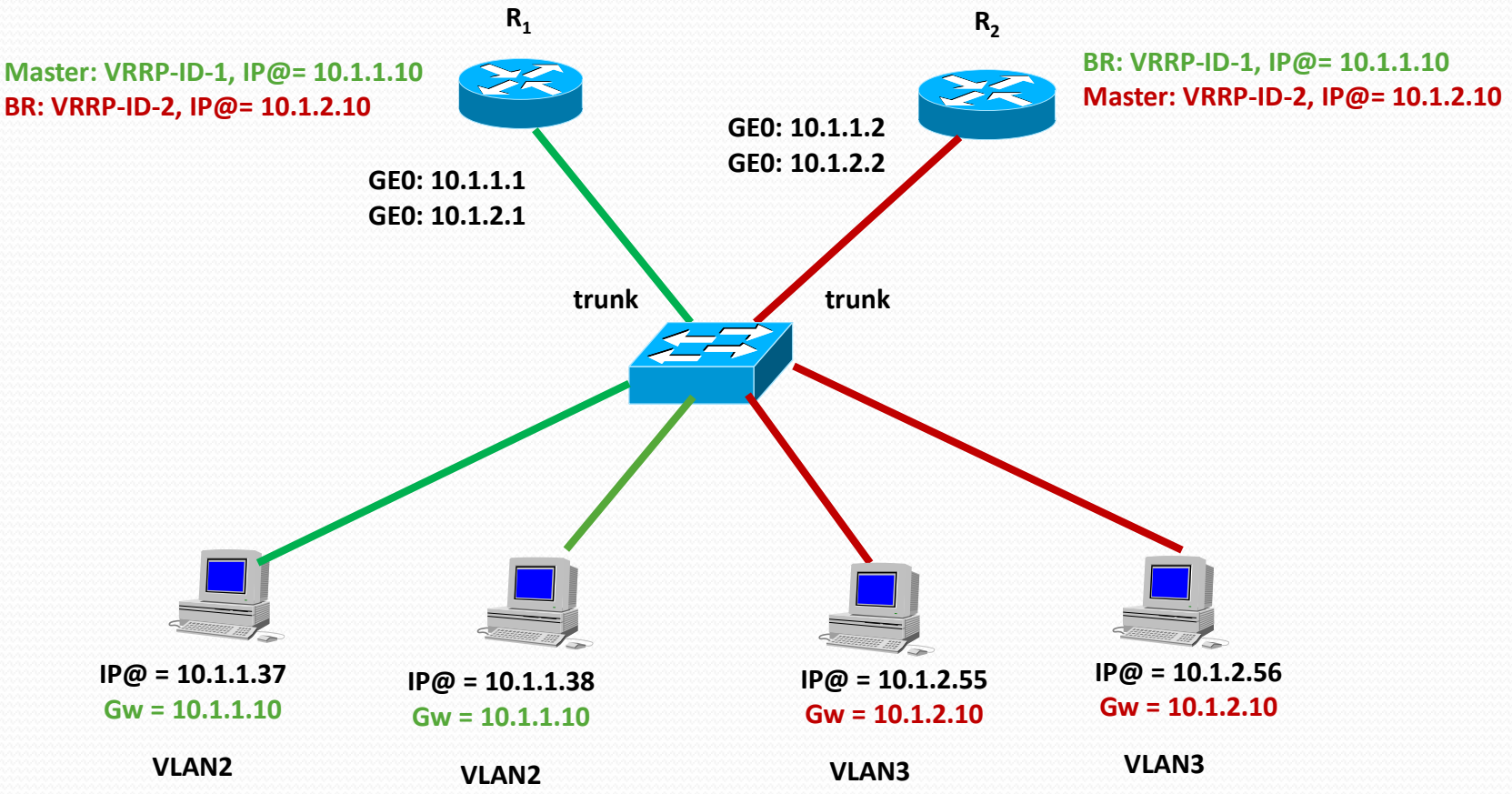
!!!! Router R1

R1(conf)# **interface** Ge0
R1(config-if)# **ip address** 10.1.1.1 255.255.255.0
R1(config-if)# **vrrp** 1 **priority** 200
R1(config-if)# **vrrp** 1 **ip** 10.1.1.10
R1(config-if)# **vrrp** 2 **priority** 100
R1(config-if)# **vrrp** 2 **ip** 10.1.1.11
R1(config-if)# **no shutdown**

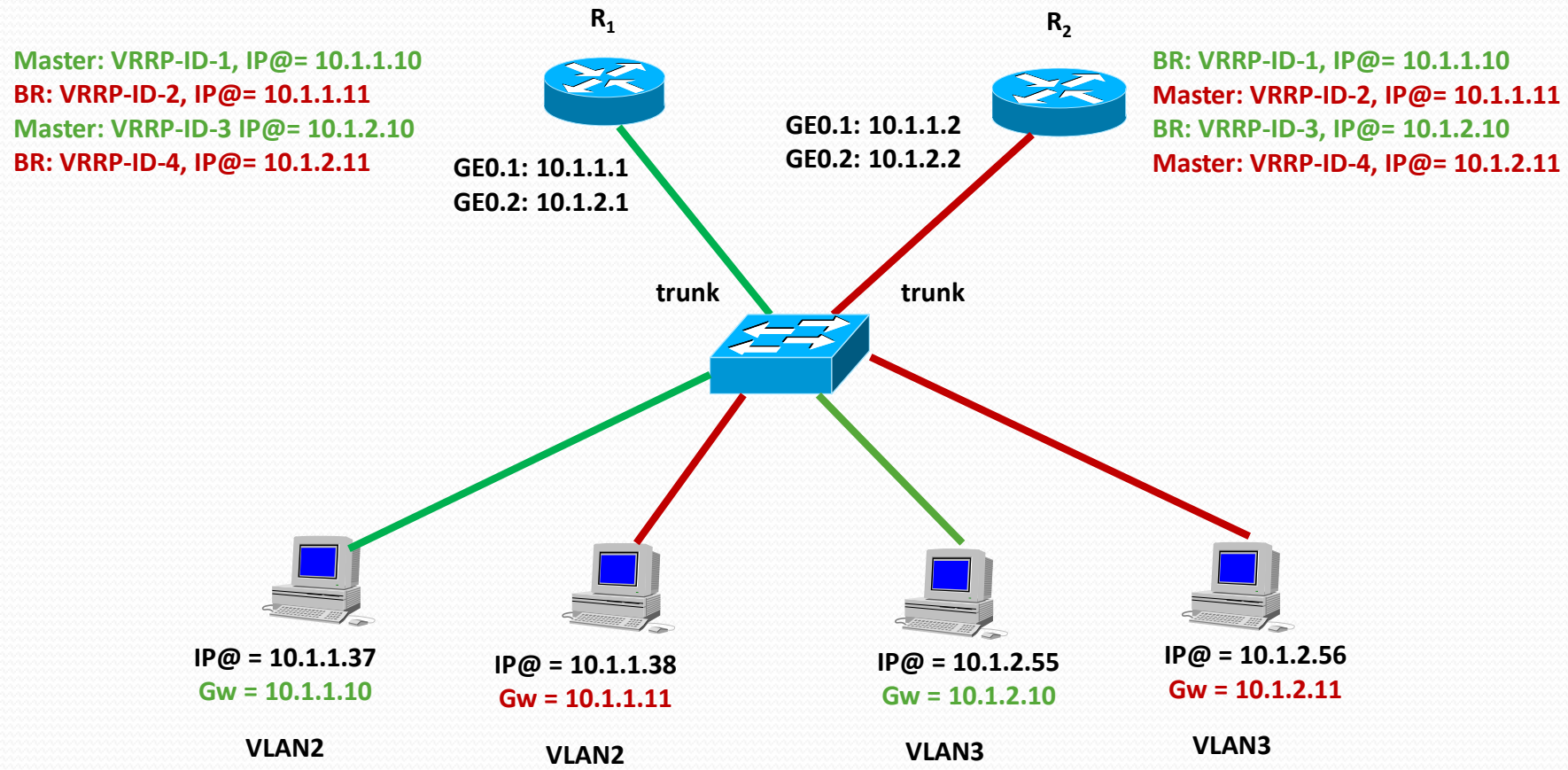
!!!! Router R2

R2(conf)# **interface** Ge0
R2(config-if)# **ip address** 10.1.1.2 255.255.255.0
R2(config-if)# **vrrp** 1 **priority** 100
R2(config-if)# **vrrp** 1 **ip** 10.1.1.10
R2(config-if)# **vrrp** 2 **priority** 200
R2(config-if)# **vrrp** 2 **ip** 10.1.1.11
R2(config-if)# **no shutdown**

- **VRRP + MSTP + VLAN (CISCO IoS): 2 VLAN with load balancing**
  - Other possibility is to balance using VLANs; all VLAN2 hosts have R_1 as gateway, and all VLAN3 hosts have R_2. Solve the code as exercise.



