

Normas:

- ISO
- ITU
- ISOC / IETF
- ATM Forum
- IEEE

Redes:

Clasificación:

Según su extensión: Pero hoy día no es tan sencillo

1. LAN: Local Area Network
2. WAN: Wide Area Network
3. GAN: Global Area Network

Topología: Forma física

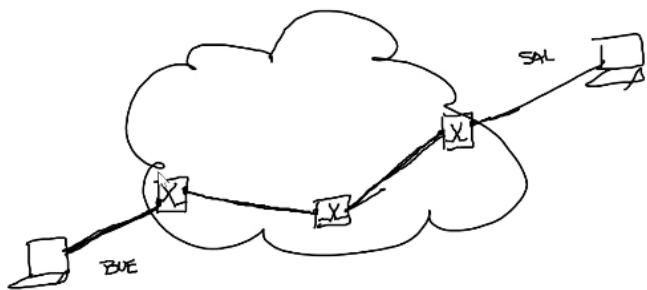
1. Bus o barra
2. Anillo
3. Estrella
4. Híbrida

Cableado estructurado: Especificación de cómo hacer un tendido de red en una planta - edificación.

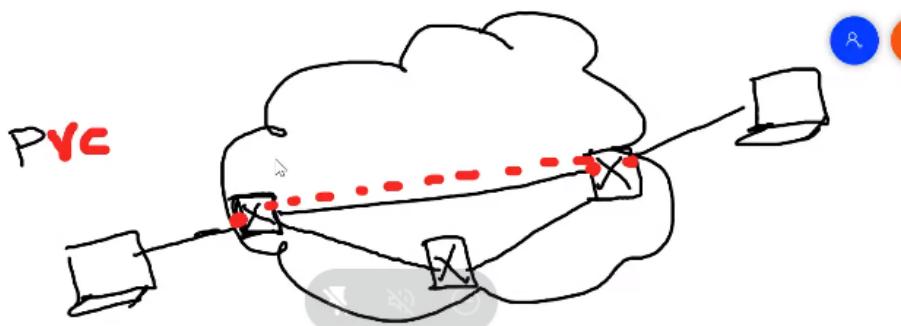
Red LAN vs Red WAN:

Diferencias en:

- Públicas / privadas: En LAN es completamente privada, en WAN es pública porque comparto infraestructura con otro, es la de las empresas proveedoras de telecomunicaciones.
- Diferentes anchos de banda. velocidades de transferencia y tasa de error: En ancho pensamos en Mbits(WAN) o Gbits(LAN) depende la red. En LAN nos aseguramos la tasa de error que podemos tener (10^{-9}), pero en WAN son mucho mayores, por la distancia y otros factores entonces pensamos en anchos de banda menores que en LAN, en WAN debemos contemplar que tan ocurrentes son los errores e implementar métodos de recuperación (checksum, etc).
- Gestión del enlace: Si se rompe algo o cae en LAN lo tengo que comprar/arreglar, en WAN sería delegar eso a quien lo gestiona (proveedor de servicios).
- Redes de commutación de circuitos / paquetes: Conmuta circuitos cuando reserva recursos y puertos entre un abonado y el otro. Se cobra por el tiempo que esos recursos se encuentran bloqueados. Ej: Teléfono.



Comunicación de paquetes: Configuración en los nodos para que los paquetes lleguen de un abonado al otro, los distintos caminos no están reservados completamente para mi paquete sino que para varios, la red se puede sobreender (vender a varios clientes) esperando que no todos la usen al mismo tiempo, sino genera congestión. PVC: Red virtual. circuito virtual permanente



En red LAN es conmutación de paquetes (todas estas redes operan así). En WAN podés usar conmutación de circuitos también porque a veces necesito garantía de performance a un costo alto. (Recursos dedicados).

- Protocolos diferentes: Conjunto de reglas preestablecidas, en LAN puedo usar un procedimiento determinado que quiera, en WAN tengo que utilizar un procedimiento, tomar en cuenta protocolos de detección y corrección de errores. Protocolos diferentes para etapas diferentes.

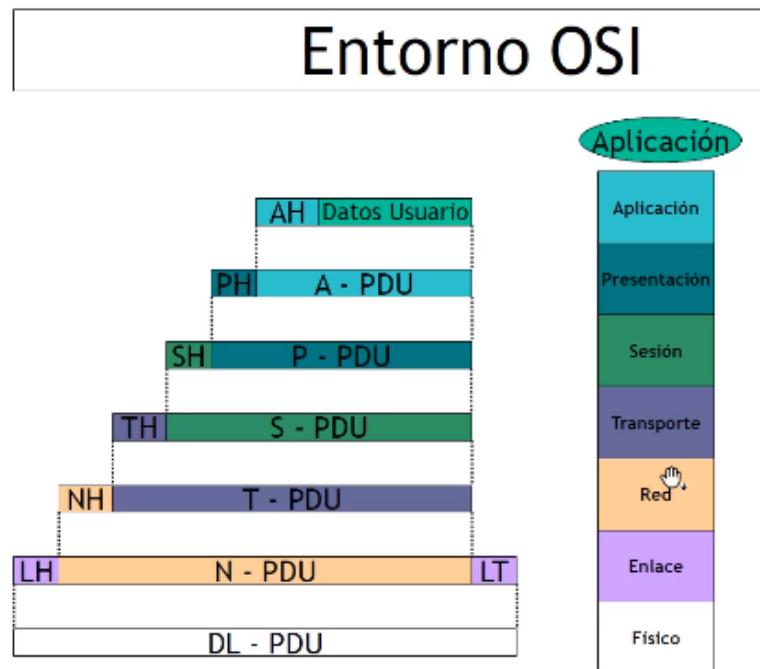
Modelo OSI:

- Open System Interconnection
- No propietarios - sistema que se puede conectar.
- Objetivo: Permitió que los dispositivos de distintos fabricantes se puedan comunicar entre sí.
- Modelo de referencia

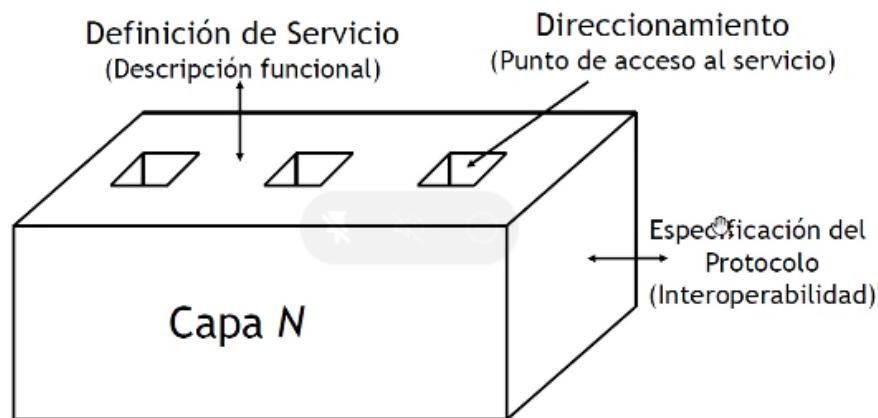
Características:

- Capas separadas para funciones diferentes
- Funciones similares dentro de la misma capa
- Interacción mínima entre capas
- Permite la implementación parcial

- Protocolos entre dos capas iguales, servicios entre capas adyacentes. Una capa brinda servicios a la capa superior y consume servicios de la capa inferior.
- Se ingresa por la capa 7 - Usuario.
- Protocolo de capa par entre cada una de emisor-receptor (ej; una comprime la otra descomprime)
- Cada capa agrega un header o cabecera de capa (Ej: Cabecera de aplicación). Este header sirve para que el receptor (la misma capa pero del dispositivo receptor) pueda a través de un protocolo en común obtener el mensaje original que tiene datos adicionales para el envío. Ej: Separar un mensaje en paquetes y mandarlo, necesitas agregar info de orden por ejemplo, que el receptor debe sacar para obtener solo el mensaje. Encapsulamiento: Lo que me pasa la capa anterior lo agrego a un header. La capa de enlace agrega un trailer para delimitar de los 1 y 0 dónde arranca y termina.



- Cada capa tiene puntos de acceso al servicio (SAP): Direcciónamiento.



Direcciónamiento: Software, cómo se invoca una función, cómo la capa superior

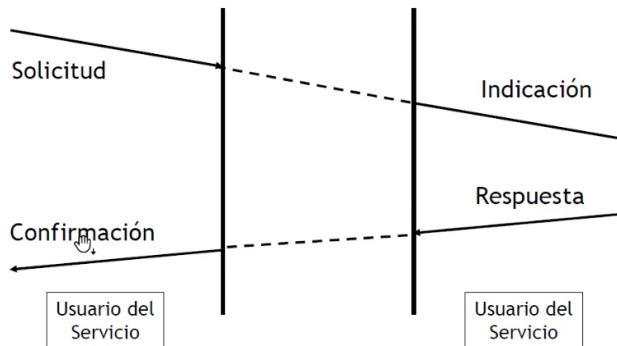
invoca a la inferior. Una descripción funcional y una especificación del protocolo (interoperabilidad, que hay en la cabecera de la capa N. qué se escribe, qué significa el contenido, etc)

Especificación del protocolo: Ej: comprimido (si/no).

- Para consumir un servicio entre capas: Emiten solicitudes a las capas inferiores y si se espera una respuesta se utilizan confirmaciones:

Primitivas de Servicio

- Utilizadas para la comunicación entre capas adyacentes. Tienen parámetros asociados.



- Para cada capa hay características que debe cumplir

Capa Física:

- Encuentro Bits
- Protocolo de capa par determina que códigos usar, de banda base como Manchester, HBN 3, Manchester diferencial, etc o modulación. Determina el medio de comunicación a usar. Determina en los dispositivos que significa un uno y un cero, el amperaje, el voltaje, etc.
- Encuentro un medio de comunicación (alámbrico, eléctrico, una señal de radio frecuencia o fibra óptica).
- No hay control de errores porque no se puede dar cuenta si un 1 o 0 está bien.

