2️⃣

Tema 2 | VLAN

1. Tipos de Switch

- **Unmanaged Switch**
  - Funcionalidad básica.
  - Nada que configurar.
- **Smart Switch**
  - Opciones limitadas de configuración mediante web.
  - Otras funcionalidades, como VLAN.
  - Hardware fijo. No es modular.
- **Managed Switch**
  - Configuración completa mediante web y consola.
  - Hardware modular.

1.1. Parámetros Gestionables

En un *switch*, nosotros como administradores podemos modificar bastantes parámetros.

- **Spanning Tree Protocol**
- **VLANS**
- **Configuración IP**
- **…**

2. VLANS

Todos los dispositivos que estén conectados en la misma LAN pueden comunicarse entre ellos de forma directa.

📌 **¿Qué nos ha motivado a crear VLANS?**

Dominios de Difusión
Necesidad de crear dominios de difusión para un mejor uso el ancho de banda.

Movilidad
Sería útil que el usuario viera la misma LAN independientemente de que se conectase en un punto u otro de la red de nuestra empresa.

Seguridad
Es de interés restringir el acceso a ciertos dispositivos a los usuarios de una red.

Hasta ahora necesitábamos poner *switches* para cada una de las *LANS* que queríamos crear, lo cual es caro y acaba saturando la red. Además no proporciona ventajas como la movilidad.

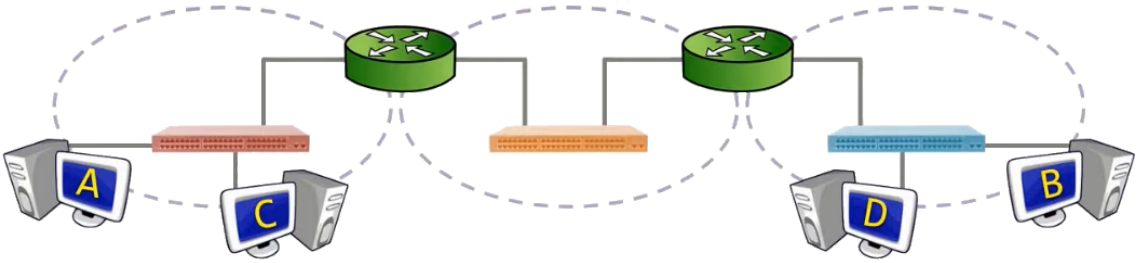
📌 **VLAN**
Es el territorio en el cual se extiende una trama *broadcast*.

Ventajas

- **Gestión de Dispositivos**
  Un dispositivo puede estar en una *VLAN* por su función y no por su localización.
- **Gestión por Software**
  Puedes gestionar los puertos del *switch* directamente desde un panel, sin

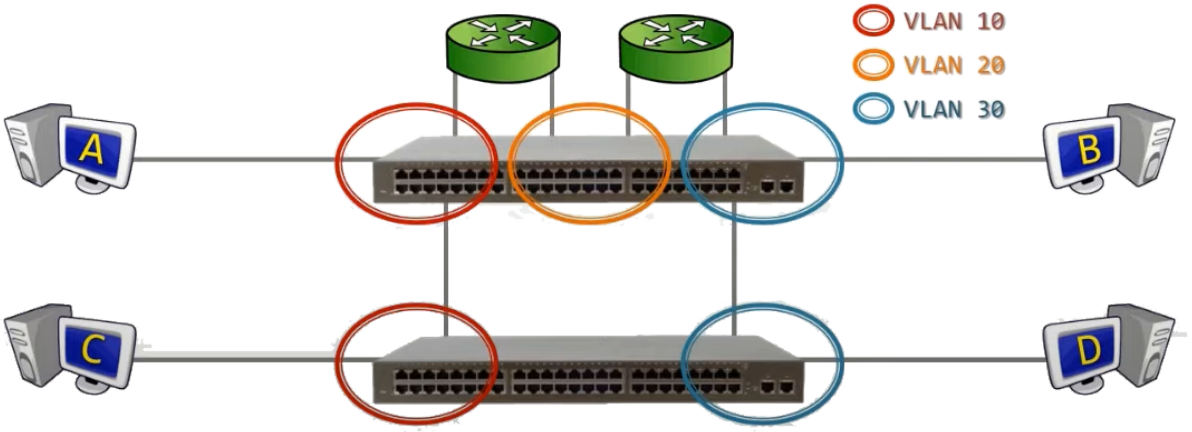
necesidad de acudir físicamente al *switch*.