- **VRRP + MSTP + VLAN (CISCO IoS): 2 VLAN with load balancing**
  - In this example, we balance in the following way; half of the VLAN2 hosts use R_1 as gateway and backup, and the other half use R_2 as gateway, idem with hosts of VLAN3,



- **VRRP + MSTP + VLAN (CISCO IoS): 2 VLAN with load balancing**

!!!! Router R1, is a trunk port

R1(conf)# **interface** Ge0

R1(config-if)# **no shutdown**

!!!! Subinterface (Virtual) of Ge0 associated to VLAN 2 → Ge0.1

R1(conf)# **interface** Ge0.1

R1(config-if)# **ip address** 10.1.1.1 255.255.255.0

R1(config-if)# **encapsulation** dot1q VLAN2

R1(config-if)# **vrrp** 1 **priority** 200

R1(config-if)# **vrrp** 1 **ip** 10.1.1.10

R1(config-if)# **vrrp** 2 **priority** 100

R1(config-if)# **vrrp** 2 **ip** 10.1.1.11

!!!! Subinterface (Virtual) of Ge0 associated to VLAN 3 → Ge0.2

R1(conf)# **interface** Ge0.2

R1(config-if)# **ip address** 10.1.2.1 255.255.255.0

R1(config-if)# **encapsulation** dot1q VLAN3

R1(config-if)# **vrrp** 3 **priority** 200

R1(config-if)# **vrrp** 3 **ip** 10.1.2.10

R1(config-if)# **vrrp** 4 **priority** 100

R1(config-if)# **vrrp** 4 **ip** 10.1.2.11

- **VRRP + MSTP + VLAN (CISCO IoS): 2 VLAN with load balancing**

!!!! Router R2, is a trunk port

R2(conf)# **interface** Ge0

R2(config-if)# **no shutdown**

!!!! Subinterface (Virtual) of Ge0 associated to VLAN 2 → Ge0.1

R2(conf)# **interface** Ge0.1

R2(config-if)# **ip address** 10.1.1.2 255.255.255.0

R2(config-if)# **encapsulation** dot1q VLAN2

R2(config-if)# **vrrp** 1 **priority** 100

R2(config-if)# **vrrp** 1 **ip** 10.1.1.10

R2(config-if)# **vrrp** 2 **priority** 200

R2(config-if)# **vrrp** 2 **ip** 10.1.1.11

!!!! Subinterface (Virtual) of Ge0 associated to VLAN 3 → Ge0.2

R2(conf)# **interface** Ge0.2

R2(config-if)# **ip address** 10.1.2.2 255.255.255.0

R2(config-if)# **encapsulation** dot1q VLAN3

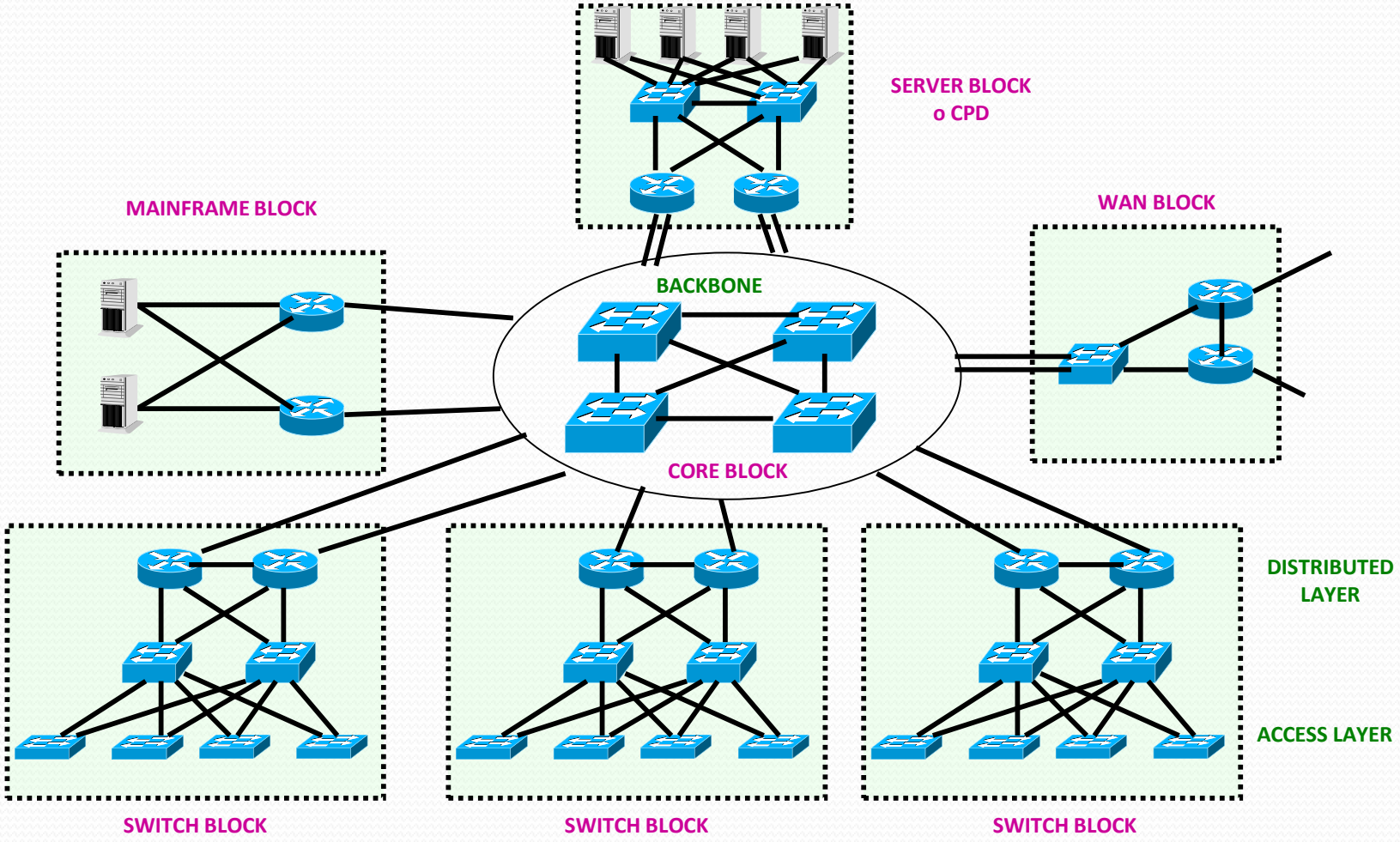
R2(config-if)# **vrrp** 3 **priority** 100

R2(config-if)# **vrrp** 3 **ip** 10.1.2.10

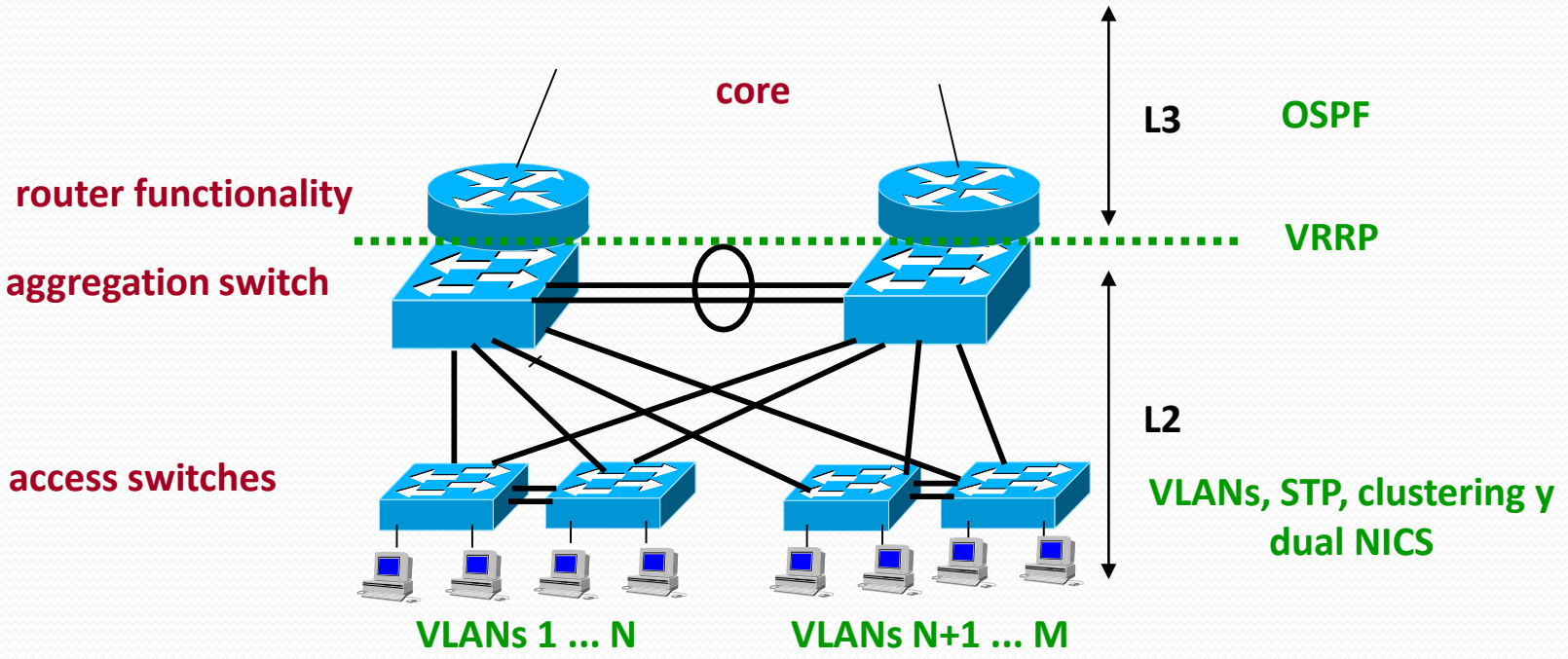
R2(config-if)# **vrrp** 4 **priority** 200