Capa Física

Define la interfaz física entre dispositivos y reglas para la transmisión de bits

- **Mecánicas:** Propiedades físicas de la interfaz y del medio de comunicación
- **Eléctricas:** Niveles de tensión, velocidad de transmisión. Sincronismo de bit.
- **Funcionales:** Funciones que realiza cada circuito entre el sistema y el medio de comunicación
- **De procedimiento:** Secuencia de eventos para el intercambio del flujo de bits

Ejemplo: EIA-232-F, ISDN, LAN

Capa de enlace:

-Debe delimitar las tramas, se encarga de determinar de todos los unos y ceros que llegan donde empieza y termina el mensaje. Porque de lo que se manda hay bits auxiliares y esta capa descarta todo eso y se queda solo con el mensaje.

-Además garantiza entrega libre de errores, ejecuta mecanismos de detección y corrección de errores. Entre el emisor y el dispositivo siguiente (por ejemplo entre nodos).

Por ejemplo: Le calcula un CRC al mensaje y se lo adosa al envío.

El header y el trailer permite delimitar el comienzo y fin del mensaje.

Capa de Enlace

Intenta brindar un enlace seguro y provee mecanismos para activar, mantener y desactivar el enlace.

- Delimitación del flujo de bits
- Detección y corrección de errores
- Control de flujo
- Recuperación de datos perdidos, duplicados o erróneos.

Ejemplo: HDLC, LAP-B, PPP

Control de flujo: Parar de emitir o parar de recibir.

Capa de red:

Debe encontrarle un camino al mensaje, tiene un origen y destino.

Indica por ejemplo: Mi comunicación es con GMAIL si quiero abrir esa página y no con Fibertel.

Me permite ir más allá de quien tengo conectado adyacentemente.

Ej: IP, IPX, X.25

Capa de Red

- Funciones de conmutación
- Encaminamiento
- Oculta a las capas superiores los detalles de la red subyacente (paquetes/circuitos)
- Gestión de prioridades
- Interconexión de redes

Capa de Transporte:

Control de errores de extremo a extremo (emisor y receptor y no nodos como capa de enlace porque puede ser que uno de los nodos no haga detección, por ejemplo el cable modem, el router, etc.).

Capa de Transporte

Provee mecanismos para el intercambio de datos
Extremo a Extremo

- Familia de 5 estándares, cada uno especificado para un determinado servicio
- El servicio orientado a la conexión asegura la información libre de errores, en orden, sin pérdidas ni duplicaciones
- Proporciona la calidad de servicio solicitada por la capa de Sesión

Ejemplo: TCP, SPX

Capa de sesión:

Detecta si la comunicación sigue activa, si se cayó, si hay que reiniciarla, etc.

Capa de Sesión

Los mecanismos descriptos en esta capa suelen implementarse en la capa 7 (Aplicación)

- **Control de diálogo:** Solicitud de canales simultáneos (full-dúplex) o alternados (half-dúplex)
- **Recuperación:** Procedimientos de puntos de comprobación para recuperación de fallos e interrupción de operaciones

Capa de Presentación:

Genera compresión, indica formato de compresión para que el receptor pueda descomprimir el mensaje.

Capa de Presentación

Define el formato de los datos que van a intercambiarse.

- **Conversión de códigos:** Adaptación de diferentes códigos utilizados por los extremos (por ejemplo: ASCII, EBCDIC, etc.)
- **Compresión:** La compresión de los datos se realiza a este nivel.
- **Encriptación**

Capa de aplicación:

Toma los datos de usuario y agrega una cabecera de aplicación (Lo que necesita la capa de aplicación del receptor para captar el mensaje original a través del envío).

Capa de Aplicación

Proporciona a los programas de aplicación un medio para acceder al entorno OSI

- Incluye funciones de administración general y los mecanismos para la implementación de sistemas distribuidos
- A esta capa pertenecen las aplicaciones de uso general: Transferencia de archivos, correo electrónico, acceso a terminales remotos, etc.

Ejemplo: Telnet, FTP, SMTP, etc.

20/8

Redes LAN:

- Hoy día hay una red de facto (única) → Ethernet.

Ethernet:

Formada por varias empresas y estandarizada.

Ethernet

- Comité IEEE 802.3
- Peer to Peer
- CSMA/CD (Carrier-Sense Multiple Access / Colision detection)
- Half-duplex

- Peer to peer: todos los miembros de la red son iguales, no hay más ventaja o mayor prioridad.

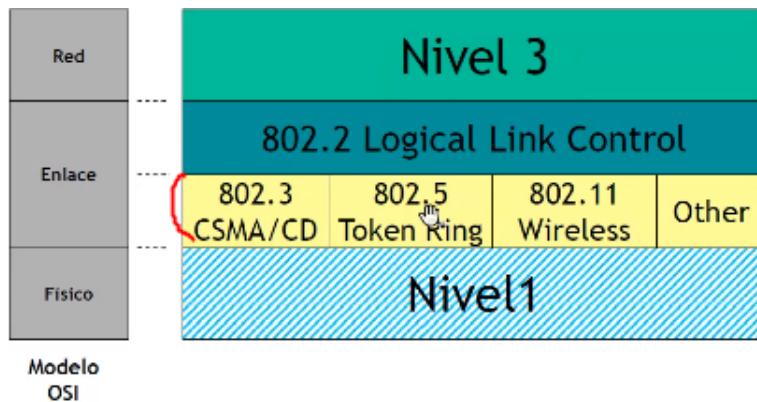
- Redes de acceso aleatorio porque no sabe cuando va a poder emitir.

- Opera con mecanismos de control de acceso al medio (CSMA/CD - Acceso múltiple por detección de portadora con detección de colisiones) → habitación con una mesa y gente que quiere hablar (ejemplo de resumen de final).

- Con los años los estándares de ethernet fueron aumentando (10Mbps, 100Mbps... 10Gb) pero perdura la estructura.

- No pueden emitir más de uno a la vez, se aplica TDMA (transmisión por división de tiempo) en esta primera versión es half-duplex.

- No existe confirmación

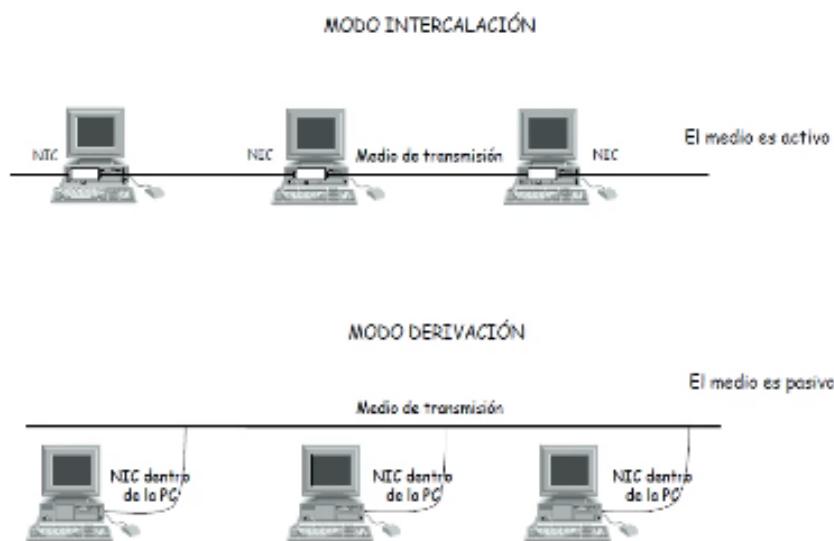


Capa de enlace formada por:

1. Capa MAC: Control de acceso al medio
2. LLC: Logical Link Control

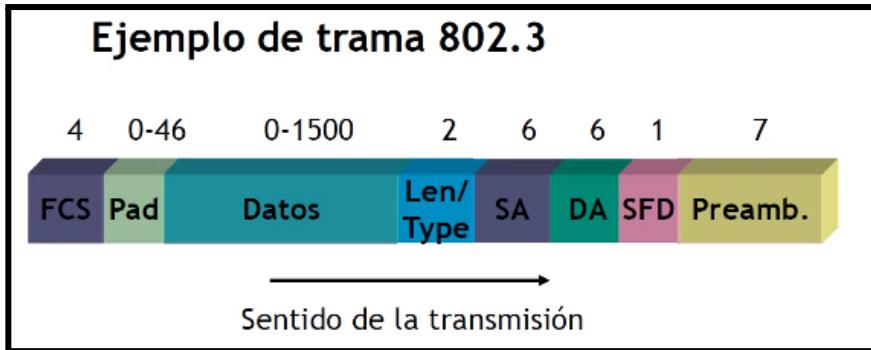
Modos de conexión de terminales:

Ya no es más un bus físico con cable coaxial sino cableado estructurado (estrella, se concentra en el gabinete) que utiliza topología estrella. Aunque sigue operando en la modalidad de la imagen, hay un único medio por donde pasa la info.



- Conexión punto - multipunto: Me obliga a usar direccionamiento ya que hay muchos equipos en el mismo medio, identificar quién envía y quién recibe. Cuando una transmite todas las demás escuchan. Utiliza la siguiente trama:

Ejemplo de trama 802.3:



En este caso usa el código de línea Manchester (balancea unos y ceros, recupera sincronismo y me permite detectar la transmisión en la línea).