> ❗ **Una trama no se puede asociar a más de una *VLAN*.**

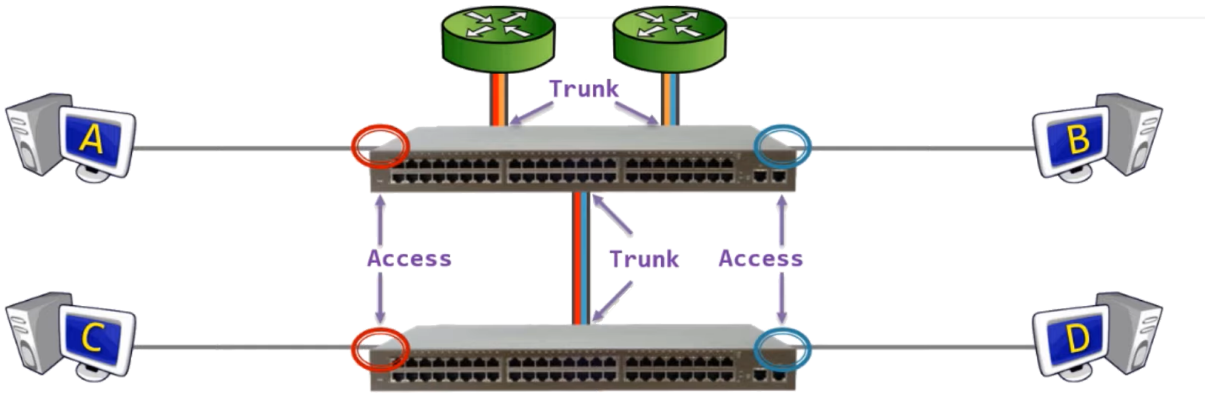


Aquí encontramos `3 redes` totalmente distintas, unidas por `2 switches`.

Esta estructura puede replicarse fácilmente con `1 switch`, o con `2 o más switches` si queremos tener más puertos disponibles.



- Una *VLAN* nos permite partir un *switch* en varios *mini switches*.
- Una *VLAN* nos permite extender esos *mini switches* con otros *switches*.
- Para ahorrar puertos en los *switches*, podemos conectar en un mismo cable, varias *VLAN*.
  Estas zonas compartidas se conocen como *Trunk*.



Al introducir estas zonas compartidas, nos surge un problema. ¿Cómo podemos diferenciar a qué *VLAN* debe ir una trama?

Esto podemos hacerlo poniendo una etiqueta a la trama siempre que vaya a circular por una zona compartida. Vamos a verlo en el apartado siguiente.

2.1. VLAN Tagging

El *tagging* es el proceso dedicado a clasificar una trama en una de nuestras VLAN, para indicar su pertenencia. Esto se consigue añadiendo un identificador en la trama.

¿Cómo saber a qué *VLAN* pertenece una trama?

- **Método Implícito**
  Analizar la trama y aplicar las reglas de pertenencia.
  Consume tiempo y es costoso.
- **Método Explícito**
  Comprobando su etiqueta en la trama.
  Es rápido y barato.

Vamos a intentar realizar el *método implícito* a la entrada de la trama en nuestra red, aprovechando para etiquetar la trama. De esta forma el resto de *switches* podrán comprobar fácilmente a que *VLAN* pertenecen.