R2(config-if)# **vrrp** 4 **ip** 10.1.2.11

- Corporate Network architecture: the switching blocks

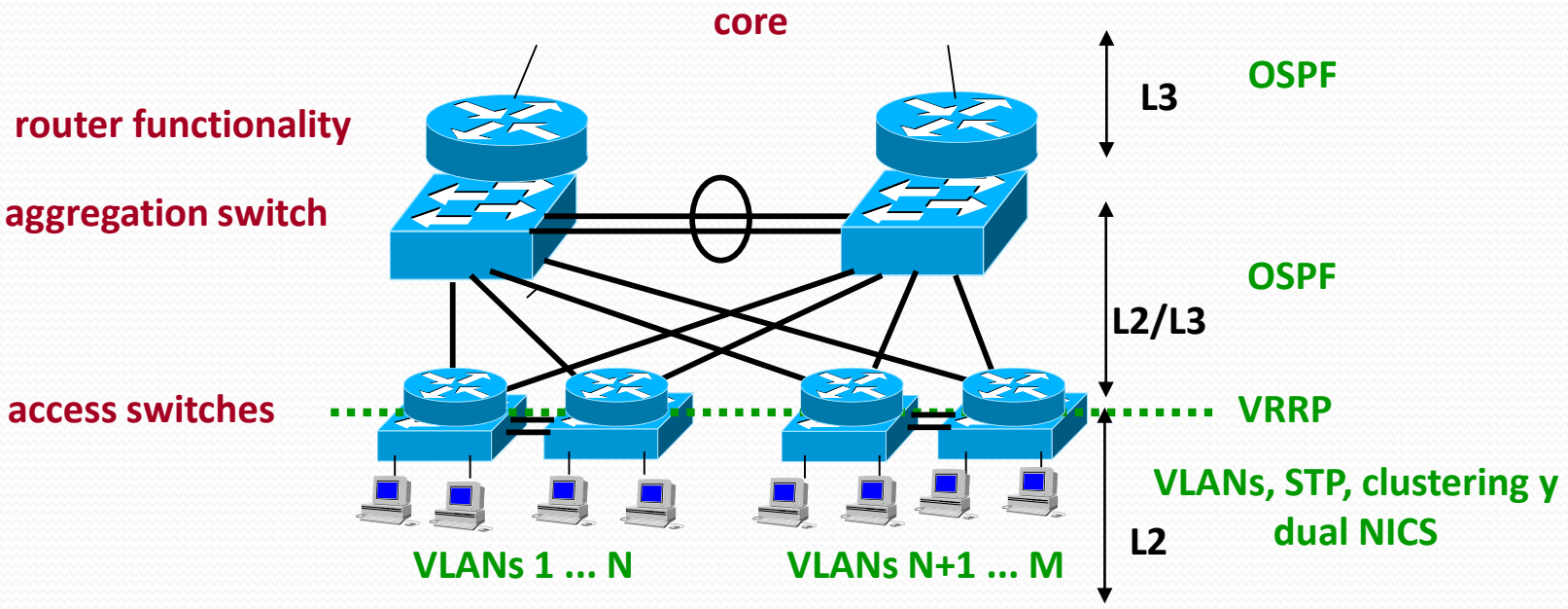


- **Switched block and Data Center**
  - Aggregation switch→ can be Multilayer Switch with routing functionalities (firewalling, load balancing, fail tolerance, ...)
  - L2 server adjacency → needed to exchange session information, synchronization, ...



Access Switches→ e.g., Multilayer Switch

- Avoid STP blocking when the aggregation modules are reached since there is a router that isolates the access to aggregation
- Limit broadcasts and improves convergence latencies
- clustering and NIC teaming restricted to groups of switches (need L2 adjacency for synchronization purposes)



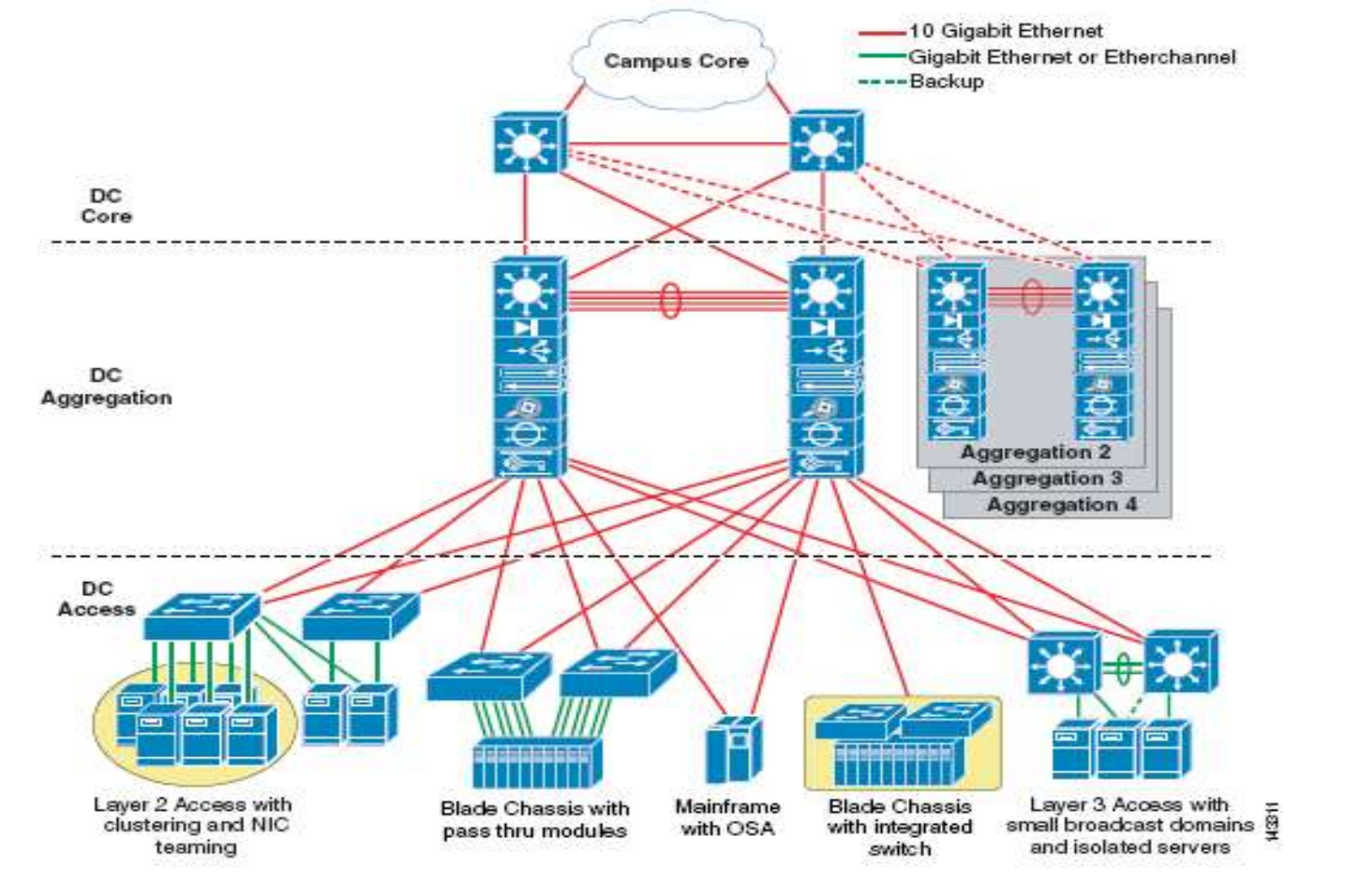
Data Processing Centers (CPD)

- A **data center** is a facility used to house computer systems and associated components, such as telecommunications and storage systems.
- It generally **includes** redundant or backup power supplies, redundant data communications connections, environmental controls (e.g., air conditioning, fire suppression) and security devices:
  - Operate and manage a carrier's telecommunication network,
  - Provide data center based applications directly to the carrier's customers,
  - Provide hosted applications for a third party to provide services to their customers,
  - Provide a combination of these and similar data center applications,
  - **Lista de CPD's en España por ciudad** (no incluye los CPD's propios de las redes corporativas, solo los que pertenecen a empresas que dan servicios a terceros): http://www.centrodedatos-datacenter.es,
  - En https://www.datacentermap.com/spain/ teneis el número por ciudad, e.g., BCN hay 15, Madrid 21, Valecia 6, Sevilla 2, Bilbao 1, Girona 1,

Data Processing Centers (CPD)

Tier Level	Requirements
1	•Single non-redundant distribution path serving the IT equipment •Non-redundant capacity components •Basic site infrastructure with expected availability of 99.671% (fail less than 4’44s per day)
2	•Meets or exceeds all Tier 1 requirements •Redundant site infrastructure capacity components with expected availability of 99.741% (fail less than 3’43s per day)
3	•Meets or exceeds all Tier 1 and Tier 2 requirements •Multiple independent distribution paths serving the IT equipment •All IT equipment must be dual-powered and fully compatible with the topology of a site's architecture •Concurrently maintainable site infrastructure with expected availability of 99.982% (fail less than 15s per day)
4	•Meets or exceeds all Tier 1, Tier 2 and Tier 3 requirements •All cooling equipment is independently dual-powered, including chillers and heating, ventilating and air-conditioning (HVAC) systems •Fault-tolerant site infrastructure with electrical power storage and distribution facilities with expected availability of 99.995% (fail less than 4s per day)

- **Data Processing Centers (CPD)**



- **High Availability (HA)**
  - Applications, network equipment (servers, routers, switches, ...) and network interfaces can fail→ there is a need to improve fail tolerance
    - Applications: automatically (script) to re-initiate processes (SO)
    - L2: Spannning Tree Protocol handle fail tolerance at L2
    - L3: VRRP and OSPF handle fail tolerance at L3
    - Equipment: improve server performance (e.g. using clustering)
      - Clustering for High Availability: group of computers that support server applications that can be reliably utilized in a minimum of down-time.
      - Dual connections: use of more than one network interface card (NIC) in Servers.

