REDES LAN

Tecnología Ethernet

Es un **estándar de redes de área local.**

Es una red **Peer to Peer** donde el control esta totalmente **descentralizado**. **No se necesita interactuar con un dispositivo central**, cada nodo corre la lógica del protocolo.

El principio de **funcionamiento** original se conoce como **CSMA/CD** (Carrier-Sense Multiple Acess / Colision Detection) o **Acceso múltiple por detección de portadora con detección de colisiones.**

Originalmente esta tecnología es **half-duplex** (ambos hablan pero solo uno a la vez).

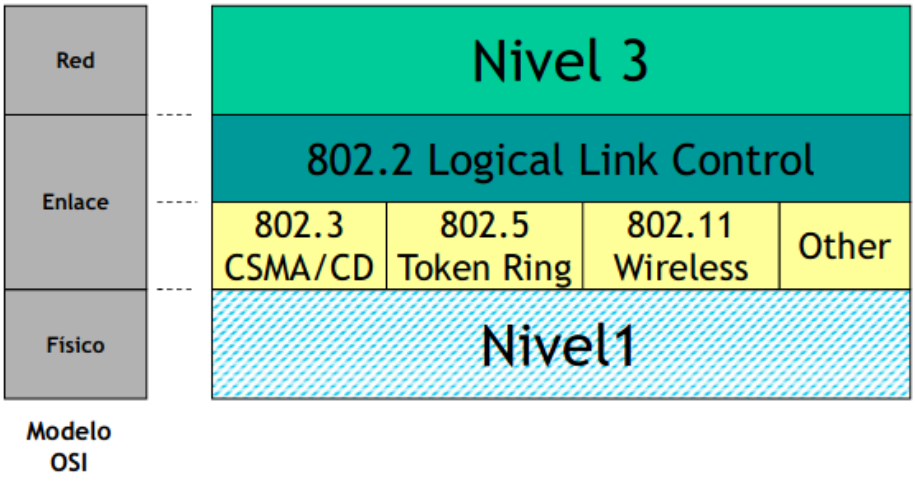
La IEEE 802.3 estandarizo el CSMA/CD que es lo que creo Ethernet (luego incorporo a Ethernet). Por lo que es lo mismo decir **protocolo 802.3 y Ethernet.**

Evolución

- 1970 → Aloha Radio Network (Hawaii).
- 1979 → DIX Ethernet II (Digital, Intel y Xerox).
- 1985 → IEEE 802.3 Standard (10Mbps).
- 1995 → Fast Ethernet (100Mbps).
- 1998 → Gigabit Ethernet.
- 2002 → 10Gb Ethernet.

Durante 40 años el formato de la trama no se ha alterado, por lo que un dispositivo del 1970 se podría conectar a uno de hoy en día.

Donde se situá en el modelo OSI



Encuadramos al protocolo **802.3 (CSMA/CD)** (o Ethernet) como un **protocolo que opera en la capa de enlace**.

Estos protocolos operan en la capa de enlace pero **no cumplen con todas las especificaciones** de la misma según el modelo OSI, por lo que la IEEE estandarizo el

protocolo **802.2 que viene a cumplir lo que los otros protocolos (802.3, 802.5, etc) no hacen.**

A estas dos se la llaman subcapas donde la primera (inferior) es **subcapa MAC** (media access control) y la segunda (superior) **subcapa LLC.**

Subcapa LLC

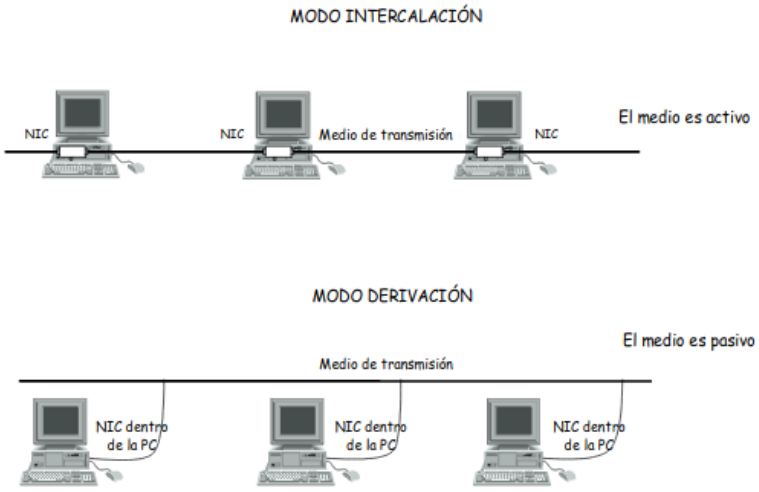
La subcapa LLC de Ethernet se ocupa de la **comunicación entre las capas superiores y las capas inferiores.** Se implementa mediante software (no depende del hardware).