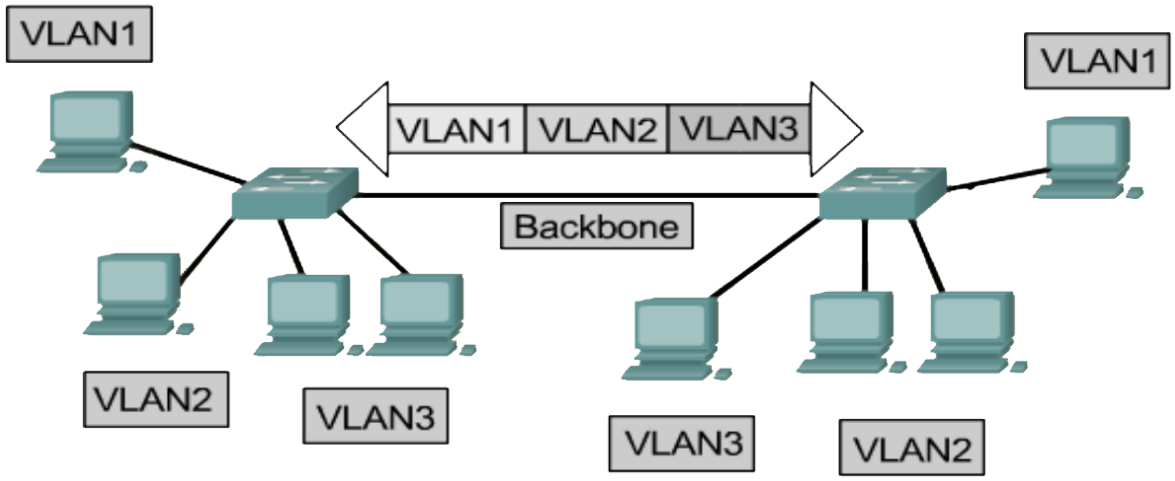
Haremos que los *switches edge* (que tienen menor carga) se dediquen a etiquetar el tráfico, y que los *switches core* tan solo necesiten lee la etiqueta de la trama.

2.2. VLAN Awareness

❗ **Clasificación Switches**
Además de la clasificación *edge* y *core* que vimos en el *Tema 1*, vamos a añadir dos grupos donde clasificar a los *switches*, esta vez por su función.

- **Tag-Aware**
  Un *switch* que es capaz de `poner` / `quitar` / `interpretar` tags de las tramas.
- **VLAN-Aware**
  Un *switch* que no usa etiquetas, pero es capaz de diferenciar las tramas de una *VLAN* y otra.
  `Ejemplo` Nuestro router de casa es capaz de diferenciar las tramas de la red wi-fi principal y de la red de invitados.

Ser **VLAN-Aware** es un nivel inferior a **Tag-Aware**.



💡 **Backbone**
La red principal de alta velocidad que une diferentes redes. Suele utilizarse para unir edificios o plantas en empresas. Se identifica porque sus *switches* dan a otros *switches*.

> 💡 **Trunk**
> Una red generalmente con menor velocidad que conecta con usuarios o dispositivos.
> Se identifica porque sus *switches* dan mayoritariamente con dispositivos.

El tráfico de todas las *VLAN* atraviesa generalmente tanto el *backbone* como algún *trunk*.

Es necesario etiquetar las tramas antes de mandarlas por el *trunk*.

2.3. VLAN Reglas de Asociación

Encontramos diferentes opciones a la hora de crear una *VLAN*, según como queramos hacerlo.
Cada una tiene sus ventajas e inconvenientes.

- **Port Based** `Nivel 1`
  Generalmente la más utilizada.
  Asocias uno o varios puertos a una *VLAN*.
  Puedes establecer que los puertos `1 - 4` corresponden a la `VLAN 1` y los puertos `4 - 10` a la `VLAN 2`.
  - **Inconvenientes**
    - Es poco seguro, puesto que solo se comprueba el puerto.
    - No acepta movilidad.
- **MAC Address Based** `Nivel 2`
  Vincula una `MAC` a una `VLAN`.
  Esto permite tener la misma `VLAN` independientemente de la localización.
  - **Inconvenientes**
    - El administrador debe configurar ordenador a ordenador.
    - Es algo más seguro, pero sigue siendo bastante vulnerable.
      Puedo adivinar una dirección MAC que me permita conectar con el recurso que me interesa, y ponerla en mi tarjeta de red. Debo asegurarme de que el dispositivo que tenía esa MAC está desconectado.
- **IP Subnet Based** `Nivel 3`
  Definimos las *VLAN* por rangos de `IP`.