- **Scalability**
  - React to company growth → implies a growth in the following points:
    - Growth in the number of network connections
    - Growth in the capacity of the network

Both impact the network infrastructure in terms of equipment (switches, routers, …)

  - Growth in the computation capabilities: use of **clusters** in order to increase
    - the server capacity (e.g. clustering for Load Balancing)
    - the computational capacity (e.g. clustering for Computational power), and
    - reliability

- **Data Processing Centers (CPD): cluster servers (farm servers)**
  - Main objective: execute multiple application in multiple machines
    - Clustering techniques allow dispatch queries to those servers that are more reliable or unloaded (fail tolerance)

Server 1	**Application 1**	**Application 2**
Server 2	**Application 1**	**Application 2**
Server N	**Application 1**	**Application 2**

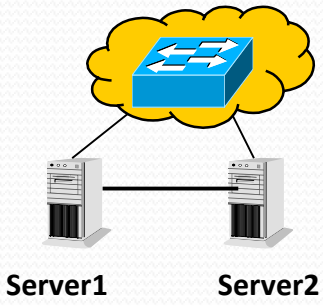
VIPA (Virtual IP address) → allows redirect a query to a set of servers (in fact it is the IP address of the "dispatcher" that receives the client query)

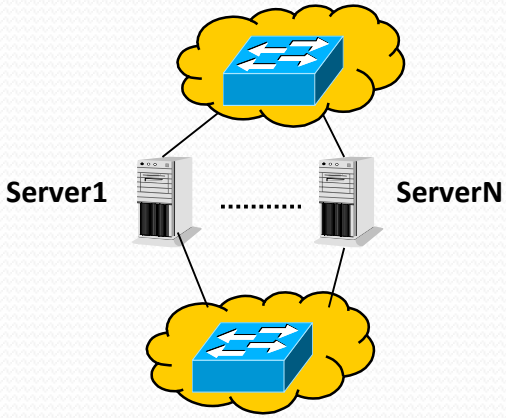
Distribute client queries to servers → linked to load balance techniques

Backup of the dispatcher !!! It can also fail

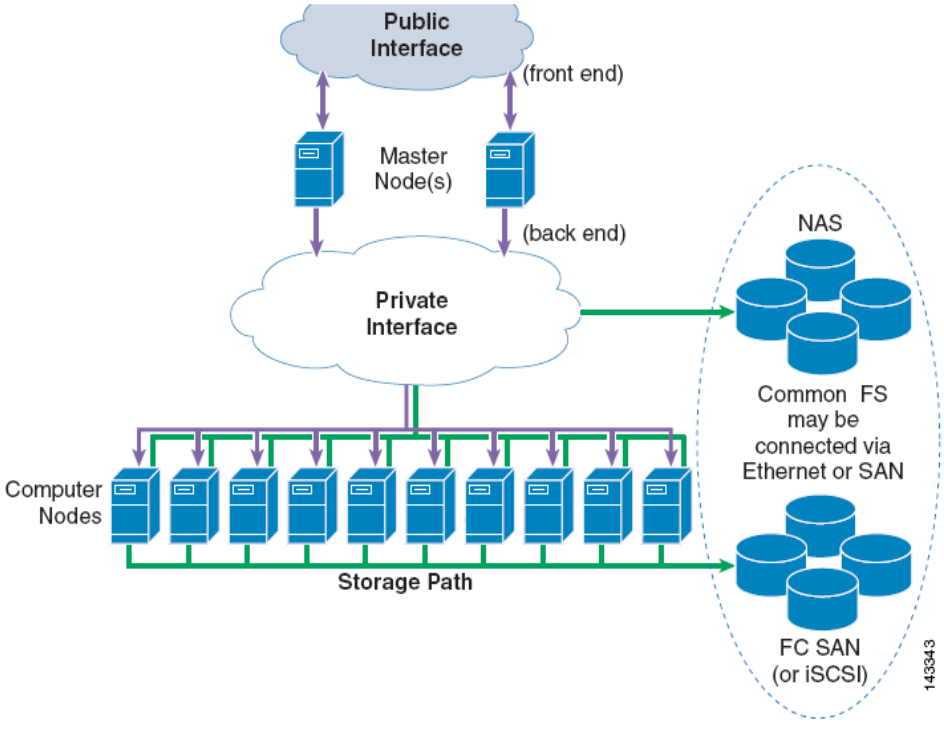
- **Data Processing Centers (CPD): cluster servers (farm servers)**
  - If two servers:
    - They can be interconnected (e.g. via a crossover wire) in order to exchange information (e.g. exchange of data, session state or monitoring states).
  - If more than two servers
    - Interconnected via LAN

Require adjacency at L2





Data Processing Centers (CPD): cluster servers (farm servers)



Front end: interfaces with external access to the cluster (e.g. Application servers or users that send jobs to be executed in the cluster)

Master Nodes: responsible of managing switched nodes in the cluster and to optimize the computing capacity

Back-end high speed fabric: the media that uses the master to communicate with the computation nodes (low-latency and high bandwidth). Typically 10GigaE or Infiniband

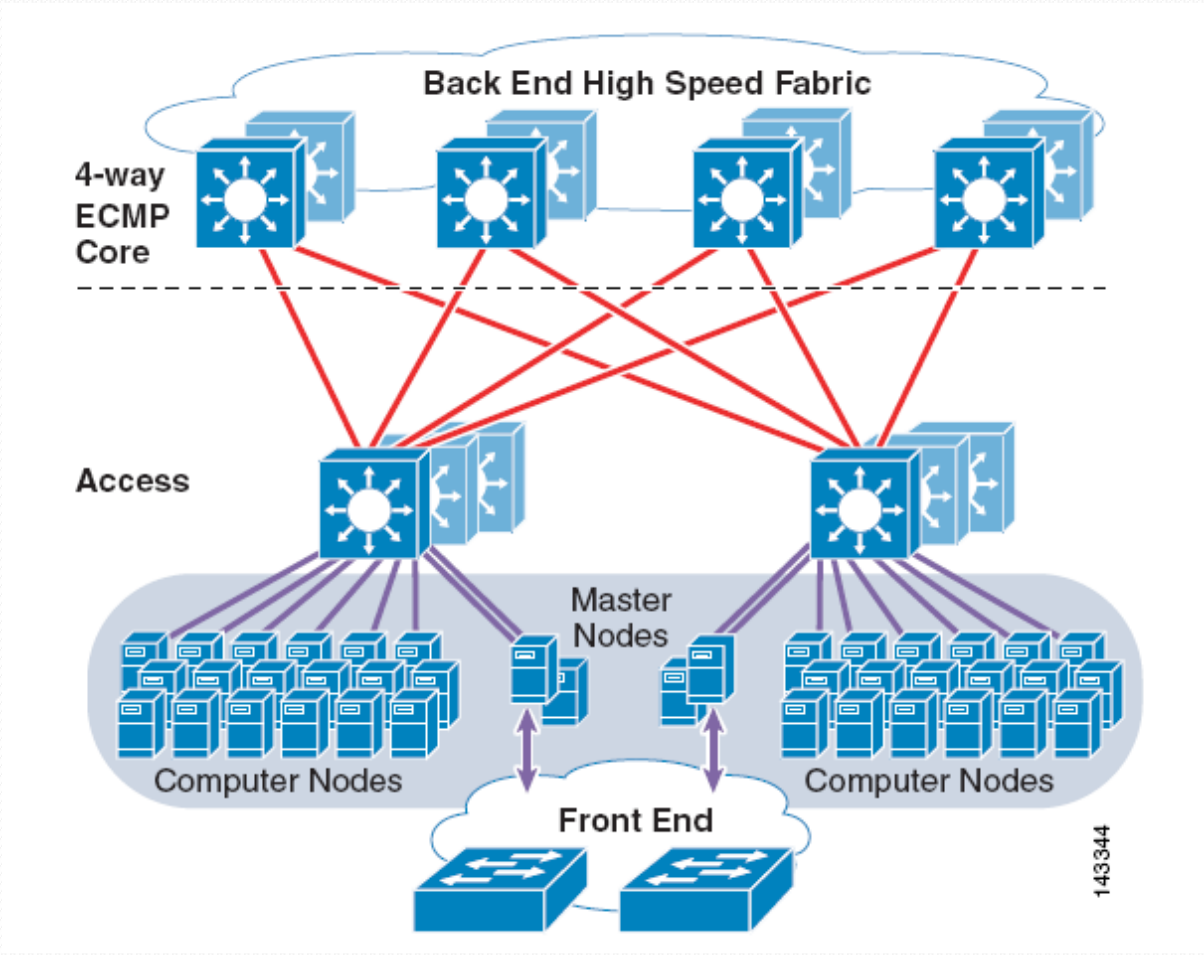
Computer Nodes: Computation nodes with an OS responsible of intensive operations

Storage Path: Ethernet or Fibre Channel for connecting with the storage capabilities (SAN: Storage Area Network)