- **Preamb:** Es una secuencia para que se sincronicen las estaciones. 7 octetos. (10101010)
 - **SFD:** Delimitador de comienzo de trama y fin de preámbulo. (secuencia 010101011)
 - **DA:** Dirección destino, a quién le interesa leer el mensaje.
 - **SA:** Dirección origen. Quién lo envía.
- DA y SA : Son direcciones MAC, todo dispositivo por su fabricante tiene su dirección MAC particular.
- **Len / Type:** Tiene un significado en el estándar ethernet (nos indica que hay dentro del campo de datos, por ejemplo si es ipv4(0800) o v6(86),0806 → ARP, quién me lo envió y quién lo debe recibir) y otro en el 802.3 (indica cuál es la longitud del campo de datos que es variable). Todos los etherype (ethernet) son mayores a 1536
 - **Datos:** Longitud variable, pero tiene una longitud máxima de 1500 bytes por estándar.
 - **Pad:** Opcional y de relleno. Para asegurarnos de llegar al mínimo de longitud (64 bytes) y que no se generen colisiones entre estaciones.
 - **FCS:** CRC, controla la presencia de errores.

- Preamble (7bytes) : 10101010
- SFD (Start of Frame Delimiter) - 10101011
- DA (Dirección destino) - 6 bytes
- SA (Dirección Origen) - 6 bytes

Direcciones de 48 bits formadas por:

- OUI: Organizationaly Unique Identifier (3 bytes)
- DUI: Device Unique Identifier (3 bytes)

Direcciones especiales:

- 0xffff.ffff.ffff : Broadcast
- 0x0000.5e00.0000 - 0x0000.5eff.ffff : Multicast

Aclaraciones de la imagen:

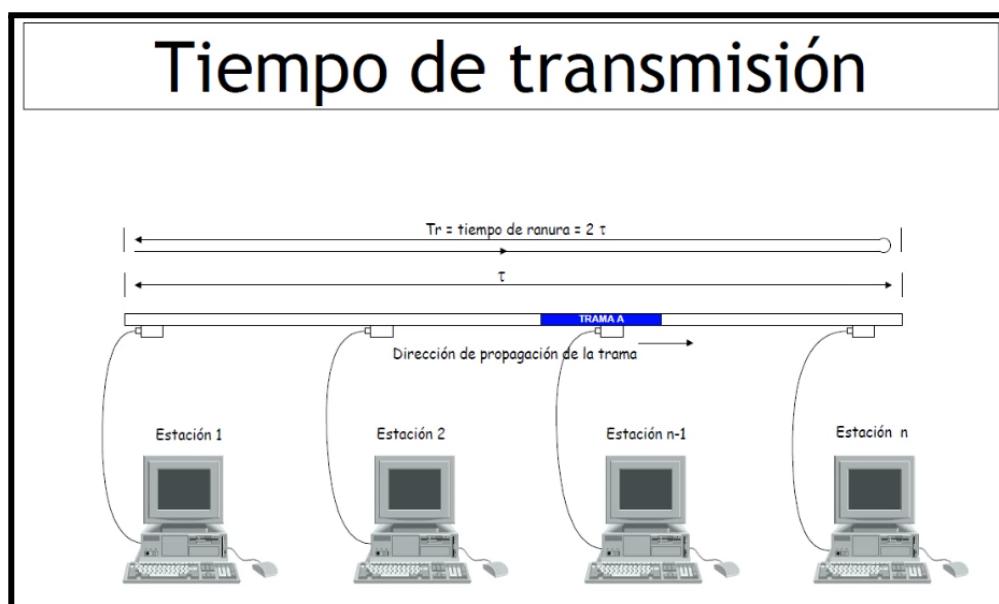
- OUI es de la marca. Ej: Intel. DUI: del aparato.
- Broadcast: El mensaje está dirigido a todos los dispositivos de la red. Difusión. Es de 1 → a todos. (48 unos seguidos o FFFF en hexa).

- Multicast: De 1 → a algunos. Un grupo de destinatarios.
- Unicast: De 1 → 1.

Longitud de la trama:

La trama posee una longitud mínima de 64 bytes y una longitud máxima de 1518 sin incluir preámbulo ni SFD.

Forma y tiempo de transmisión de un mensaje:



CSMA / CD (detectar colisiones):

- Las estaciones escuchan (lo que hay en el medio) y transmiten al mismo tiempo, transmiten si ven libre. Si ambas transmiten al mismo tiempo se generan colisiones. Cuanto más largo es el medio mayor es la probabilidad que dos estaciones piensen que esté libre y transmitan al mismo tiempo.
- Para controlar colisiones el estándar plantea una longitud máxima del medio (2500m con 4 repetidores). 5 segmentos con 4 repetidores máximo. 2500 metros total con segmentos de 500 m.

UN PROTOCOLO DE CAPA SUPERIOR SOLICITA ENVÍO →

- Envío el mensaje, colocó señal sobre el medio, se propaga, todos los receptores reciben el mensaje, chequean CRC y si está ok se fijan la dirección destino a ver si ellos son los destinatarios.
- Todas las estaciones escuchan todos los mensajes pero chequean a través de la dirección destino si son para ellos o no. Si no son, los descartan.

Por qué la longitud mínima:

- Cuando una estación comienza a transmitir, la señal se propaga con una velocidad de 10Mbits, voy a necesitar transmitir 512 bits (64bytes) para llenar el medio (llenar los 2500 metros). Los primeros 64 bytes son la ventana de colisión para que si mando un mensaje más corto otra estación no me colisione sin darme cuenta.

Si yo transmito 512 bits y nadie me interfirió significa que llegué a la otra punta del bus y nadie me va a colisionar. Necesito estar transmitiendo para darme cuenta si me colisionaron.

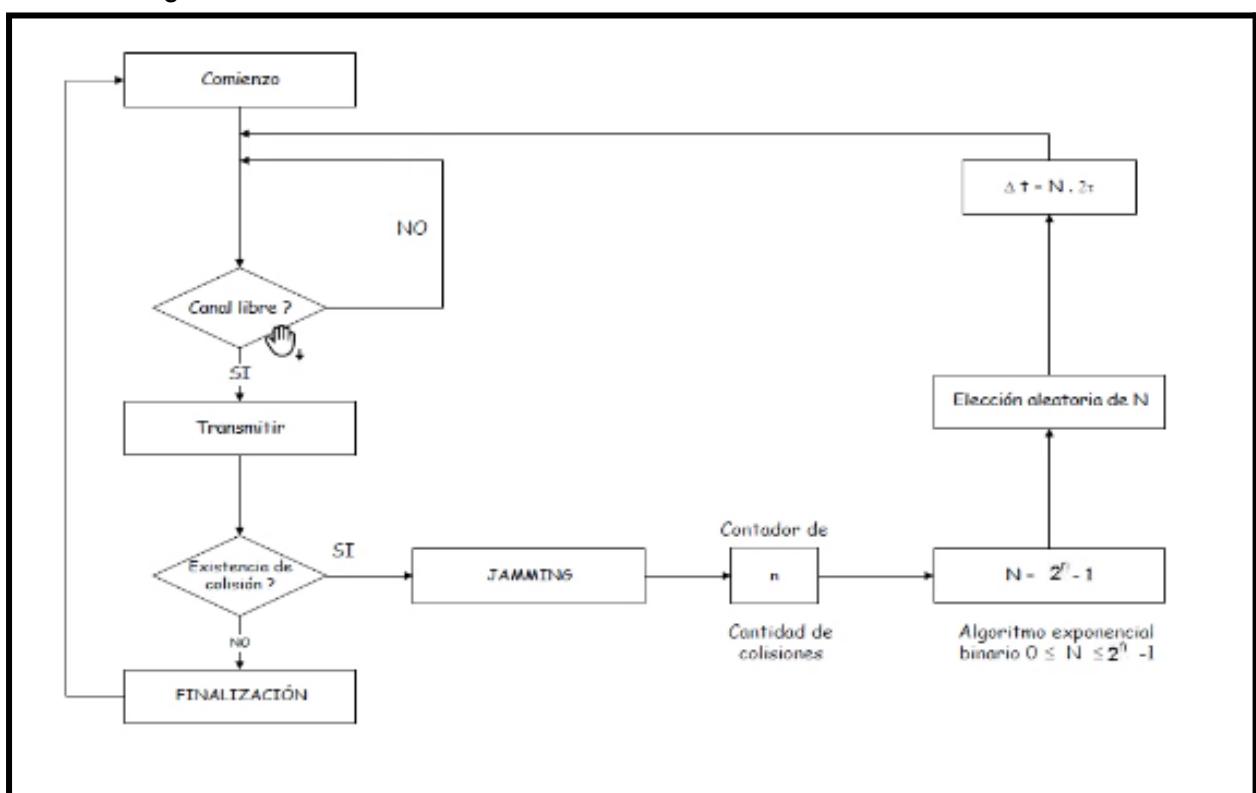
Necesito un mensaje mínimo de 46 bytes + 18 bytes de cabecera.

No confirmación de llegada:

En ethernet no existe confirmación de llegada. Si hay un error no lo corrige porque se maneja en entornos con alto ancho de banda y pocos errores, se puede asumir que casi siempre llega correctamente a destino.

Algoritmo de espera exponencial binario:

Permite arreglar las colisiones. El CD es detección de colisiones.



Jamminns: Si la estación detecta colisión debe mantener la transmisión a pesar de no ser exitosa, sostiene la colisión un tiempo para que todas las estaciones sepan de esa colisión, así no cae ninguna más.

Eigen un tiempo random de espera para volver a intentar.

Disminuye exponencial: Cuanto más colisiones más grande es n y más chica es la probabilidad de que vuelvan a coincidir en el tiempo (vuelvan a colisionar).

n = hasta 10.

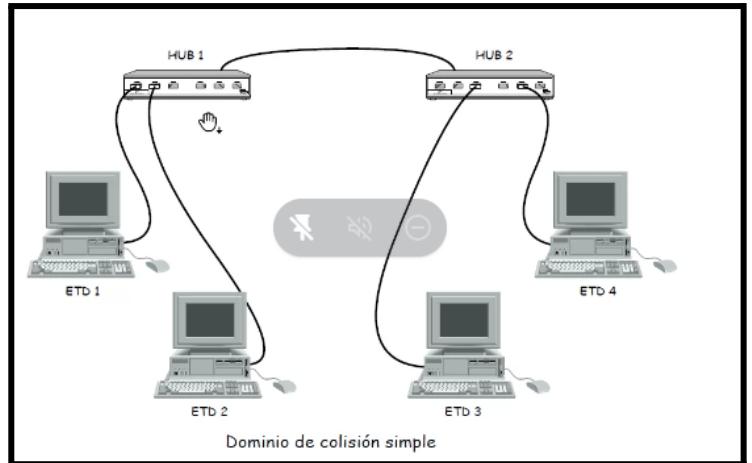
Y el estándar indica que se intenta hasta 6 veces más, o sea hasta 16 intentos de envío → si llegan a ese nro y no hay envío exitoso → se notifica que no hay conexión a la red a la capa superior.