Un servidor *DHCP* puede tener asociadas las `MAC` con la correspondiente `IP` que debe concederle.

    - **Inconvenientes**
        - Sigue requiriendo trabajo manual por parte del administrador.
        - No es demasiado seguro.
          Necesito que se me asigne una `IP` que me interese para conectarme con el dispositivo. Puede hacerse directamente o usando una `MAC` registrada en el servidor `DHCP` en el rango de `IP` que me interesa.

- **Protocol Based** `Nivel 2`
  Se crea de forma automática usando la etiqueta `TYPE` de las tramas.
- **Application Based** `Nivel 7`
  Asigna una misma *VLAN* a quienes utilicen la misma aplicación.
  El *switch* debe conocer que `PORT` se usa, y tener una lista de los puertos asignados a cada aplicación. En algunos casos tendremos que leer parte de los datos para vera que aplicación corresponde.
    - **Inconvenientes**
        - Supone una carga de trabajo muy alta para el *switch*.

Además, podemos tener criterios que mezclen varios aspectos, incluido el día o la hora en la que estamos.

El campo `Nivel` indica a que nivel de red puede necesitar acceder el *switch* para realizar el filtrado. Cuando menor sea el nivel, menor carga tendrá.

2.4. Distribución de Tramas

Este es el protocolo que van a seguir los *switches* dentro de nuestra red cuando se enfrenten a un paquete y tengan que reenviarlo a la *VLAN*.

`Unicast Conocido`

SI `destino` está en nuestra `VLAN`

> Lo mandamos al dispositivo de nuestra `VLAN`.

SI NO

> Descartamos el paquete.

`Unicast Desconocido` / `Multicast`

> Lo mandamos por todos nuestros puertos.

2.5. El Problema de las Tags

Introducir las `tags` en las tramas *Ethernet* supone aumentar en $4\ Bytes$ el *MTU.*

$$[64..1518] \rightarrow [64..1522]$$

3. DHCP Relaying

3.1. Switches de Nivel 3

Este tipo de *switches* son capaces de leer el nivel 3 de datos y encaminar tramas según su IP. Esto es parecido a lo que hacen los *routers*.

Con los switches de nivel 3, es posible crear diferentes VLAN en una red y enrutar el tráfico entre ellas, lo que permite una mayor flexibilidad y escalabilidad en la configuración de la red.

📌 **Switch Nivel 2**
Enruta tramas mirando la *MAC destino* de las mismas.

📌 **Switch Nivel 3**
Enruta tramas mirando la *MAC destino* o la *IP destino* de las mismas.

Los *switches nivel 3* son peores que los *routers*, pero también son más baratos. Los *routers* de nuestras casas pueden considerarse un *switch nivel 3* con *wifi*. Normalmente tienen varios puertos *Ethernet*.

3.2. DHCP Relaying

Normalmente necesitamos un *servidor DHCP* en cada una de nuestras *LANs*. Lo que vamos a intentar es hacer que podamos tener un único *servidor DHCP* compartido entre todas nuestras *VLANs*. Esto nos va a permitir ahorrar dinero, y centralizar las peticiones *DHCP*.

Para que eso ocurra, vamos a tener que **reencaminar** las *tramas DHCP* a nuestro *servidor DHCP* en nuestros *routers*.

Normalmente es tan sencillo como especificar una línea de configuración en nuestros routers.
`IP Helper Address: 10.100.30.2`

📌 Con esta opción, nuestro router transformará las peticiones *DHCP* sin `IP` y `broadcast`, y la va a retransmitir al servidor *DHCP* transformándola en una trama `IP`.

El *router* pone como *dirección origen* la dirección base de la *VLAN*. Esto es así para que el *servidor DHCP* sepa de que *VLAN* viene la petición y le asigne al dispositivo una `IP` de esa *VLAN*.