Common parallel FS (File System) to access all computation nodes

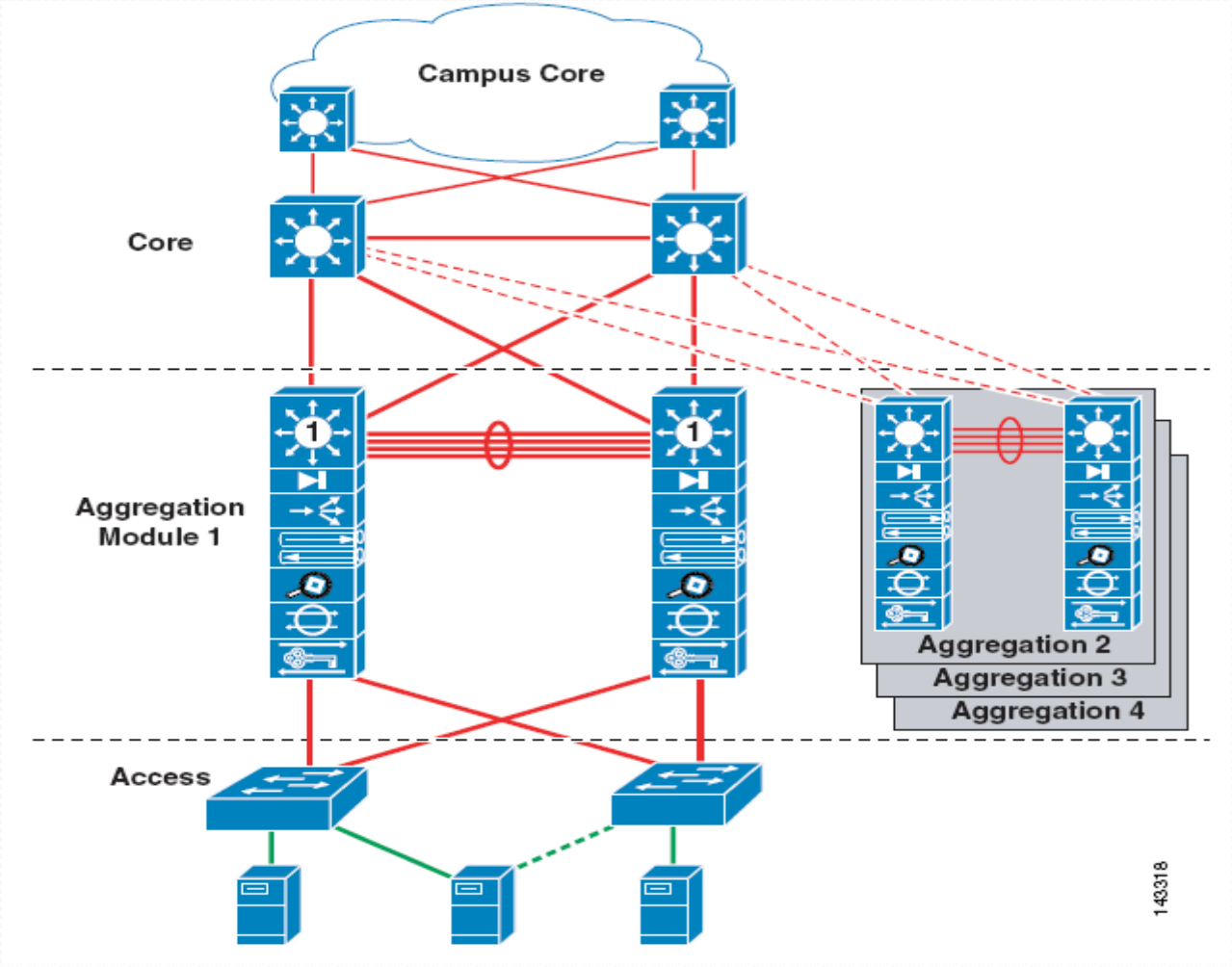
- **Data Processing Centers (CPD): cluster servers (farm servers)**
  - **Load Balancing**
    - If there are several lines that interconnect the servers, we can distribute the **load** is such a way that this tends to be more **symmetric** (have equivalent loads in all servers)
    - Load Distribution
      - As a function of the knowledge that the dispatcher has on the work-load of the systems
      - If the dispatchers have no information related to the servers, they can distribute the queries different policies:
        - distribute via round-robin,
        - distribute as a function of the number of on-going queries,
        - others ...

- **Data Processing Centers (CPD): cluster servers (farm servers)**



Multi-tier Data Centers:

- Most common DC design technique in many companies



Data Center Core Layer

- Connects the Core Distribution (backbone) Campus layer with the Data Center Aggregation layer,
- Not always required, some designs don't have a core layer,
- Gives high speed connection towards the aggregation modules,
- If the core is independent of the Core Campus (backbone) layer, is allowed to implement independent policies (e.g. QoS, Access list, maintenance, ...) in the Data Center and in the Campus backbone,
- Uses 10 GEth ports (both to the backbone and to the aggregation layer).

Data Center Aggregation Layer

- Aggregate thousands of connections that want to access the Data Center
- The aggregation switches must be capable of supporting many 10 GigE and GigE interconnects while providing a high-speed switching fabric with a high forwarding rate.
  - Support of 10 GEth ports (towards the core) and 1GEth ports (towards access)
- The aggregation layer switches carry the workload (Root Bridge) of spanning tree processing, and default gateway redundancy protocol processing (VRRP) if L3 switches are used.

- **Data Center Aggregation Layer**
  - Support high value services such as Load balancing, Firewalling and Intrusion Detection, SSL (Secure Socket Layer) to the servers, caches, network monitoring, etc,
    - High port densities
    - 10Gbps ports
    - VLAN
    - 802.1s (MSTP), 802.1w (rapid STP)
    - MPLS-VPN
    - Hardware-based NAT
    - QoS
    - Load Balancing and security modules

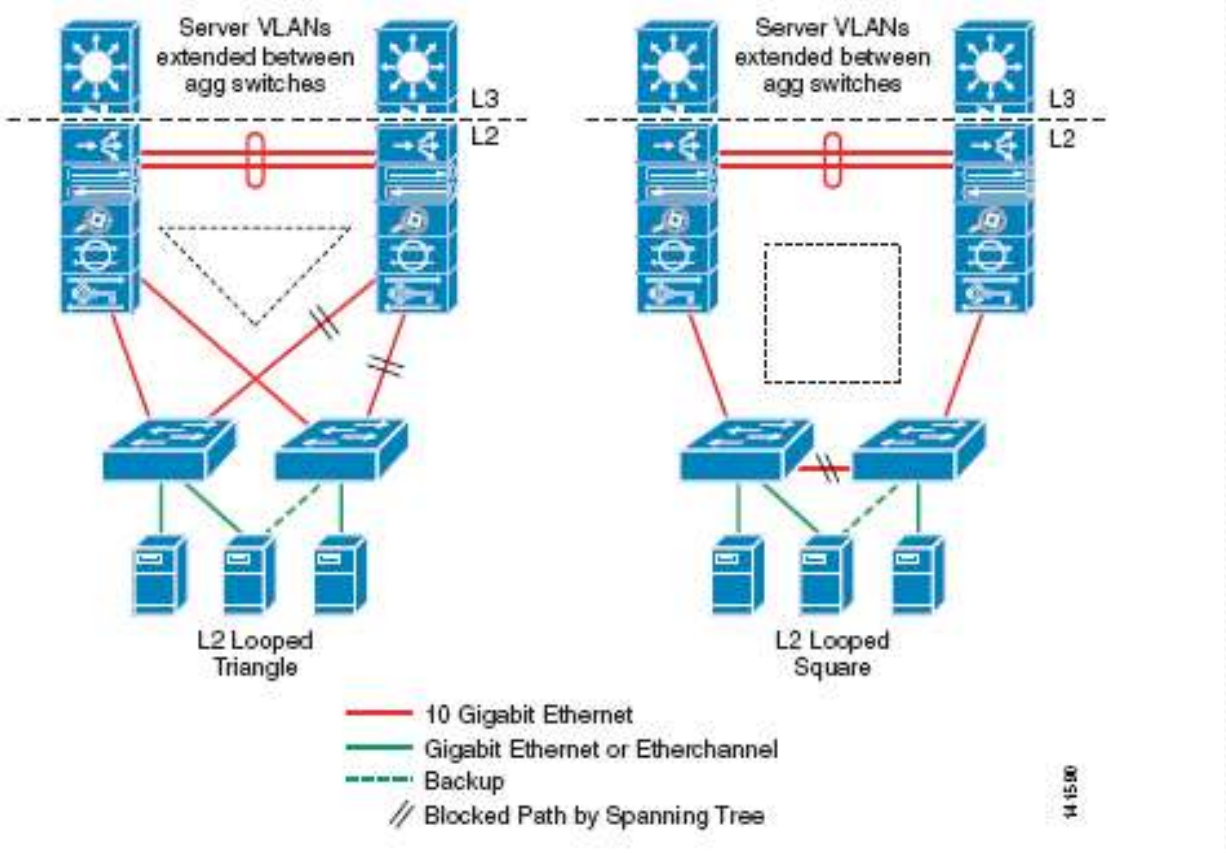
Data Center Access Layer

- Gives a connection point to the servers and operates at L2 and L3
- The mode plays a critical role in meeting particular server requirements such as NIC teaming, clustering, and broadcast containment.
- The access layer is the first oversubscription point in the data center because it aggregates the server traffic onto Gigabit EtherChannel or 10 GigE/10 Gigabit EtherChannel uplinks to the aggregation layer.

- **Data Center Access Layer**
  - Always Redundancy: at least a couple of switches inter-connected with STP
    - see following slides for examples
    - **Looped** configurations such as triangle and square or **looped-free** such as U or U-down
      - Triangle/square configurations usually the best one but needs a lot of experience,
      - U's configurations are simple and used when low STP experience or because STP is undesired (e.g. all uplink links active) → in general use STP

- **Data Center Access Layer**
  - **Looped** configurations are desirable because:
    - **VLAN extension**: is a key requirement in most data centers. The ability to add servers into a specific VLAN across the entire access layer,
    - **Resiliency/robustness:** Looped topologies are inherently redundant,
    - **Service module interoperability**: Service modules operating in active-standby modes require Layer 2 adjacency between their interfaces,
    - Server requirements for Layer 2 adjacency in support of NIC teaming and high availability clustering.