Problemas de topología bus:

Cable coaxial con todo conectado, resultó poco práctico porque tendrías que tender coaxial a todas las estaciones de trabajo, además, solía ser inestable. También si se perdía la carga en algún extremo se cortaba la red, y si se desconectaba un coaxial de alguna estación, se caía la red. Osea que no podías mover los equipos porque tirabas la red.

Topología física de estrella:

Se dejan los cables coaxiales y se usan cables UTP conectados a un HUB (conecta eléctricamente a todos sus puertos). Opera lógicamente como un bus porque el HUB cuando una estación transmite todos los demás puertos lo reciben, no se puede transmitir más de uno a la vez. Compiten por el uso del canal.



HUB - REPETIDOR:

-Inteligencia de capa 1, capacidad eléctrica, básicamente conecta equipos como un bus.

Lo que se transmite se propaga eléctricamente a todos los puertos.

Cuando uno transmite todos escuchan en cambio un switch puede tener escuchando y transmitiendo muchos a la vez.

-Puede ser un switch o cualquier otro dispositivo que físicamente es igual pero tiene distintas características.

-Opera lógicamente como un bus.

-Podés conectar HUBS con un cable cruzado (uplink) si te quedas sin puertos.

Dominio de broadcast:

Cuando una estación envía un msj se envía a todos los que están conectados a ese dominio de broadcast.

Dominio de colisión:

Todas las estaciones en el segmento de red compiten por los 10 Mbits por seg., van a competir entre ellos para ver quien usa eso.

Red de acceso aleatorio:

Característica de ethernet donde una estación desea transmitir y no sabe cuándo lo hará si está ocupado, si está libre intenta pero puede colisionar, no hay certeza.

Configuraciones históricas:

- **10Base5** 10Mbps, transmisión en banda base, 500m. Longitud máxima del segmento. Coaxil grueso RG-218
- **10Base2** 10Mbps, transmisión en banda base, 200(185)m. 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)

Aclaraciones a sniffer de red:

Los octetos se transmiten con el bit de menor peso primero. El último bit del octeto se transmite primero. Tiene un significado i, en cero es una dirección individual, si es 1 es una dirección de grupo (broadcasts o multicast).

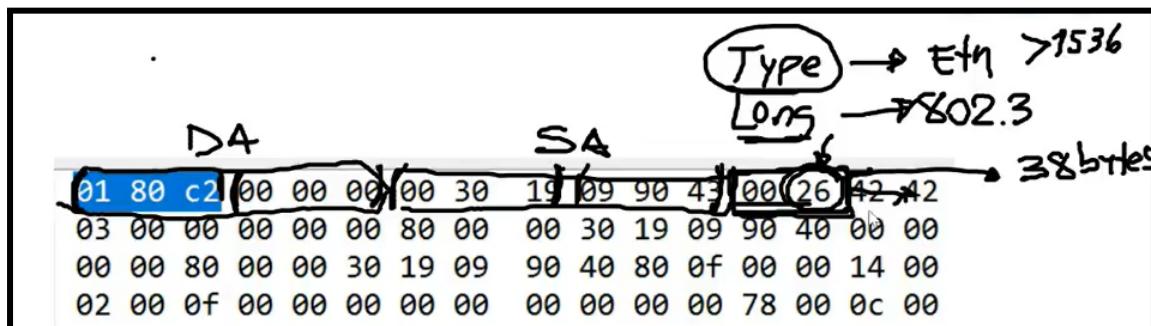
El siguiente bit que se transmite es L, si es cero la dirección es única (mac asociada a la placa), si altero/falsifico el bit se setea a 1. (Ejemplo cuando creo máquinas virtuales donde la IEEE no me da la dirección MAC, la creo yo).

```

Ethernet II, Src: Sagemcom_T7:cc:ef (6c:99:b1:t7:cc:ef), Dst: IntelCor_34:c5:92 (3c:f0:11:34:c5:92)
  Destination: IntelCor_34:c5:92 (3c:f0:11:34:c5:92)
    Address: IntelCor_34:c5:92 (3c:f0:11:34:c5:92)
      .... ..0. .... .... .... = LG bit: Globally unique address (factory default)
      .... ..0. .... .... .... = IG bit: Individual address (unicast)
  > Source: Sagemcom_f7:cc:ef (6c:99:61:f7:cc:ef)

```

Así viene en el parcial:



- DA = Dirección destino con DUI al principio
- SA = Dirección origen
- Type o Long dependiendo el tipo, todos los ethertype son > a 1536. Si esos dos bytes son menores a 1536 es de tipo 802.3 (longitud)

En 802.3: Si viene 802.3 viene 802.2 (vienen de la mano).

La dirección de grupo (broadcast o multicast) sólo puede ser dirección destino.

| |
|--|
| ✓ Destination: Broadcast (ff:ff:ff:ff:ff:ff) |
| Address: Broadcast (ff:ff:ff:ff:ff:ff) |
|1..... = LG bit: Locally administered address (this is NOT the factory default) |
|1..... = IG bit: Group address (multicast/broadcast) |
| 10 ff ff ff ff ff 00 12 79 be 4a 66 08 06 00 01 v 1E |

| |
|---|
|1..... = IG bit: Group address (multicast/broadcast) |
| ✓ Source: HewlettP_be:4a:66 (00:12:79:be:4a:66) |
| Address: HewlettP_be:4a:66 (00:12:79:be:4a:66) |
|0..... = LG bit: Globally unique address (factory default) |

Y si fuera 1, es local y no es única, no la asignó la IEEE (Ej; máquinas virtuales).

Como ARP (0806) se envía SOBRE ethernet y son solo 28 bytes se necesitó adicionar un padding para alcanzar 64 bytes mínimos para que funcione correctamente el protocolo.

| No. | Time | Source | Destination | Protocol | Info |
|------|-----------|-------------------|-------------|----------|--|
| 426 | 1.873703 | Sagemcom_f7:cc:ef | Broadcast | ARP | Who has 192.168.0.15? Tell 192.168.0.1 |
| 537 | 2.897734 | Sagemcom_f7:cc:ef | Broadcast | ARP | Who has 192.168.0.15? Tell 192.168.0.1 |
| 672 | 3.922356 | Sagemcom_f7:cc:ef | Broadcast | ARP | Who has 192.168.0.15? Tell 192.168.0.1 |
| 1304 | 7.508512 | Sagemcom_f7:cc:ef | Broadcast | ARP | Who has 192.168.0.15? Tell 192.168.0.1 |
| 1529 | 8.533382 | Sagemcom_f7:cc:ef | Broadcast | ARP | Who has 192.168.0.15? Tell 192.168.0.1 |
| 1691 | 9.553531 | Sagemcom_f7:cc:ef | Broadcast | ARP | Who has 192.168.0.15? Tell 192.168.0.1 |
| 2608 | 11.571213 | Sagemcom_f7:cc:ef | Broadcast | ARP | Who has 192.168.0.15? Tell 192.168.0.1 |

```

> Frame 426: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{DDE36802-8205-4F68-9EAA-51BBA43
✓ Ethernet II, Src: Sagemcom_f7:cc:ef (6c:99:61:f7:cc:ef), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: Sagemcom_f7:cc:ef (6c:99:61:f7:cc:ef)
    Type: ARP (0x0806)
    Padding: 00000000000000000000000000000000
  > Address Resolution Protocol (request)
<

0000 ff ff ff ff ff 6c 99 61 f7 cc ef 08 06 00 01 .....1.a.....
0010 08 00 06 04 00 01 6c 99 61 f7 cc ef c0 a8 00 01 .....1.a.....
0020 00 00 00 00 00 00 c0 a8 00 0f 00 00 00 00 00 00
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

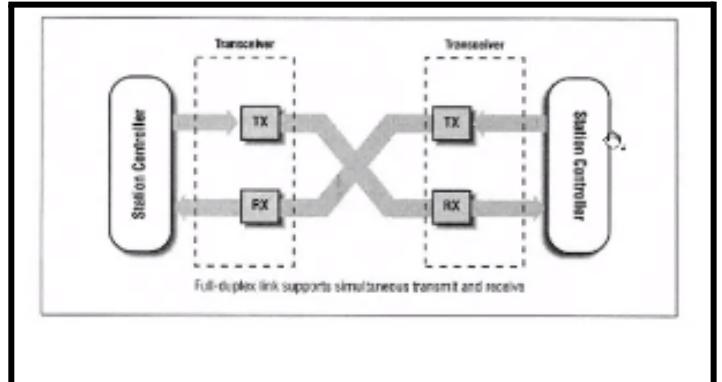

```

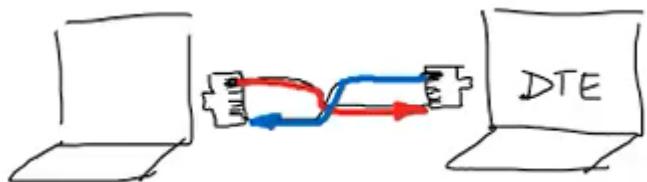
26/8

Interfaz Full-Duplex:

- Caminos de transmisión y recepción independientes que operan en simultáneo.
- Dos estaciones conectadas punto a punto con un vínculo full duplex.
- No hay contención, se elimina el CSMA/CD, pueden escuchar por un lado y recibir por otro. Porque el camino siempre está libre (son 2 pcs) y no un dominio punto multipunto.