Subcapa MAC

La MAC se implementa mediante hardware. Esta subcapa tiene dos responsabilidades:

- **Encapsulación de datos.**
- **Control de acceso al medio.**

Topología original red LAN Ethernet

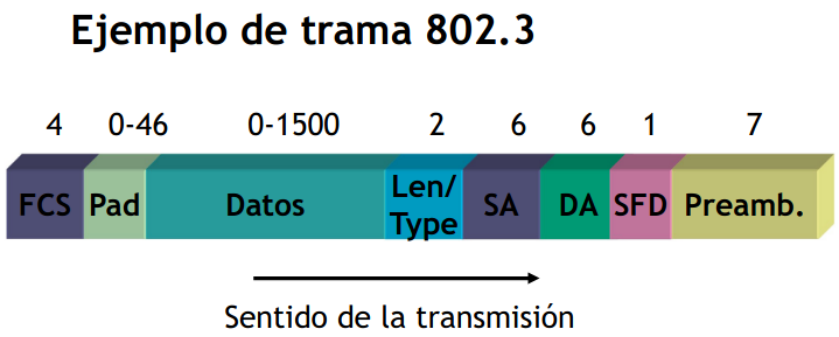


Se dice topología original porque el BUS hecho con cable coaxil que era inflexible, se suplanto por cable UPC. Se usa esta topología para explicar porque explica bien el funcionamiento y es fácil de interpretar.

Modo intercalación: es también llamado **coaxil fino**, iba interconectando por las diferentes unidades con un conector T que permitía que el cable siga a la siguiente estación.

Modo derivación: es también llamado **coaxil grueso**, las unidades se iban “colgando” del coaxil.

Formato de la trama Ethernet / 802.3



La trama posee:

- **Longitud mínima** de 64 bytes.
- **Longitud máxima** de 1518 (sin incluir preámbulo ni SFD).

Preámbulo: Son los siguientes **7 bytes** 10101010. Cumple la función de **establecer el sincronismo**. Una estación la transmite, todas reciben la señal, empiezan a decodificar el manchester y encuentran el preámbulo hasta que encuentran la SFD.

SFD (Start of frame delimiter): Es un unicamente el bit 1. **Indica que finalizo el preámbulo y el periodo de sincronización**, y lo que viene a continuación es el primer campo significativo.

DA: 6 bytes. Es la **dirección destino** (a quién va dirigido el mensaje).

- Es lo que se conoce como **MAC address**. Cada estación tiene una dirección MAC única (**no hay posibilidad de ambigüedad o duplicidad**). Estos **6 bytes** están **compuestos por dos campos** de 3 bytes cada uno:
  - OUI (Organizationally Unique Identifier) o **identificar único del fabricante** (primeros 3).
  - DUI (Device Unique Identifier) o **identificador único del dispositivo** (de red).
- **Tipos** de direcciones MAC:
  - **Dirección de Broadcast** (48 unos). Unicamente puede ser destino e indica que el mensaje esta destinado para todos los nodos de la red.
  - **Dirección multicast:** El mensaje va dirigido a un grupo de estaciones.
  - **Dirección unicast:** Destinado a una única estación.

SA: 6 bytes. Es la **dirección origen** (quién transmite el mensaje). MAC address.

Len / Type: 2 bytes. Es una de las diferencias entre Ethernet y 802.3.

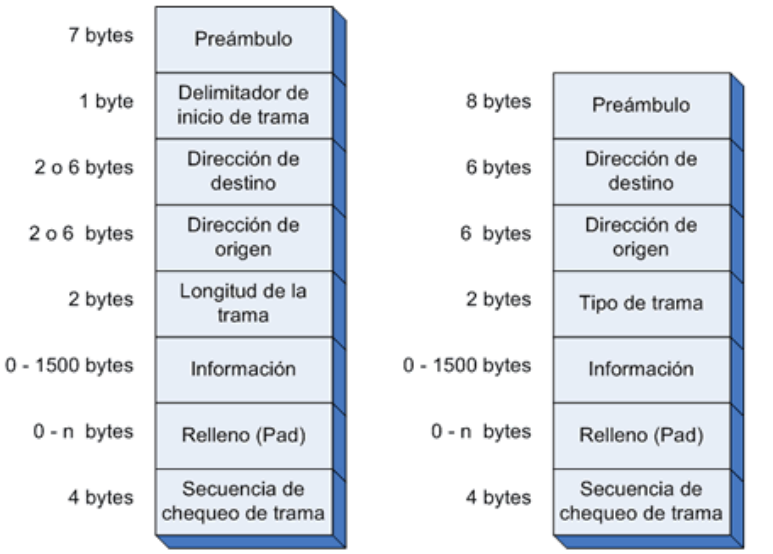
- Para **802.3** (Frame length) es el campo longitud que indica cual es la **longitud del campo de datos** de longitud variable que sigue a continuación.
- Para **Ethernet** (Ethertype) es el campo que indica cual fue la **capa usuaria (la capa superior) del modelo OSI que origino el mensaje**, así sabe a que capa entregárselo.