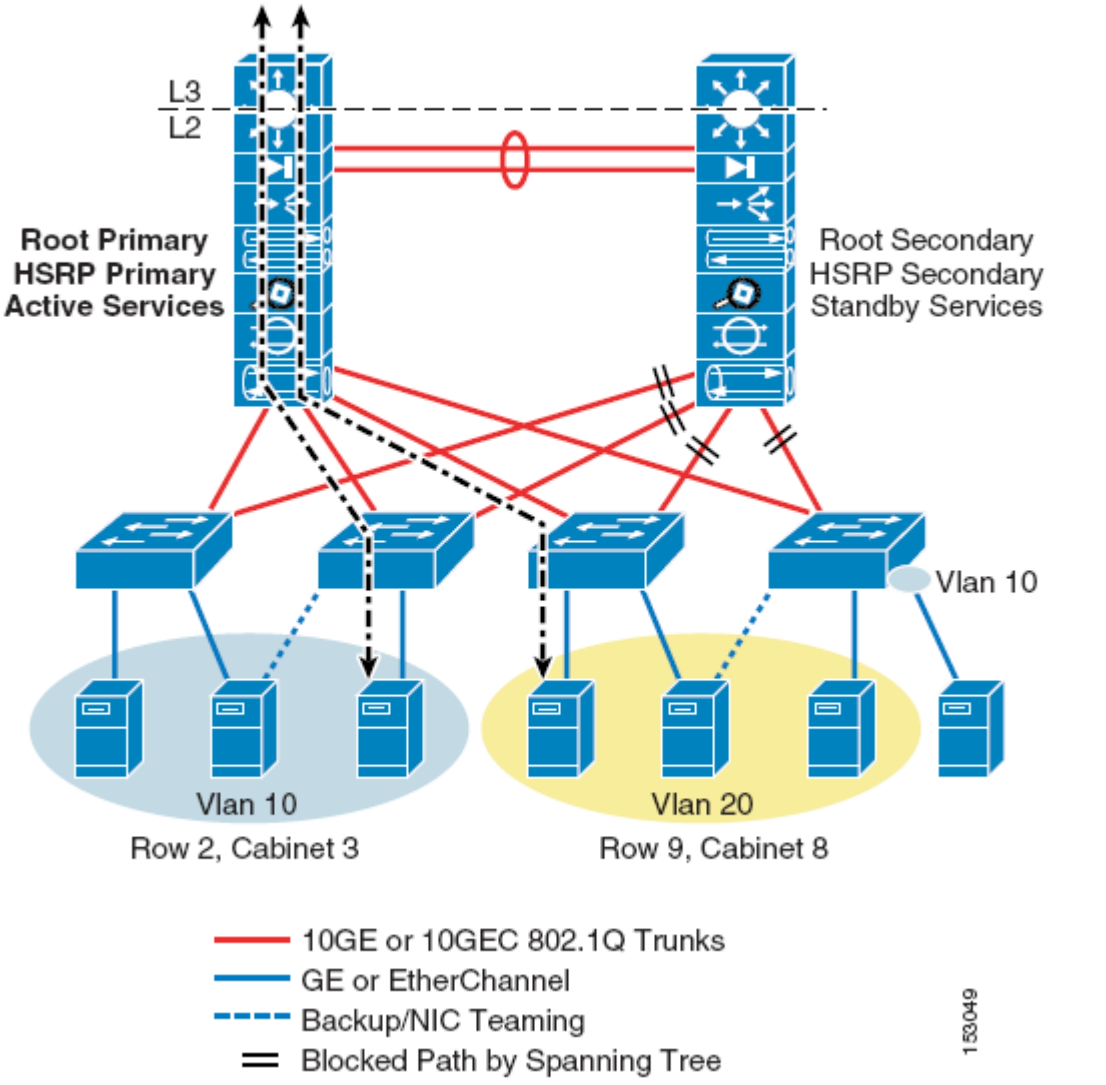
- **Data Center Access Layer**



Access Topologies (Looped) using L2 technologies

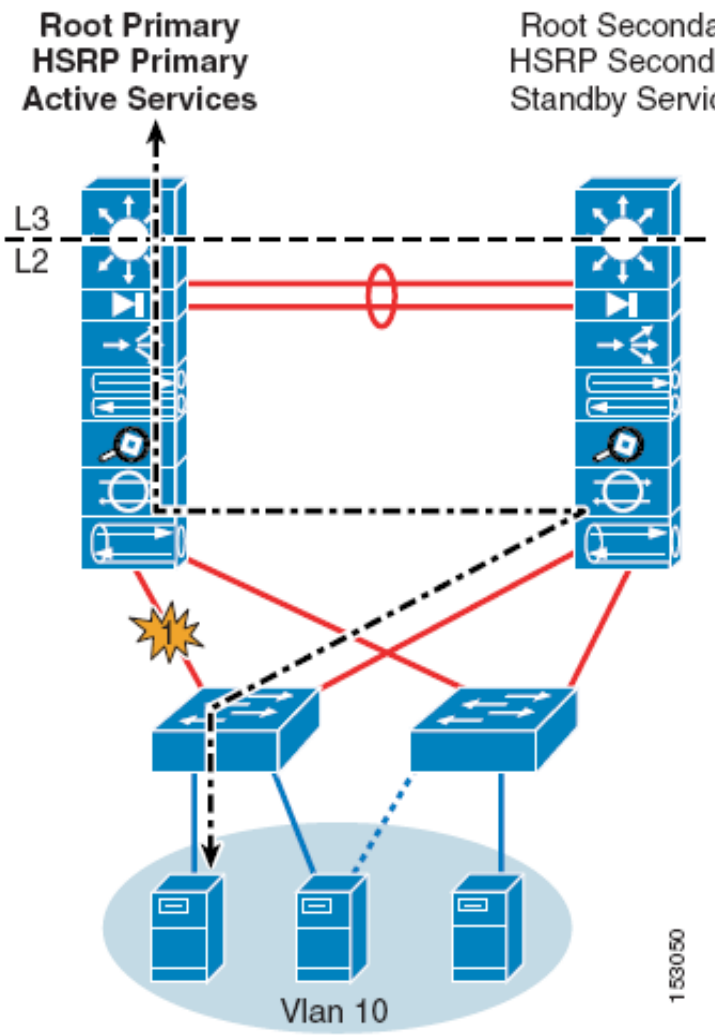
- **Data Center Access Layer:**
  - Triangle looped topology + VRRP (or HSRP)

Figure 6-4 Triangle Looped Access Topology



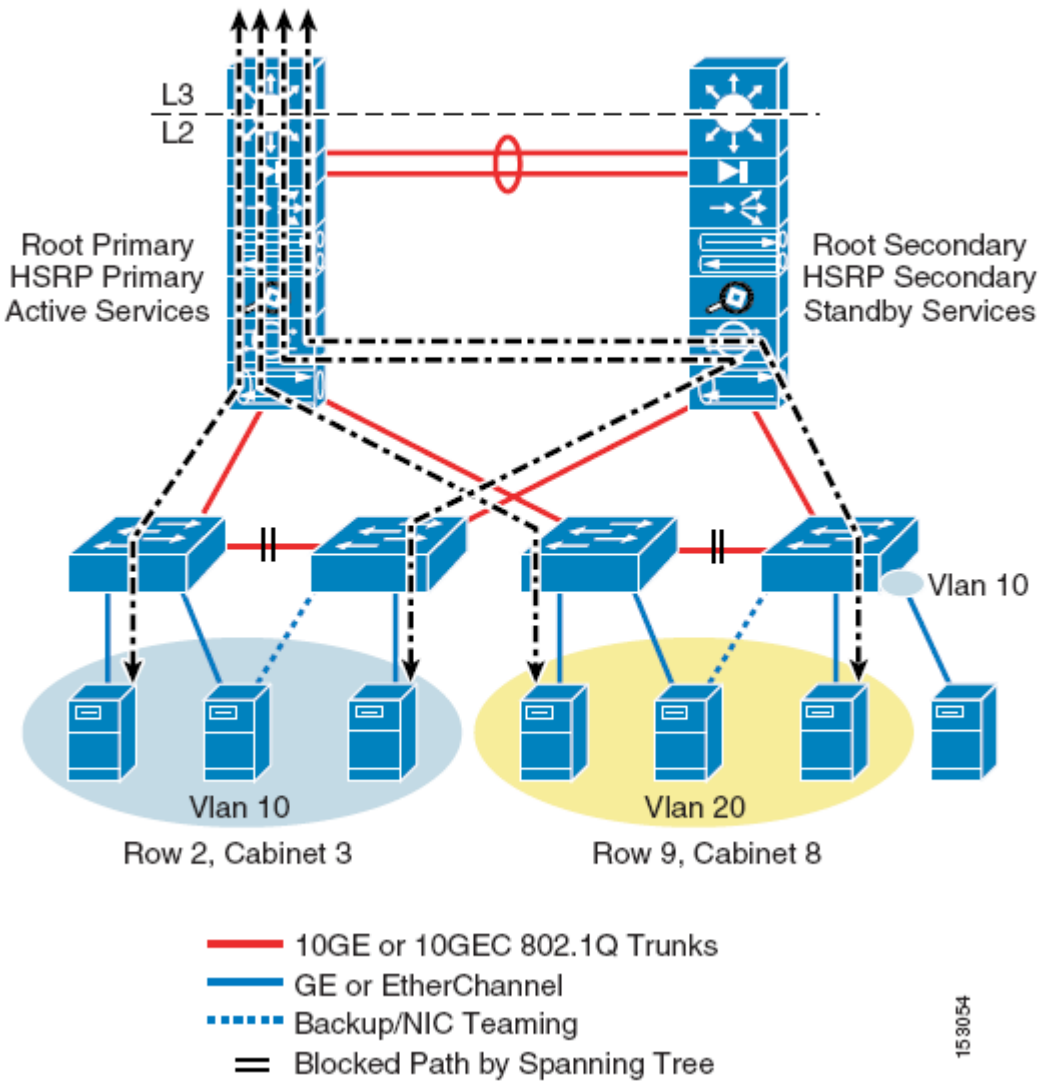
- **Data Center Access Layer:**
  - Triangle looped topology + VRRP (or HSRP)