Si conecto la pc con un EDT tengo todos los puntos iguales, si los cruzo tengo caminos independientes y puedo transmitir y recibir a la vez.





Bridge:

Dispositivo que los distintos segmentos operan como si no estuviera. Opera pasando tráfico como si 2 segmentos estuvieran directamente conectados y va aprendiendo quién está en cada extremo.

Para mejorar la eficiencia de los 10mbits del bus

-Se debió a que a más dispositivos conectados menos eficiencia había porque se concentraban en arreglar colisiones. Para solucionarlo surgió el dispositivo bridge

-La idea es, las estaciones generan tráfico, el tráfico local de un segmento quede en ese segmento y no pasa por el otro lado del bridge.

-Hace 2 operaciones:

- Filtrado de paquetes: Si se mandan msj entre elementos de un segmento, el bridge se da cuenta y los ignora (no los pasa al otro segmento).
- forward (envío de paquetes).

Si debe enviar una trama a una MAC address que desconoce la debe mandar igual porque opera de forma transparente.

-Dispositivo transparente → no puede interferir en el comportamiento protocolo si un segmento manda un broadcast tiene que mandarse a todos como si no estuviera.

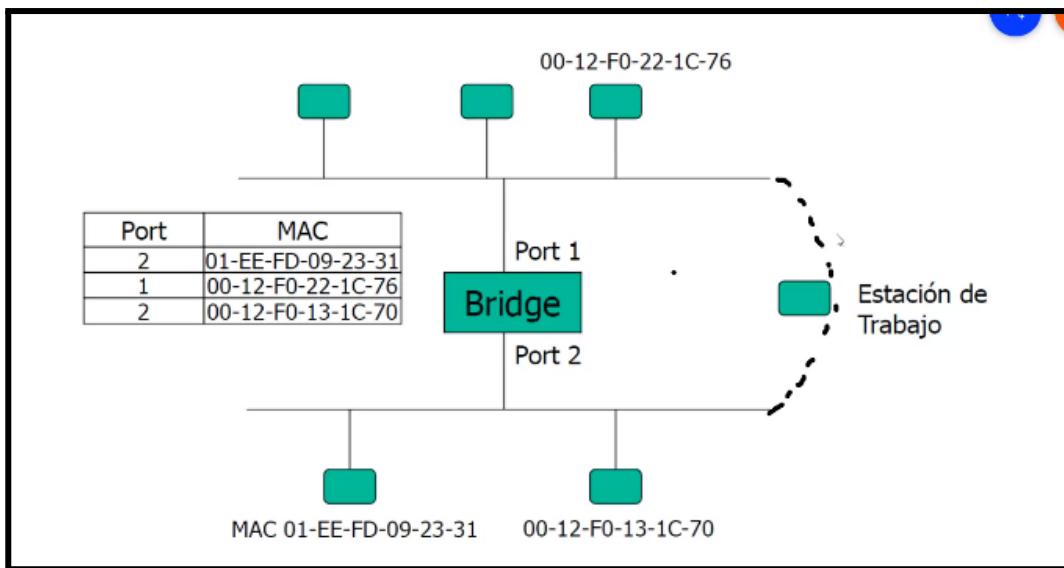
-Los mensajes unicast (entre dos elementos de un mismo segmento) → El bridge almacena tramas, lee la dirección MAC destino, si está en el mismo segmento descarta/ignora la trama.

-No existe más.

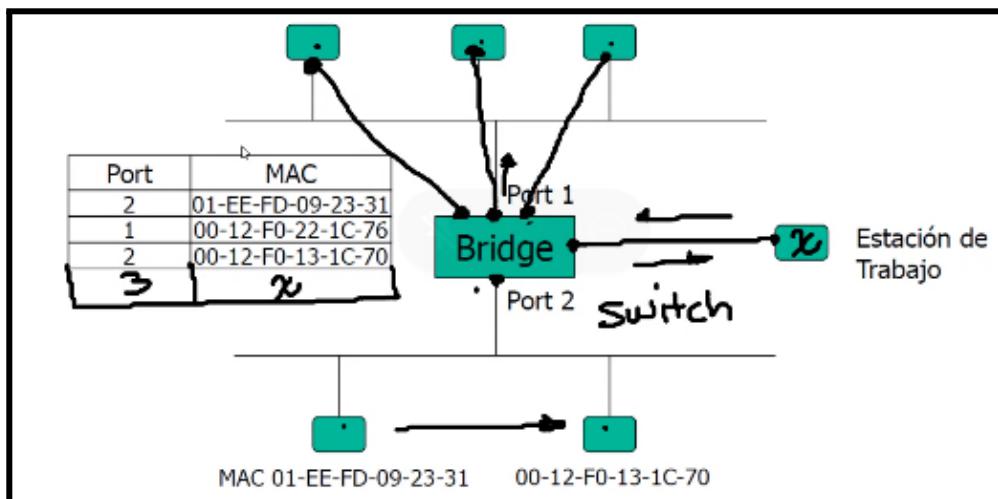
Beneficios:

Tiene 2 interfaces CSMA/CD (CADA INTERFAZ CONFORMA CON EL BUS DE SU LADO UN DOMINIO DE COLISIÓN) → tiene doble bus (uno en cada extremo con sus propios 10MbitsxSeg) → Aumenta la eficiencia.

Hay 2 dominios de colisión pero sigue habiendo 1 dominio de broadcast



En vez de que las interfaces (estaciones de trabajo) se conecten a un bus (ejemplo un hub al puerto del switch) y el bus al bridge se conectan directamente, ahora transmiten y enviar, se convierten en full-duplex → se llamaría switch.



Deja de usar CSMA/CD, usa interfaces full-duplex, ningún dispositivo compite por el medio con otros.

Switch:

- Hace bridging. Vincular cada puerto con una dirección mac
- Puedo tener conexión simultánea (una estación enviando trama a una estación y otras dos haciendo lo mismo, no compiten).
- Elimina los dominios de colisión.
- Tiene capacidad de conmutación haciendo creer a la estación que tienen toda la capacidad del enlace

| Port | MAC |
|------|-------------------|
| 2 | 01-EE-FD-09-23-31 |
| 1 | 00-12-F0-22-1C-76 |
| 2 | 00-12-F0-13-1C-70 |
| 3 | x |

Teoría:

Cada puerto del switch es un dominio de colisión independiente → es cierto si existe un hub conectado a cada puerto del switch.

Bridging:

Bridging

Transparent Bridge

- Operan en el nivel 2 y utilizan las direcciones MAC para encaminar las tramas.
- Aprenden automáticamente la ubicación de los hosts.
- Las tramas soportan dos procesos. "Filtering" y "Forwarding"

Translating Bridge

- Realiza además conversión de protocolo y velocidad

Modos de operación:

1. Cut-Through:

Basta con leer los primeros 6 bytes de la trama para tomar una decisión y conmutar la trama. Que es la dirección MAC destino

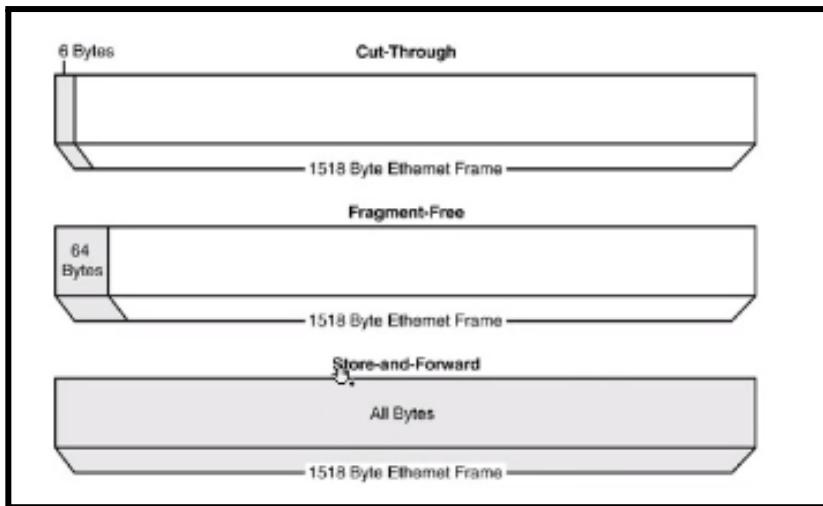
Inconvenientes: Se produce una colisión mientras se está pasando la trama, y no va a poder terminar de pasarla.

2. Fragment-Free:

En vez de esperar a recibir los primeros 6 bytes, espero a recibir los 64 bytes (porque la colisión sólo se puede dar en los primeros 64 bytes de la trama si cumplimos la premisa del largo máximo, si es más largo ahí sí pueden haber colisiones). Porque se llena toda la longitud de la línea con la señal entonces no puede haber colisión ya que sabrías si hay otro transmitiendo.