Datos: Campo de longitud variable entre 0 y 1500 bytes.

Pad: 0-46 bytes. Campo de relleno. **Sirve para asegurarse que la trama tenga al menos 64bytes.**

FCS (Frame Check Sequence) o CRC (Verificación de redundancia cíclica) : 4 bytes. **Sirve para detectar si hay un error en la transmisión** de la trama. Se aplica sobre dirección destino, dirección origen y data.

Diferencia entre trama IEEE 802.3 y Ethernet



Formato de la trama IEEE 802.3 Formato de la trama Ethernet

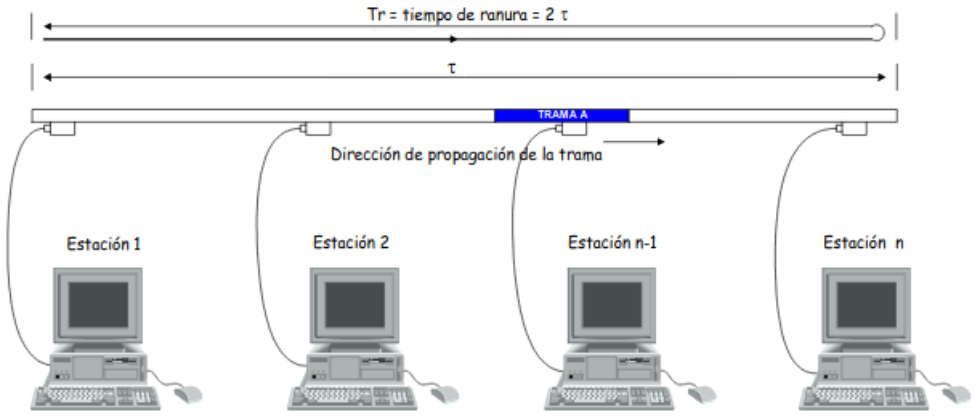
Una de las diferencias entre el formato de las dos tramas está en el **preámbulo**. El propósito del preámbulo es anunciar la trama y permitir a todos los receptores en la red sincronizarse a si mismos a la trama entrante. **El preámbulo en Ethernet tiene una longitud de 8 bytes pero en IEEE 802.3 la longitud del mismo es de 7 bytes**, en este último el **octavo byte se convierte en el comienzo del delimitador de la trama.**

La segunda diferencia entre el formato de las tramas es en el **campo tipo de trama que se encuentra en la trama Ethernet**. Un campo tipo es usado para especificar el protocolo que es transportado en la trama. Esto posibilita que muchos protocolos puedan ser transportados en la trama. El campo tipo fue **reemplazado en el estándar IEEE 802.3 por un campo longitud de trama**, el cual es utilizado para indicar el numero de bytes que se encuentran en el campo da datos.

La tercera diferencia entre los formatos de ambas tramas se encuentra en los **campos de dirección**, tanto de destino como de origen. Mientras que el formato de **IEEE 802.3 permite el uso tanto de direcciones de 2 como de 6 bytes, el estándar Ethernet permite solo direcciones de 6 Bytes.**

La cuarta diferencia es que en la **capa de enlace la 802.3 tiene 2 subcapas LLC y MAC, la Ethernet no tiene esas dos capas.**

Transmisión



Mecanismo CSMA/CD

Escucha el medio y si esta libre (no hay nadie transmitiendo), **empieza a trasmitir** el preámbulo y el resto de la trama (sino espera que este libre). **La trama se propaga por el medio y le llega a todas las terminales**, ahí finaliza la transmisión (**en Ethernet no existe la confirmación**, se asume exitosa la transmisión si no hay colisión). Se da cuenta que no hay nadie transmitiendo porque no hay señal (manchester).

Si **dos o mas estaciones comienzan a transmitir al mismo tiempo** (ambas encontraron el medio libre y al mismo tiempo comenzaron), las señales se interfieren mutuamente, a esto se lo llama **colisión**. El resultado de una colisión es que **ambas estaciones van a tener que transmitir en otro momento**. Las estaciones detectan las colisiones porque **al mismo tiempo que transmite, escuchan lo que hay en el medio**, si lo que escuchan en el medio no es lo que están transmitiendo, detectan la colisión.