Figure 6-5 Triangle Looped Failure Scenario 1—Uplink Down



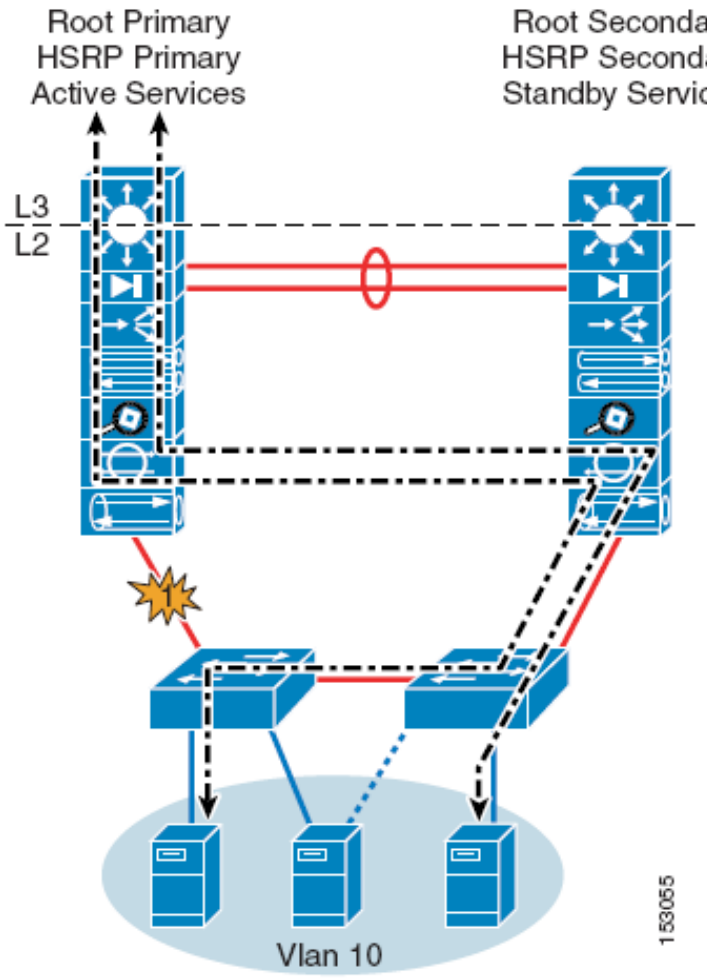
- **Data Center Access Layer:**
  - Square looped topology + VRRP (or HSRP)

Figure 6-9 Square Looped Access Topology

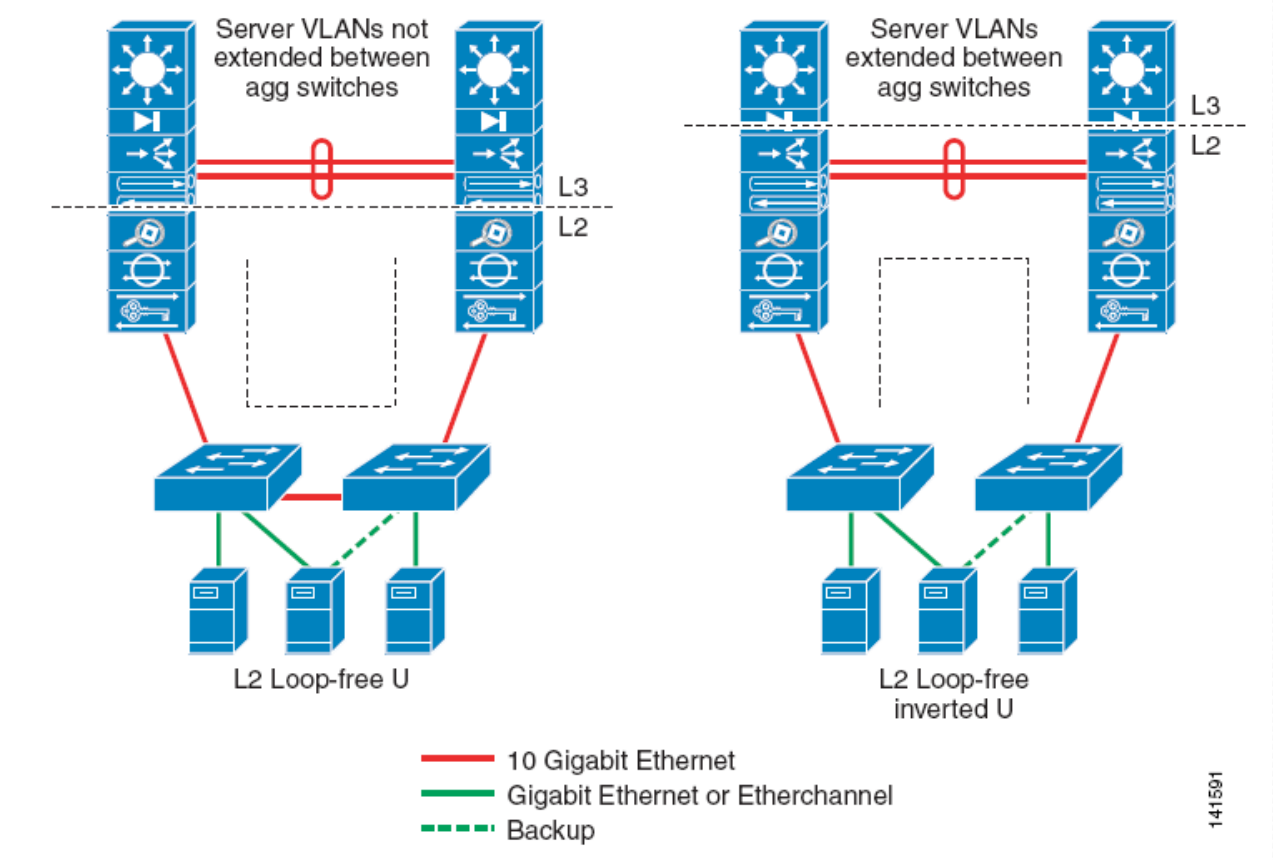


- **Data Center Access Layer:**
  - Square looped topology + VRRP (or HSRP)

Figure 6-10 Square Looped Failure Scenario 1—Uplink Down



- **Data Center Access Layer**



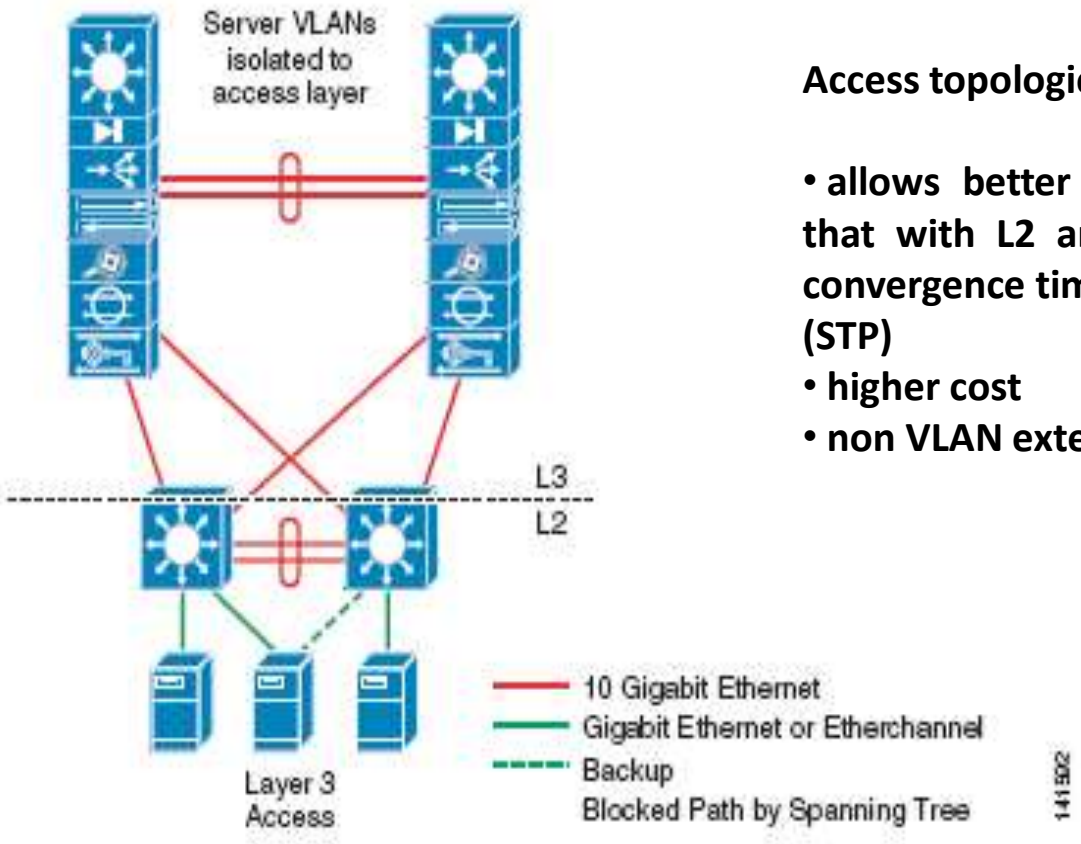
Access Topologies Looped-free using L2 technologies

- **Data Center Access Layer**
  - Main differences between a **Looped** and a **looped-free** configuration:
    - No blocking on uplinks, all links are active in a loop-free topology
    - Layer 2 adjacency for servers is limited to a single pair of access switches in a loop-free topology
    - VLAN extension across the data center is not supported in a loop-free U topology but is supported in the inverted U topology.
  - In any case, even in loop-free topologies, activate STP to prevent loops

Exercise

- Draw an scheme showing whether triangle, square, U and U-down provide VLAN extension.