3. Store-and-Forward:

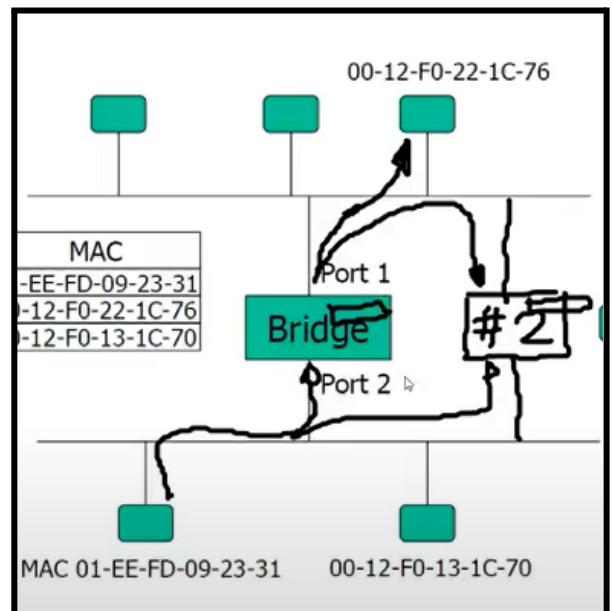
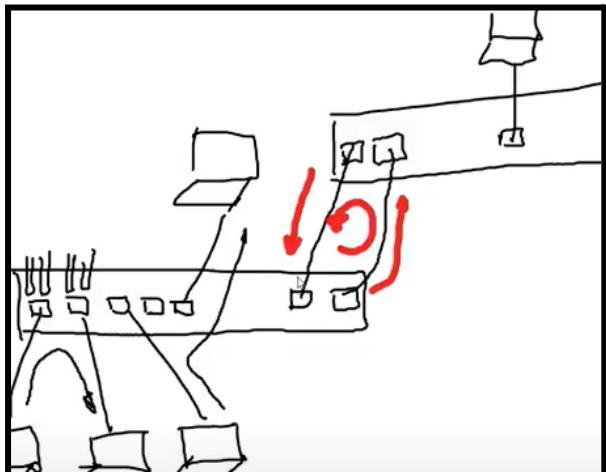
El switch espera a recibir la trama completa, calcula el FCS y si da bien lo transmite al MAC que indica la trama.



Para que el bridge no sea un único punto de fallo (si se cae no hay conexión) → se agrega uno redundante -> esto genera:

Problemas de redundancia:

Genera **loop de capa dos** porque no se conocen entre sí (el broadcast va por un cable y vuelve por el otro, y toma todos los recursos) → SPA/STP - Spanning Tree para arreglarlo.



STP - Spanning Tree:

Los bridge ejecutan el protocolo, periódicamente envían una trama que indican que son un bridge, delatan su presencia y se reconocen. La idea del algoritmo es detectar un loop de capa dos, bridge loop y desactivar un puerto lógicamente.

No funcionan a la vez sino que uno toma el poder del otro si se cae, esta caída se nota si un bridge deja de emitir tramas de bridge. Antes de eso, el redundante desactiva su puerto lógicamente.

Los bridges se designan en un root y demás leafs (hojas). En caso que un link se desconecte se activa nuevamente el STA, TODOS los bridges tienen que participar del proceso de elección de root.

Se envían BPDU cada 2 segundos para anunciarse (los switches mandas BPDU por cada puerto), según el Bridge ID (mac address + bridge priority). El que tenga menor prioridad se designa root, si la prioridad es la misma desempata el mac address.

Los cambios de color del switch son los cambios de estado, si se conecta un nuevo switch puede cambiar la topología completa, se vuelve a hacer SPA. Todos los switches en un principio dicen ser root. Mandan sus Bridges IDs al resto y deciden cuál es menor.

Cuando se compara un switch con otro switch y determina que el otro es menor designan su root port, el puerto que te lleva al root y siempre está activo. Además le cuenta a los otros con el BPDU que está a una distancia de 1 de root y quien es root. Cuando esa info le llega a otro comparan bridge priority, si el nuevo es mayor designa el root port hacia el bridge que le envió el bpdu, si el root se conecta después con este último switch tiene que darse cuenta que la distancia a root es inferior por ese lado, por lo tanto anula el anterior post y designa ese nuevo como root.

802.1d Spanning Tree

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

- Descubre loops y desactiva vínculos redundantes
- En caso que un link se desconecte, se dispara nuevamente el STA, para activar el link desconectado por el STP.
- Todos los bridges(switches) en una red participan del proceso de elección del root.

Protocolo STP

- Se envían BPDU cada 2 segundos
- Todo switch tiene un Bridge ID (8bytes) compuesto de :
 - Bridge priority
 - MAC address
- La prioridad menor se designa ROOT
- Cuando cambia el estado de un port, se envian notificaciones de cambio de topología (TCN)y comienza nuevamente el calculo del arbol.

BLOCKING->LISTENING->LEARNING->FORWARDING

VLAN:

LAN Virtual, tener una LAN virtualmente dentro de una LAN física, mismo dispositivo físico permite partirse lógicamente. El switch permite asignar puertos a LANS virtuales como si estuvieran físicamente separados.

Los dominios de Broadcast pasan a ser los puertos designados para la misma VLAN. Una VLAN es un dominio de broadcast.

Hay maneras de designar VLANS por:

- MAC ADDRESS: solo pueden estar en la VLAN los que sean parte de una lista de VLANS. Si bien ofrece MAYOR SEGURIDAD, también requiere MAYOR ESFUERZO administrativo. Disp. que sin importar a qué puerto se conecten pertenecen a cierta VLAN.
- Puertos: los equipos que se conecten a ciertos puertos pasan a ser integrantes de la VLAN.

- Son creadas dentro de un mismo switch con esta facilidad
- Divide dominios de broadcast
- Aisla las redes
- Se requiere de un dispositivo de nivel 3 para interconectar las VLANs

Interconexión de VLANS - Router:

Para conectar puertos de dos VLANS distintas, necesito un Router (equipo con protocolo de capa 3). Ya que las VLANS son redes a nivel de capa 2, no pueden verse entre sí.

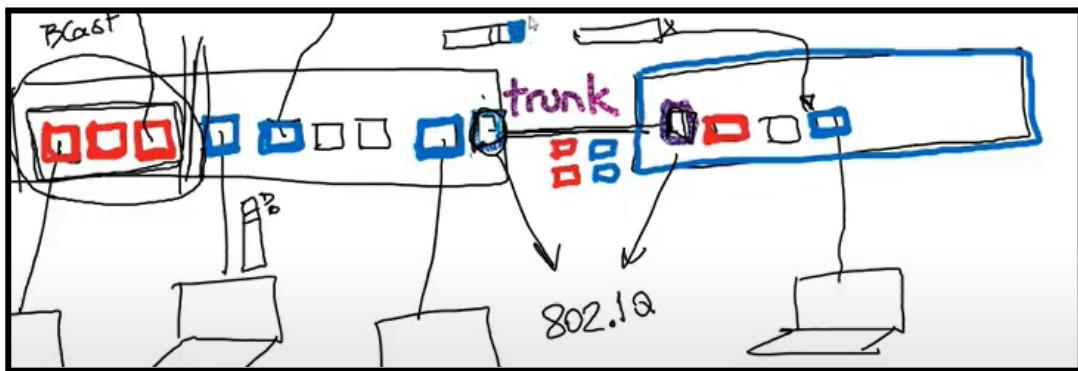
Ej: No se puede mandar una trama del disp. de VLAN1 con la MAC del disp. de VLAN2 porque son de capa 2, pero si usando una dirección de capa 3 para el disp. origen y para el disp. destino.

VLANS de != switches - puerto trunk:

Podemos designar miembros de la misma VLAN en distintos Switches.

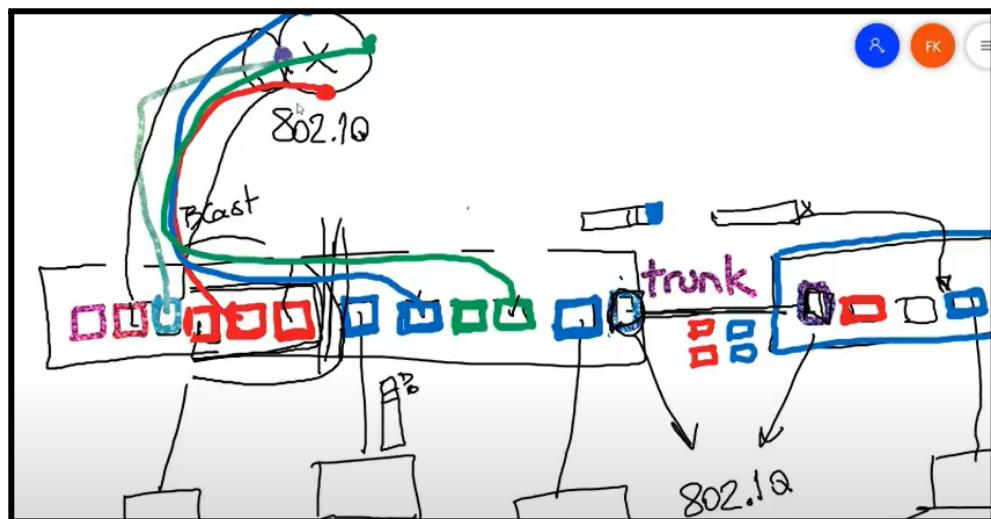
Se conectan los switches con un puerto trunk que no pertenece a ninguna VLAN.

Se pueden enviar paquetes de cualquier VLAN entre Switches y se marcan las tramas a que VLAN pertenece con **802.1Q VLAN TAG** (sino no se marcan, ya que con los puertos es suficiente saber donde deben enviarse para un solo switch).



Asimismo, si queremos que distintas VLANS de distintos Switches se conecten, necesitamos conectar un puerto TRUNK al router, y el router al otro Switch.

Tengo un puerto físico al cual conecto lógicamente una interfaz lógica a la VLAN roja, otra a la azul, y otra a la verde. Físicamente es una sola interfaz pero como tengo 802.1Q activo puedo enviar paquetitos "de colores"



El puerto trunk permite extender las vlans pero siguen siendo aisladas (las verdes solo ven verdes y así) para que se vean necesitamos un router.