La **capa MAC es la que escucha al medio.**

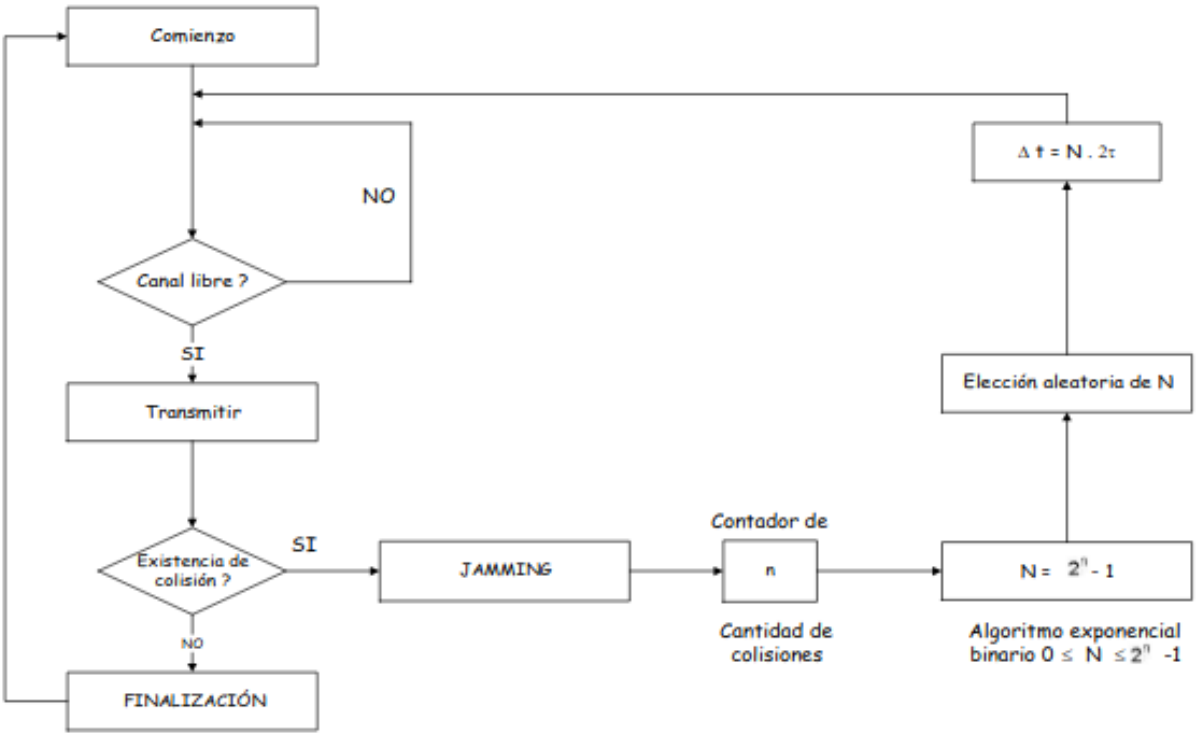
Al **detectar una colisión**, la estación pone en funcionamiento el algoritmo de **backoff exponencial**, que lo hace es ordenar las transmisiones en el tiempo.

La **probabilidad de colisión aumenta a medida que el medio es mas largo**. Por ello se estableció la limitación de la **longitud máxima de la red** (cable coaxil) de 500m, pero con la utilización de repetidores (máximo 4) se aumento a 5x500m = 2500m.

Se establece la transmisión de una **trama mínima** de 64bytes **debido a que con el tiempo que se tarda en transmitirlo, se recorren los 2500m por lo que hasta la ultima estación escucharía la trama y detectaría una colisión en caso de también estar transmitiendo.** A esto se lo llama **ventana de colisión**. Si fuera por debajo de 64bytes nunca se enteraría que su transmisión colisiono con otra.

Slot: Ranura de espera (**tiempo que se tarda en enviar 64bytes**). A una velocidad de 10Mbps 512 bits/10Mbps = 51,2 useg.

Algoritmo exponencial binario (backoff exponencial)



Backoff exponencial es un algoritmo que **se utiliza para espaciar retransmisiones** repetidas del mismo bloque de datos de manera multiplicativa, a menudo como parte de la **evitación de congestión de red.**

Jamming es un código que no da CRC correcto, utilizado para sostener la transmisión una vez que se detecto la colisión para que las demás estaciones también la detecten.

Luego **agrega una unidad al contador de colisiones (n**, que es una variable global), **el algoritmo exponencial alimenta a una variable N** (cantidad de veces del tiempo de ranura que debe esperar antes de volver a intentarlo) que puede valer entre 0 a 2^{n-1}, **elige aleatoriamente un valor de N** y **espera un valor N de tiempo para volver a intentar a transmitir.**