- **Data Center Access Layer**



Access topologies using L3 technology:

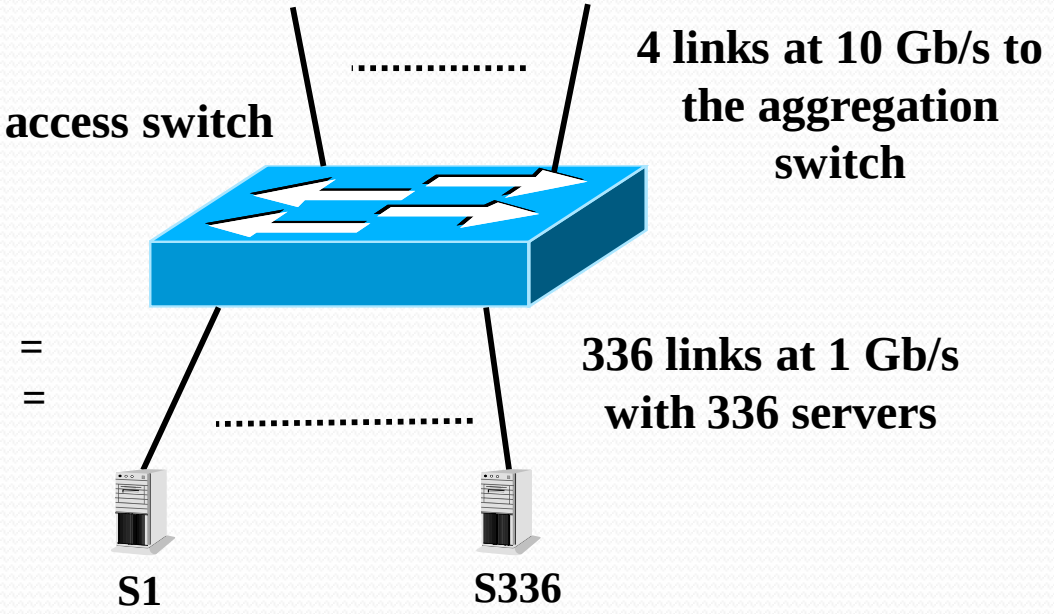
• allows better control (isolate the servers) that with L2 and L3 access and has better convergence times (e.g. OSPF) with respect L2 (STP)
• higher cost
• non VLAN extensions across the data center

- # Data Center Access Layer
  - Scalability: number of servers versus switches
    - **Average bandwidth (or Throughput) per server:** the OS (operator system) and NIC (Network Interface Card) is able to produce traffic that occupies a % of the link capacity. E.g a server occupies a 60% of the 1 Gb/s link → occupies 600 Mb/s,
    - **Oversupscription ratio per server:** average number of servers to occupy a link capacity. E.g. if a sever occupies 60% of the link, then 1/0.6= 1.666 → oversubscription ratio is of **1.66:1,**
    - **Oversupscription ratio of a switch:** average number of servers to occupy the uplink capacity of a switch.

      We have to take into account all the access links connected to the servers and all the uplink aggregated links towards the aggregation switch.

Data Center Access Layer

- What happens if we have several links (with servers) in the access switch and several links that uplink from the access to the aggregation switch ? See the following example where we have 336 servers connected at 1 Gb/s ports in an access switch, and this switch is connected using 4 ports at 10 Gb/s each to the aggregation switch



$Throughput_{Server}$ = 4*10 Gb/s /336 =
= 40/336 = 0.12 Gb/s

Oversupscription= 1/ $Throughput_{Server}$ =
= 336 servers at 1 Gb/s / (4*10 Gb/s) =
= 336 / 40 = 8.4 → 8.4:1

- **Data Center Access Layer**
  - Scalability: number of servers versus switches
    - Maxim number of 1GEth server connections: scale with switches. Increase servers and switches in such a way that a minimum bandwidth and maximum latency is guaranteed
    - Approximate bandwidth per server: if N 10 GEth ports towards aggregation and M server ports → Nx10 GEth/M = C Mbps per server (e.g. 4x10Gbps/336 servers = 120 Mbps
    - Oversupscription ratio per server: divide the number of server connections by the access aggregate. E.g. 336 server connections with 4x10GEth → 8.4:1 (1Gbps/120Mbps=8.4)

N·(K Gb/s) ≤ M·(R Gb/s) →Thrput = N·(K Gb/s) / M

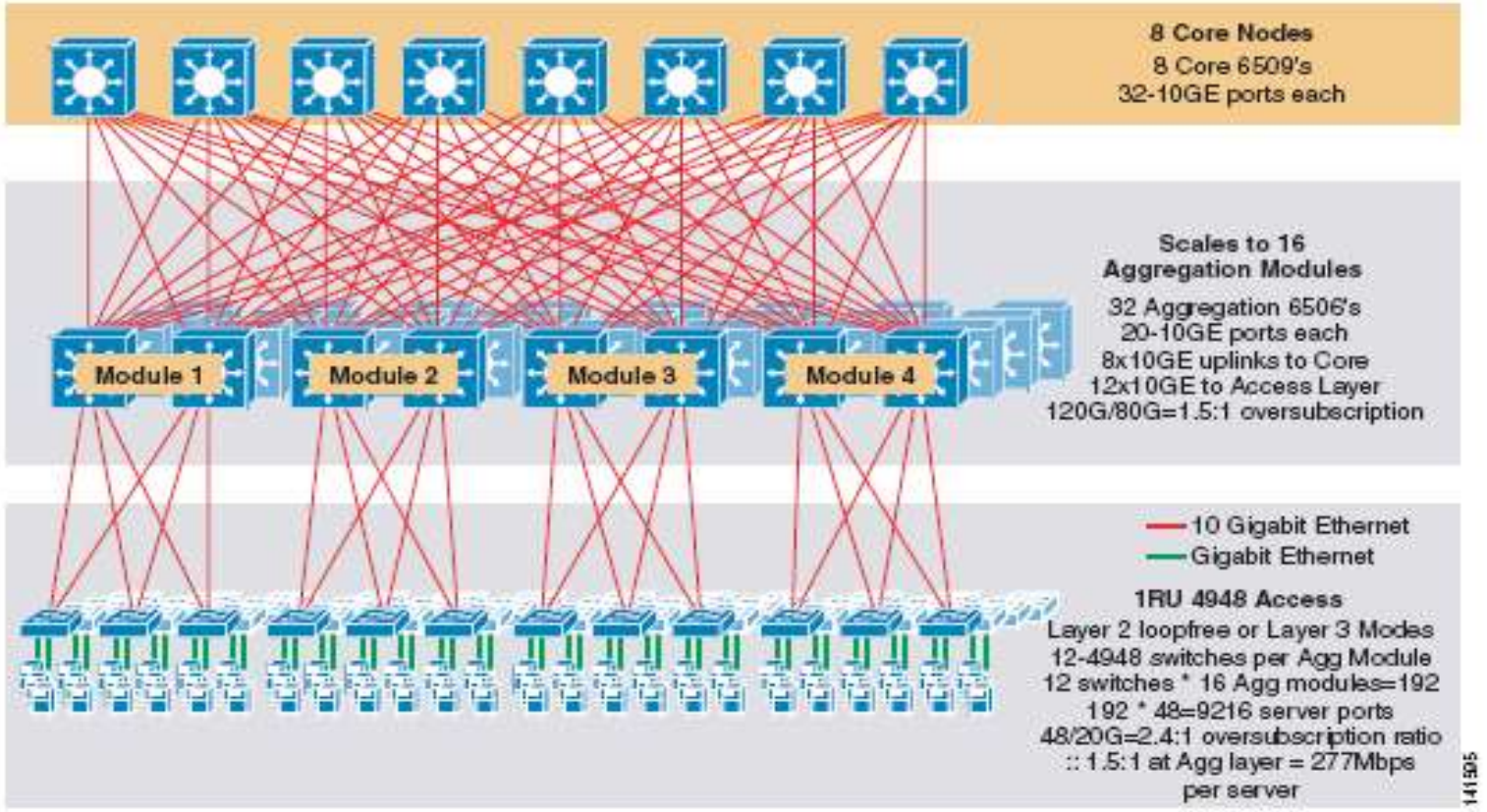
Oversubscr ratio = (R Gb/s)/Thrput (Gb/s) :1

- ## Data Center Access Layer
  - Throughput obtained in the server: 336 servers at 1 Gb/s and 4 uplink ports at 10 Gb/s each. Average Throughput = 0.12 Gb/s (uses 12% of the link capacity) and oversupscription ratio of 8.4:1 (8.4 servers in average are needed to occupy a 1 Gb/s)
    - **Case I:** the server has a performance ("rendimiento") of 10%. Then, the average throughput is of 0.10 Gb/s, since the server is unable to occupy the 0.12 that the switch offers him, we could use the capacity of the uplinks not used by the servers to add more servers. How many ? There are 4*10Gb/s/(M servers at 1 Gb/s)= 0.1 Gb/s → M= 400 servers. So we can add 64 servers to the system.
    - **Case II:** the server has a performance of 20%. The servers try to send 0.2 Gb/s, but that means 0.2 Gb/s * 336 = 67.2 Gb/s > 4* 10 = 40 Gb/s of the uplinks. The switch will cause L2 frame losses. TCP will recover at the client, and will cause that its flow control makes that the servers send at 0.12 Gb/s. Then, now 0.12 Gb/s * 336 = 40 Gb/s that it is the uplink capacity of the switch

- **Data Center Access Layer**
  - If you want better performance:
    - **Improving the uplink between access and aggregation improves the oversubscription ratio**:
      - Instead of 40 Gbps we have 80 Gbps maintaining the 336 servers → we obtain 240 Mbps and a 4.2:1 oversubscription ratio.
    - **Increase the number of access switches maintaining the oversubscription ratio will increase the number of total servers at the cost of increasing the number of access switches,**
    - What limits scalability???
      - **The uplink capacity of the access switch**.

- **Data Center Access Layer**
  - Optimized Oversupscription ratios:
    - Web servers → 15:1
    - Application servers → 6:1
    - Database servers → 4:1
    - HPC → 2.5:1 a 8:1

- **Data Center Access Layer**



- **Example:**
  - Calculate the oversubscription ratio:

Access layer:

oversubscription: 48 GE servers 2x10GE uplink towards aggregation → 2.4:1
Bandwidth per server→ 20Gbps/48Gbps = 416 Mbps

Aggregation layer

oversubscription: 120 GE downlinks to access per 8x10GE uplink a core
→ 1.5:1 → 80/120 = 666 Mbps

Calculate the real bandwidth per server: apply formula a:1 = x:d where a is the throughput (or oversupscription ratio) of the access and d is the throughput (or oversupscription ratio) of the aggregation, and then obtain the x:

ov. ratio:	**2.4:1 = x:1.5**	**→ x=3.6 (3.6:1)**
Thorughput:	**0.416:1 = x:0.666**	**→ x = 0.277 Gbps**