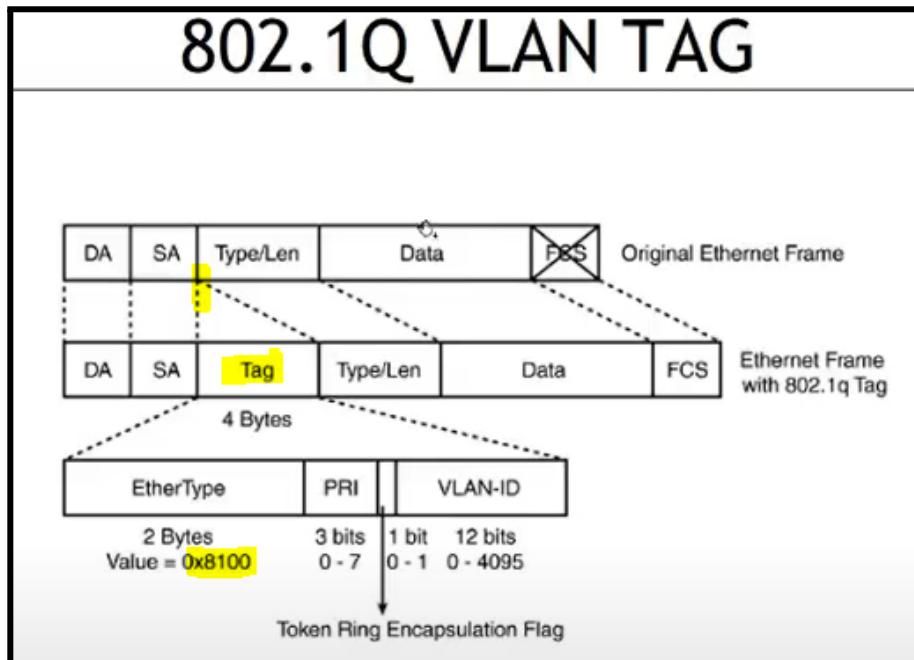
802.1Q VLAN TAG/ETIQUETADO:

Para que los paquetes de la misma VLAN puedan viajar entre Switches, se requiere un mecanismo para asignar el paquete a una VLAN (un flag). Este mecanismo le asigna una

etiqueta a la trama para enviarla, al recibirla se le quita la etiqueta y la pega en los puertos de esa VLAN.

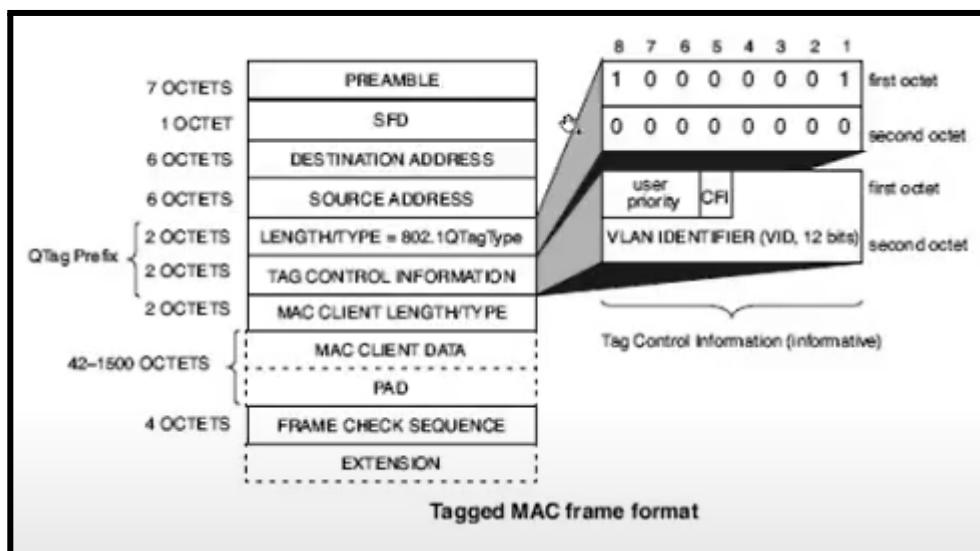
- Ethertype = 8100 → “es una etiqueta”
- PRI = Prioridad + compatibilidad para token ring
- VLAN-ID

Tengo 4096 identificadores de VLAN posibles para enumerar 4096 colores.



FCS cambia, se recalcula

Otro formato tomado de la recomendación de la IEEE:



Además del etiquetado entre switches lo puedo hacer entre un switch y el router. Al router le conecto una interfaz física al switch (trunk) que lógicamente actúa como si tuviera una

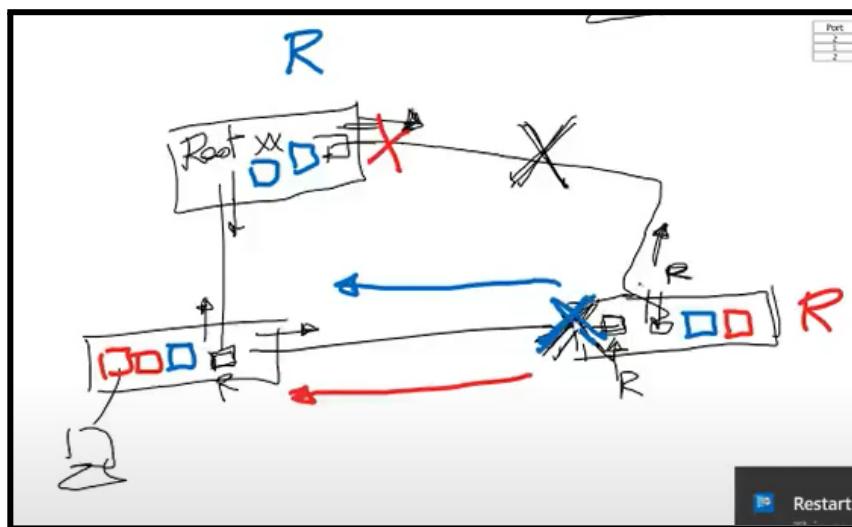
interfaz lógica a la interfaz roja, otra a la azul y otra a la verde. Sino necesitaría un router con múltiples bocas/interfaz para conectar cada vlan

PER VLAN SPANNING TREE:

Configurar instancias de Spanning Tree, una para cada VLAN. Obviamente configuraciones lógicas, no físicas. Se pueden bloquear puertos exclusivamente para Vlans.

Distinto root para distinta VLAN, así cursa tráfico por conexiones que para el otro spanning tree están bloqueadas y viceversa.

En la foto de abajo para los puertos de la VLAN roja se debería dar toda la vuelta porque se bloqueó el puerto de abajo. Habilito una nueva instancia para los rojos y una para los azules. Entonces un link que está lógicamente desactivado para un spanning para el otro no.



Dato:

Un puerto puede pertenecer SOLO a 1 VLAN, en el momento que los unis, unis el dominio de broadcast.

Si quiero que un dispositivo sea de 2 VLANS, configuro el puerto como TRUNK y habilito 802.1Q en el host.

27/8

Wireless LAN

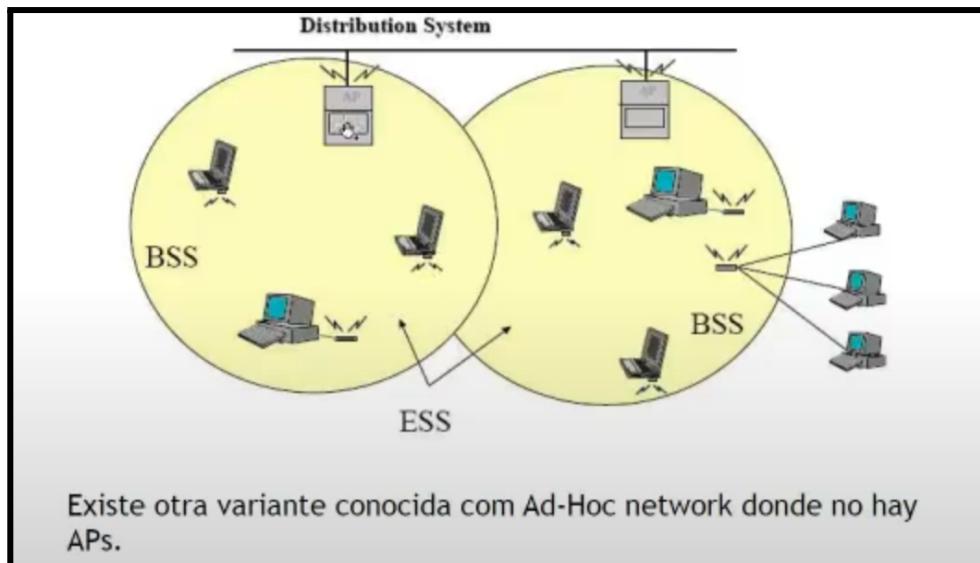
Estándar IEEE 802.11:

Elementos de la arquitectura:

- equipos terminales (un teléfono, una laptop, un televisor, etc).
- distribution system: la red lan cableada

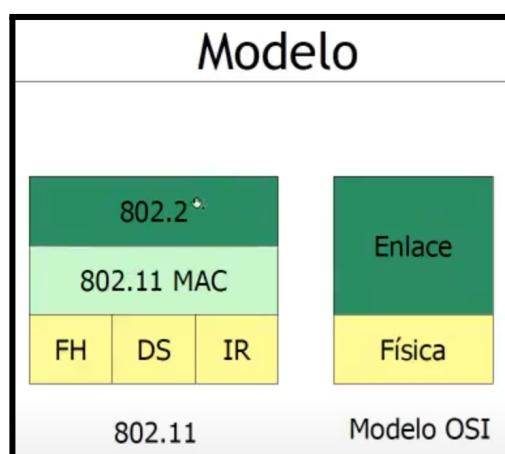
- Access point: Fundamental, donde se conectan los terminales, tiene una antena y cubre un área BSS. Lo que brinda el AP es un servicio de integración con el servicio de distribución (lo que está atrás). Básicamente nos permite conectar dispositivos wireless a nuestra red cableada.

Cuando tengo múltiples AP → configuro un ESS (extended) cubro más de una celda.



Ad-Hoc: 2 laptops cuando no hay access point, entre los dos negocian cual va a cumplir las funcionalidades de un access point.