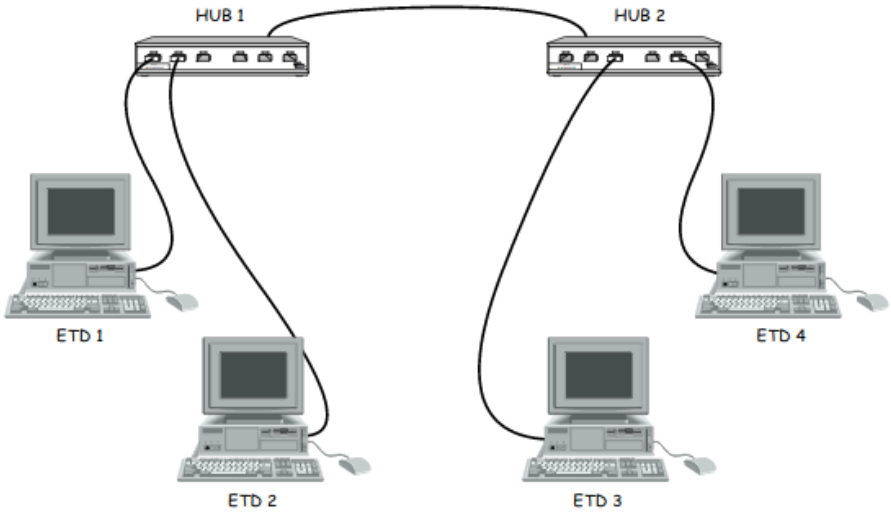
Luego de 10 colisiones consecutivas, se setea el máximo de ranuras a 1023 para bajar la probabilidad de colisión.

Luego de 16 colisiones consecutivas, la subcapa MAC aborta la misión.

Cuando logra transmitir, se resetea el contador de intentos.

N = (1/ probabilidad de colision) -1.

Dominio de colisión



Dominio de colisión simple

HUB: Dispositivo de capa 1, solo entiende de señales eléctricas. **Combina las señales eléctricas de todos los puertos como si fuera un bus.** Cuando una estación 1 transmite, todas las demás escucha. Es half-duplex. Dispositivo de red que trabaja en la capa física del modelo OSI, su funcionamiento es similar a un cable por lo que **solo cuenta con un dominio de colisión.**

Switch: Dispositivo de red usados para **dividir los segmentos de colisión**, cada puerto es un segmento diferente.

Dominio de colisión: segmento de una red **donde es posible que las tramas puedan colisionar con otras.**

Las estaciones conectadas al HUB pueden colisionar entre si, es decir, compiten por la capacidad del HUB. Si el HUB es de 10mb, hay un bus de 10mbps que va a ser utilizado de a uno a la vez, entre todos los que estén conectados. Si conecto dos HUB, estoy extendiendo el BUS y haciendo que todos formen un único dominio de colisión.

A las estaciones se las llama dominio de Broadcast, debido a que cuando enviá un mensaje, todas las demás estaciones lo van a recibir.

Dominios de Broadcast: Segmento de la red que involucra a todos los dispositivos que recibirán frames de Broadcast provenientes de cualquier dispositivo del conjunto. Los **routers son usados para dividir estos dominios**, cada uno de sus puertos pertenece a un dominio de broadcast.

Cuantas mas estaciones estén conectadas al BUS, mayor es la contención y menor el ancho de banda.

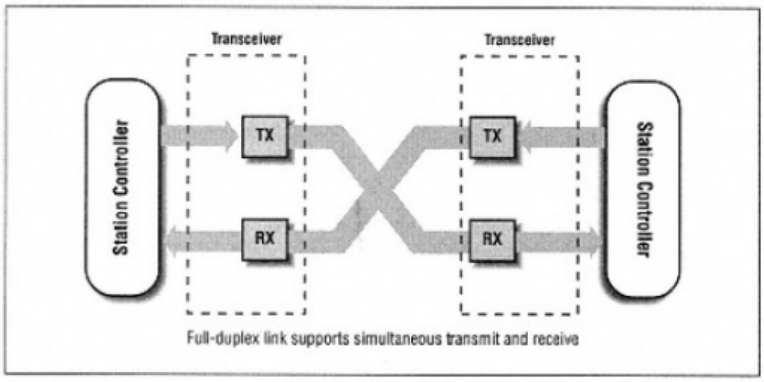
Designación de tecnologías (configuraciones físicas)

- **10Base5:** 10Mbps, transmisión en banda base, 500m (longitud máxima del segmento, coaxil grueso RG-218).
- **10Base2:** 10Mbps, transmisión en banda base, 185m (longitud máxima del segmento, coaxil fino RG-58).

- **10BaseT:** 10Mbps, transmisión en banda base, 100m (longitud máxima del segmento, cable UTP(Unshielded Twisted Pair)).
- **100BaseT:** 100Mbps, transmisión en banda base, 100m (longitud máxima del segmento, cable UTP(Unshielded Twisted Pair)).

Interfaz full-duplex

Para obtener una comunicación full-duplex dos estaciones deben estar conectadas punto a punto con un vinculo full duplex.



En el entorno full-duplex, el canal de transmisión de una estación esta vinculado al de la recepción de la otra, por lo que puede transmitir cuando quiera, es decir, **se elimina el CSMA/CD** (no hay contención, no se conecta a un BUS, sino a otra estación). Puedo transmitir y recibir al mismo tiempo.