4. VLANs para Operadores y Centros de Datos

Imaginemos que somos una empresa grande, con dos sedes principales:

- **Albacete**
- **Valencia**

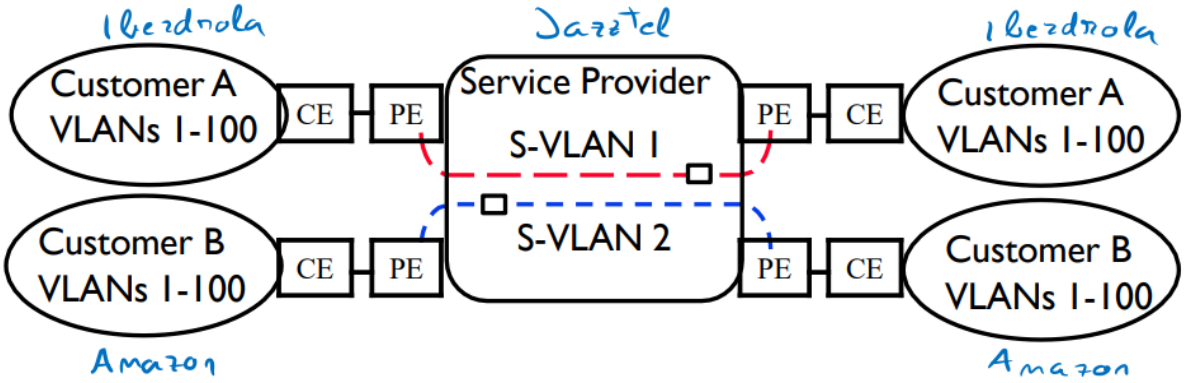
Nosotros queremos hacer que los dispositivos de Albacete estén disponibles dentro de una *VLAN* desde Valencia. Si no queremos pasar por *routers*, debemos mantener los enlaces a nivel 2 `MAC`.

Nos podemos encontrar con varios problemas:

- **Tormentas de Broadcast**
  Las tramas `broadcast` y desconocidas pueden crear demasiado tráfico.
- **Bucles**
  Aparecen fácilmente en una gran red.
- **Problemas de STP**
  - Árboles granes.
  - El nodo raíz puede convertirse en un cuello de botella y en un único punto de fallo.
  - Múltiples rutas permanecen si usar.
- **Tromboning**
  Los servidores y *switches* duales generan un tráfico cruzado excesivo.
- **Seguridad**
  Los datos en la *LAN Extension* deben estar cifrados.

A la hora de realizar esta conexión, podemos optar por dos opciones:

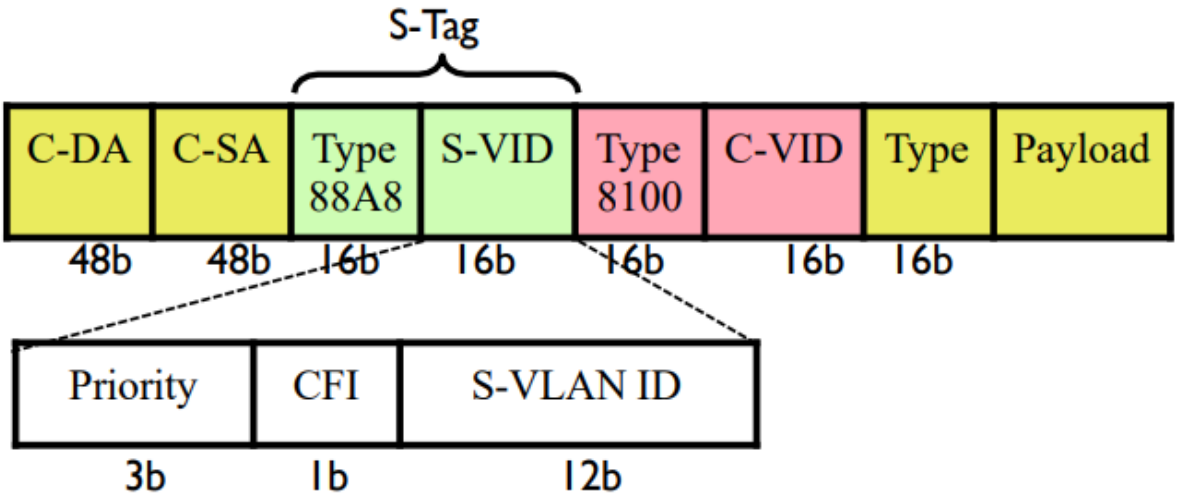
4.1. Provider Bridges



Vamos a suponer que dos grandes empresas (*Iberdrola* y *Amazon*) van a unir a través de *Jazztel* sus centros de datos en dos localizaciones. Vamos a encontrarnos con un problema:

❓ **¿Qué puedo hacer para no mezclar el tráfico de ambas empresas?**

Jazztel se va a encargar de encapsular todas las peticiones de una empresa en una *VLAN*, y las de la otra empresa, en otra *VLAN*. Para esto se encapsula la petición con otra `TAG VLAN`.



En este caso imaginemos que *Amazon* quiere mandar una trama a su *VLAN 63.* Cuando pase por *Jazztel*, va a volver a encapsular la trama en la *VLAN 10*. *Jazztel* sabe que todo lo de la *VLAN 10* es de *Amazon.* Cuando va a volver a pasar a *Amazon*, se le quita la `TAG VLAN` de *Jazztel*, y se redirigirá a donde le tocaba desde el principio.

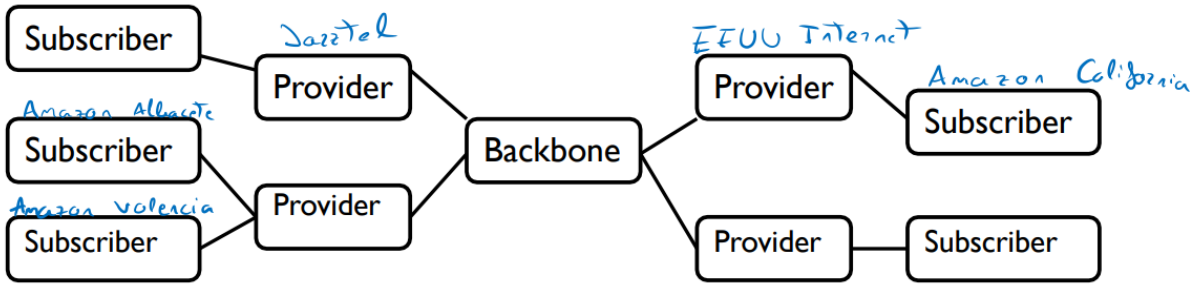
De esta forma *Jazztel* no mezcla las *VLAN* de cada empresa, puesto que ellos encapsulan todo lo de *Amazon* en la `VLAN 10` y lo de *Iberdrola* en `VLAN 15`.

Por tanto, vamos a diferenciar la `S-TAG` y la `C-TAG` en la trama *Ethernet*.

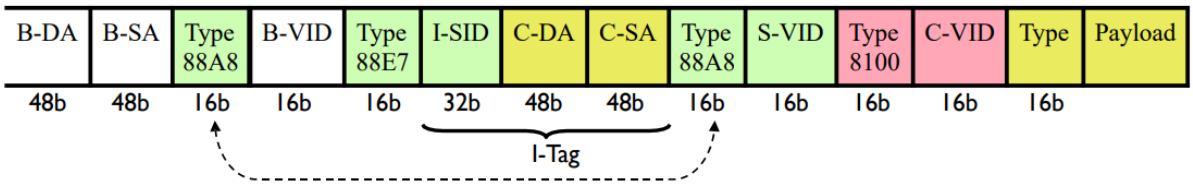
Esto es conocido como $Q - in - Q$.

4.2. Provider Backbone Bridges

Ahora queremos conectar *Amazon Valencia* y *Amazon Albacete* con *Amazon California*. Vamos a tener que pasar por un *Backbone* intercontinental.



Nuestra trama necesita crecer ahora para poder tener `B-TAG`, `S-TAG`, y `C-TAG`. La *Backbone Tag*, *Service Provider Tag*, *Client Tag*.



4.3. Resumen

- Original Ethernet
- IEEE 802.1Q VLAN
- IEEE 802.1ad PB
- IEEE 802.1ah PBB