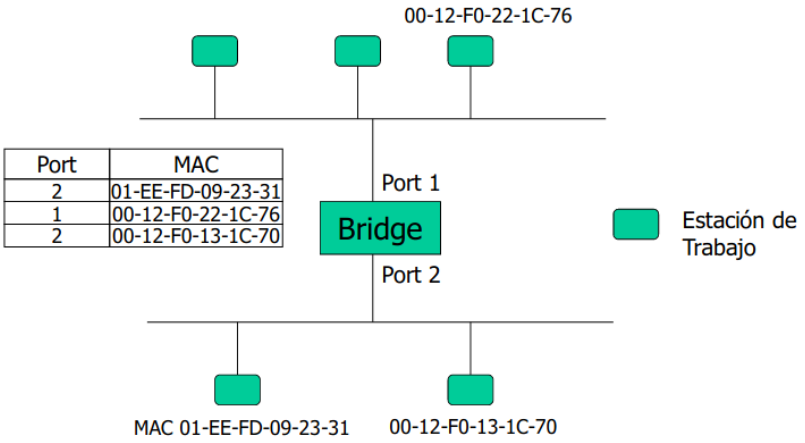
Como la **velocidad de transmisión** no se divide entre las estaciones del BUS, es **mucho mayor que en CSMA/CD** (ademas que las colisiones y la resolución de las mismas baja mucho el rendimiento).

Bridging

Sirve para **mejorar el rendimiento.**

Conecta dos o mas BUS (**redes LAN**). Cada BUS tiene conectadas estaciones que compiten por la capacidad (velocidad de transmisión) de ese BUS (el Bridge cuenta como una estación mas). Por lo que se obtienen **dos dominios de colisión y uno de Broadcasts.**

Como se ve en la siguiente imagen, si conecta un HUB, a un mismo puerto adjunta varias MACs.



Transparent Bridge

- Operan en **capa 2** y **utilizan las direcciones MAC para encaminar las tramas.**
- **Aprenden automáticamente la ubicación de los hosts**. El bridge va "aprendiendo" quien esta de cada lado de sus interfaces (es decir, que estaciones hay en cada BUS), hasta que aprende de que lado esta cada estación, va pasando las tramas de un lado al otro.
- **Las tramas soportan dos procesos: filtering y forwarding** (a la trama o se la filtra, o se la pasa, dependiendo si el destino esta en el mismo BUS o en el otro).
- **Transparente significa que ninguna otra estación conoce de su existencia**. Las estaciones se comunican como si estuvieran conectadas a la misma red y no existiera el Bridge.

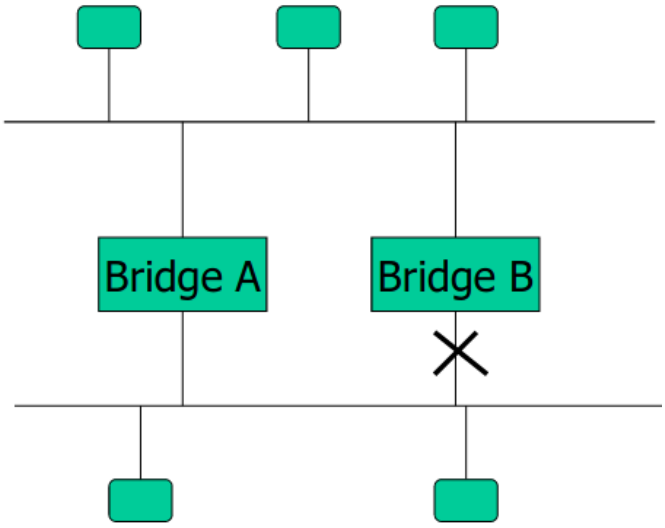
Translating Bridge

- **Realiza además conversión de protocolo y velocidad.**

Bridging Loop y STP

Debido a que un Bridge se conecta a electricidad, puedo querer tener dos Bridge en caso de que uno de los dos deje de funcionar. Esto puede producir lo llamado **Bridging Loop** que sucede cuando se **enviá una trama de Broadcast, los dos Bridges lo toman y lo empiezan a enviar sin parar** (porque también le llega ese mensaje del otro Bridge y lo vuelve a enviar).

Los briging loops **se producen por el desconocimiento de la existencia de otros bridges en la red**.



Por ello se existe un protocolo que se llama **Spanning Tree** que **impide que se generen bucles** por **enlaces redundantes.** Genera **BPDU** (Bridging PDU) que **hace que los Bridges se conozcan entre si** (los bridge envían estos BPDU, si no los enviá significa que se murió).

- **Descubre loops y desactiva vínculos redundantes** (rompe bucles).
- **En caso que un link se desconecte**, se dispara nuevamente el STA, para **activar el link desconectado por el STP.**
- Todos los bridges (o switches) en una red participan del proceso de elección del **root** (entre todos indican cual es la raíz del árbol).
- Se envían **BPDU cada 2 segundos.**
- Todo switch tiene un **Bridge ID** (8bytes) compuesto de:
  - Bridge **priority** (se puede cambiar a mano también).
  - **MAC address.**
- La prioridad menor (menos bridge ID, si tiene el mismo, desempata por MAC address mas baja) se designa ROOT.
- Cuando cambia el estado de un port, se envían notificaciones de cambio de topología (TCN) y comienza nuevamente el **calculo del árbol.**

Establecer switch raíz: Todos los switches reciben las BPDU durante el proceso de determinacion de Switch raiz, y determinan que el switch cuyo valor de BID raiz es el mas bajo sera el switch RAIZ.

El **puerto** de un **Switch cuando se conecta un dispositivo**, para no generar un bucle, pasa por los siguientes estados:

- Inicialmente se encuentra en **Blocking** ya que desconoce si el dispositivo conectado enviá BPDU o no.

- Luego a **Listening** escuchando el trafico de la red, para saber si lo que acabaron de conectar es un Switch o no.
- Luego pasa a **Learning** para aprender MAC addresses. SI un puerto no pasa a Forwarding (bloqueado por spanning tree), igualmente recibe BPDU.
- Luego pasa a **Forwarding** que seria activado (**no pasa a este estado si se formaría un bucle**). Todos los **puertos del root están en Forwarding.**

Switch

Es un dispositivo de **capa 2.**

Conecto cada estación a un puerto del Switch. Las conexiones son **full-duplex** (**divide los dominios de colisión**).

Cuando se enviá un unicast las otras estaciones nunca se enteran de la trama debido a que el Switch solo la envía al destinatario.

Los Switch tienen una **tabla** (**CAMP**) que **marcan que MAC address se conecta a cada uno de sus puertos.** Esta tabla tiene un tamaño máximo (cantidad de MAC address que puede recordar, idem el Bridge). Si se supera el limite, agenda la nueva dirección en la ultima posición (es decir, olvida la entrada mas vieja para insertar la nueva).

Modos de operación

Ingresa una trama a un Switch por un puerto, el Switch **analiza la trama**, **lee la dirección MAC destino y lo conmuta** (busca en su tabla la dirección MAC y empieza a escribir la trama en el buffer de salida del puerto asociado).

Dependiendo su forma de construcción tienen **diferentes modos de operación**:

- **Cat throw:** al leer los primeros 6 bytes de la dirección MAC destino, el dispositivo comienza el proceso de conmutación. Lo que **puede suceder es que se produzca una colisión** (con solo 6 bytes puede suceder) y se tenga que reenviar la trama.
- **Store and Forward:** espera a que la trama se reciba completamente, calcular el CRC para verificar que se haya recibido correctamente y recién después leer la dirección MAC destino y conmutarla. Espera todos los bytes de la trama.
- **Fragment free:** Intermedio entre los dos anteriores. Esperamos los primeros 64bytes para asegurarse de que no va a colisionar la trama.

Switch non-blocking: Son aquellos que no bloquean la transmisión de paquetes cuando esta transmitiendo otros paquetes. En los switch non-blocking la banda ancha interna puede manejar todas las bandas anchas de los puertos, operando a máxima capacidad.

Loop y STP

Sucede lo mismo que en los Bridges.

Cada Switch se aprende las MAC address de los otros Switch.

VIRTUAL LANs (VLANs)

Una VLAN **divide dominios de broadcast, aísla las redes** y se **requiere un dispositivo de nivel 3 para interconectar las VLANs** (es decir, perdemos visibilidad en nivel 2).

Son **creadas dentro de un mismo switch con facilidad de VLANs** (es decir, las debe soportar).

Permite dividir el switch lógicamente, es decir que el switch genera mas de un dominio de broadcast (**cuando un puerto enviá un broadcast, solo lo va a recibir los puertos pertenecientes a la misma VLAN**). Es como si tuviera dos (o mas) switch separados.

Para **comunicar dos VLAN, necesito utilizar un router** (dispositivo capa 3), es imposible que se comuniquen los puertos de diferentes VLAN sin el.

VLAN por puerto: Se configura a que VLAN corresponde cada puerto del switch.

VLAN por MAC address: Algunos switch soportan VLAN por MAC address. **Cada VLAN registrara MAC addresses** pertenecientes a ella, por lo que **independientemente del puerto** al que conectemos el dispositivo, si tiene la misma MAC address, sera reconocido por la misma VLAN. **Si se conecta un dispositivo con una address no reconocida por ninguna VLAN, sera como que no exista.** El administrador debe registrar cada MAC address.

Protocolo 802.1Q

Problema que soluciona

Si tengo un switch dividido en VLANs completas (no me quedan mas puertos libres), y deseo agregar otro puerto a una VLAN debo conectar uno de los puertos de la VLAN a otro switch. Esto lo debería repetir para cada VLAN. Debido a que esto es problemático (restrinjo el segundo switch a una única VLAN), se aplica el protocolo 802.1Q.

Protocolo 802.1Q

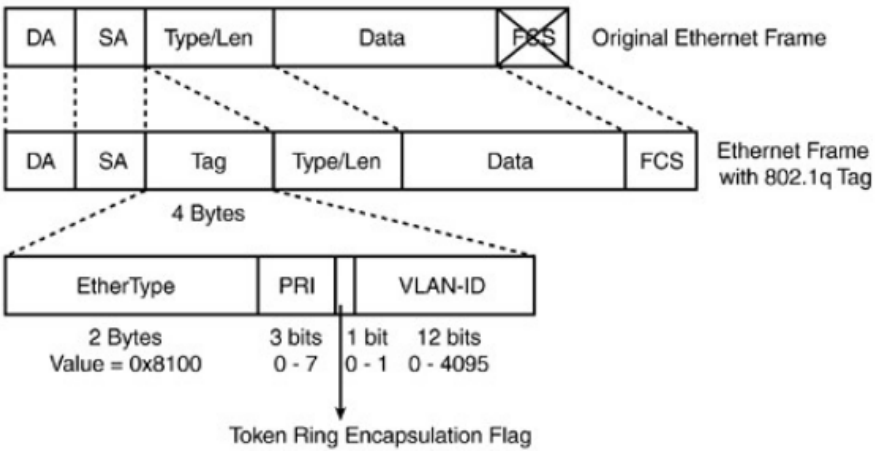
Ambos switch deben soportar el protocolo IEEE 802.1Q (protocolo de capa 2), es decir el VLAN **trunking** o VLAN **tagging**.

Al configurar el 802.1Q donde se da la conexión entre ambos switches deja de pertenecer a la VLAN y pasa a puerto de trunk. **La idea del trunk es hacer una vinculación por el que pase el trafico de todas las VLAN conectados a ese switch**. Es decir, en ese nuevo switch puedo tener puertos pertenecientes a las diferentes VLAN del switch primero.

Lo que logro es que el **dominio de broadcast del primer switch se extienda y tenga miembros en otro switch**. Esto se logra insertando una **etiqueta en las tramas** que circulan por la vinculación (cable que vincula los trunk). Es decir, cuando un dispositivo manda un broadcast, se le agrega una etiqueta que dice “pertenece a la VLAN x(ej 1)” y se lo pasa por el trunk al switch 2 (se puede configurar para que solo pase trafico de las VLAN existentes en el switch 2). El switch 2 al recibir la trama, le quita la etiqueta y se la pasa a los puertos de la VLAN correspondiente.

Ninguna estación conoce el vinculo entre los switches.

Etiqueta en la trama



Ethernet no tiene definida la prioridad de las tramas. Por lo que se aprovecho la creación del tag del protocolo 802.1Q y se implemento el **protocolo 802.1p** con esos tres bits de prioridad (PRI) que me permiten definir 8 niveles de prioridad, por lo que **el switch maneja de manera diferenciada las tramas según su prioridad**. Esto requiere que el switch implemente en cada puerto buffers (colas) diferentes para trafico prioritario y el que no lo es.

Conexión a internet

Cuando quiero proveer de internet a alguna VLAN **debo conectar alguno de sus puertos a un router.** En el caso de que varias VLAN de un switch quieran conectarse a internet, debo utilizar un puerto de cada una para la conexión con el router.

Para evitar esto, **defino un puerto del switch como trunking** (por ende, no pertenece a ninguna VLAN) **y conecto dicho puerto al router.** Para ello el **router debe soportar el protocolo 802.1Q.**

Para ello, **en el router defino interfaces virtuales** (o lógicas) que se comunican con la VLAN correspondiente, así cuando el router responda, lo hace a través de la interfaz virtual correspondiente. Las interfaces físicas en los router son los puertos (conectores físicos), en este caso utilizo un único que lo defino como trunking y lo divido en una interfaz virtual para cada VLAN.

Mientras yo no defina la interfaz virtual para una VLAN x, esa VLAN no tiene internet.

Servicio storage

Es lo mismo que el caso anterior, para evitar el uso de muchos puertos, **defino un puerto trunk en el switch y en el servidor (de almacenamiento)**. El servidor **debe soportar 802.1Q.**

LACP

Si, por ejemplo, conecto a internet todas las VLAN a través de un mismo puerto físico, podría ocurrir un cuello de botella sobre la velocidad. Por lo que conecto a un mismo router, dos trunks desde un mismo switch. Esto generaría un bucle y el spanning tree

bloquearía una de las dos conexiones. Para evitar el bucle y el bloqueo, utilizo el protocolo LACP (tanto el switch como el router deben soportarlo).

El **protocolo LACP** me **permite definir dos o mas puertos físicos como un mismo puerto lógico**, por lo que **no se genera el bucle, ni es bloqueado por el spanning tree** pero si se **aumenta la capacidad de intercambio de tramas.**