Similitud a nivel CAPAS con MODELO OSI:



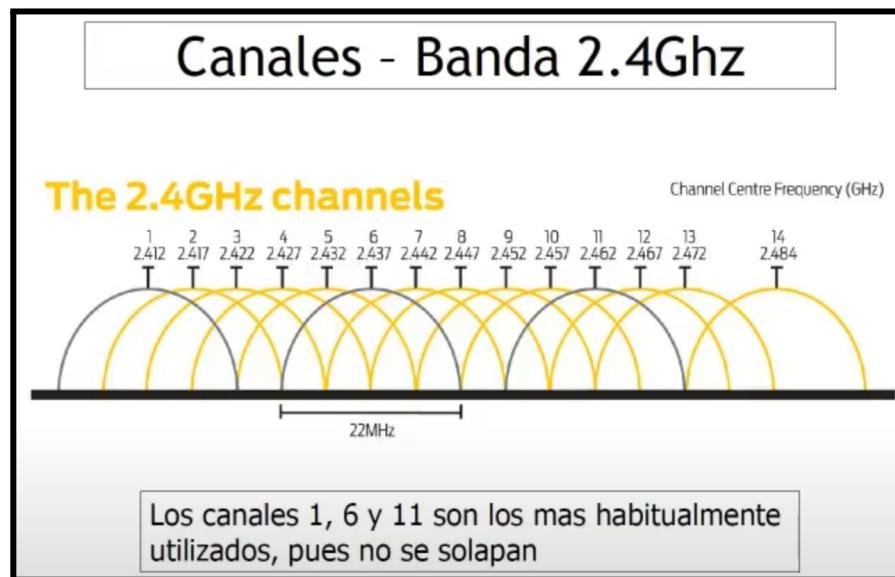
Dado que el aire es común a todos, hay que asegurarse que no interfiere al resto.

Bandas no licenciadas:

Existen 2 bandas de uso libre que no hay que solicitar permiso para su uso, cumpliendo ciertos requisitos para no interferir.

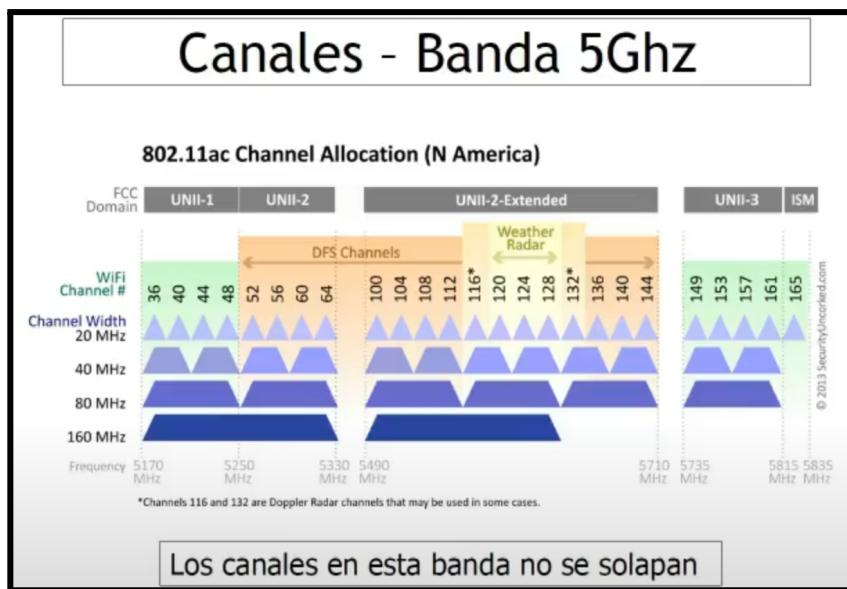
- Banda de 2.4 Ghz (disp. bluetooth, teléfonos inalámbricos, etc). Hay hasta 14 canales pero se solapan, menos el 1,6 y 11 con 22 Mhz

- Banda del 5.0 Ghz: Cada canal de 20Mhz y no se solapan



Menor atenuación, mayor alcance pero tengo menor velocidad en 2.4 Ghz

Mayor atenuación (por mayor frecuencia), menor alcance pero tengo mayor velocidad en 5.0 GHz

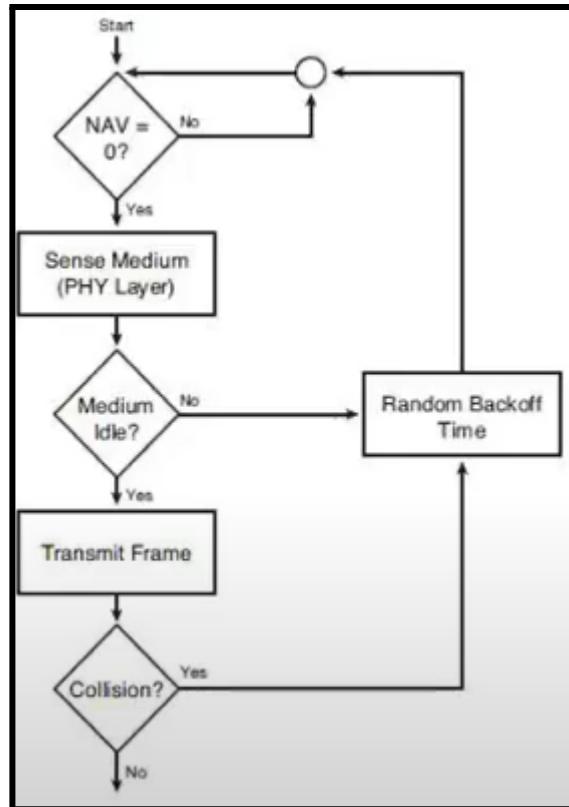


Control de acceso al medio:

Las terminales asociadas a un access point, como se ordenan para poder transmitir. En wireless, no conviene escuchar el medio porque en este hay mucha interferencia y no es tan fácil captar la señal (como en LAN con manchester diferencial). Cronológicamente DCF con RTS/CTS y luego derivó en DCF. Se intenta evitar la colisión más que detectarla como pasaba en ethernet

Diferencias con ethernet:

- NAV como ayuda para ver si está libre el medio
- ACK que confirma que no tiene ethernet.



DCF - Función de control distribuida

Distribuída porque todos deben hacerlo por su cuenta.

- DCF - Quiero transmitir:

Si la variable NAV nos dice que el medio está libre (=0) escuché el medio, si está libre, transmito, si no hubo colisión, fin de la transmisión. Si hubo ACK asumo que llegó bien, si no recibo confirmación, asumo que no llegó, entonces espero un tiempo aleatorio para volver a intentarlo. Esa es la diferencia con cableadas, existe la confirmación.

- DCF con RTS/CTS - Quiero transmitir:

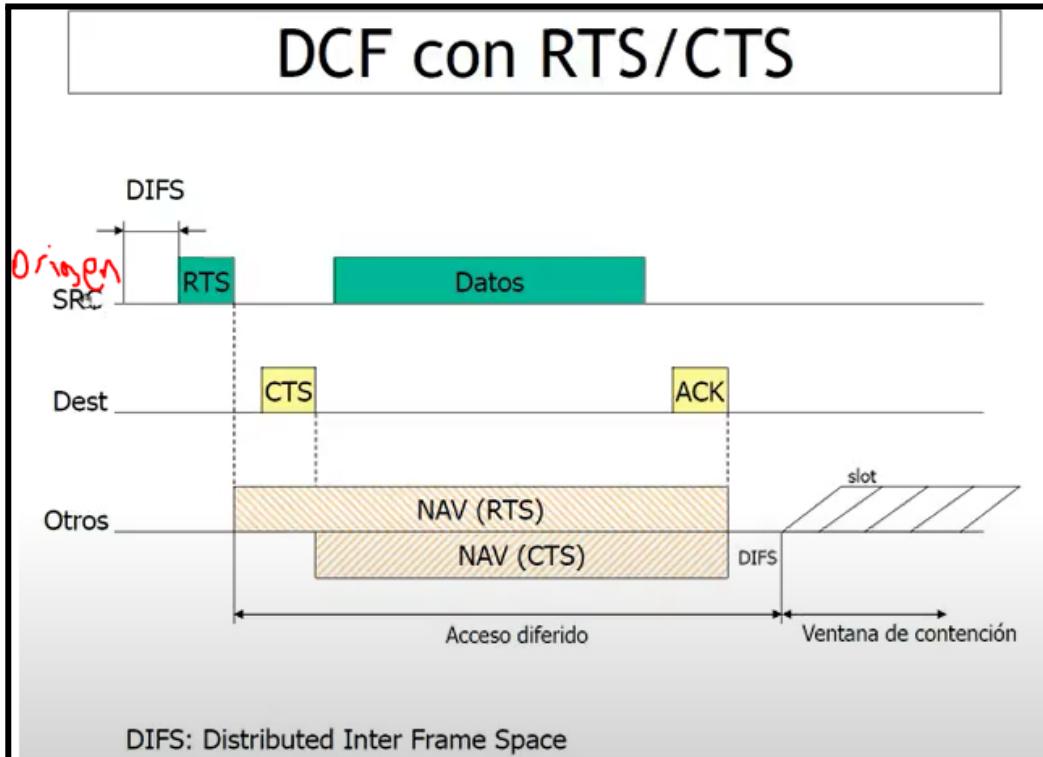
El origen escucha el medio durante un intervalo de tiempo (DIFS, espacio entre tramas distribuido), si está libre durante ese tiempo envía RTS (ready to send), que dentro del RTS se indica el tiempo que necesita la estación para transmitir, al access point ya que toda la comunicación pasa por ahí. El destino recibe esa trama y devuelve un CTS (clear to send, con el tiempo que le queda al emisor para transmitir), el origen envía datos y el destino retorna ACK.

Las otras estaciones escuchan el RTS y el CTS colocando NAV en ocupado según el tiempo que solicitó el emisor.

Nodo oculto: Un nodo de la red que tiene algo lo suficientemente sólido para que otros nodos no puedan escucharlo.

Se pasa al DCF normal ya que los RTS y CTS ocupan mucho tiempo/recursos innecesarios cuando el caso del “nodo oculto” (reserva) no ocurre muy a menudo porque las redes LAN son reducidas. Aunque existe un umbral (mensajes muy largos) donde conviene usarlo.

Vista desde el origen:



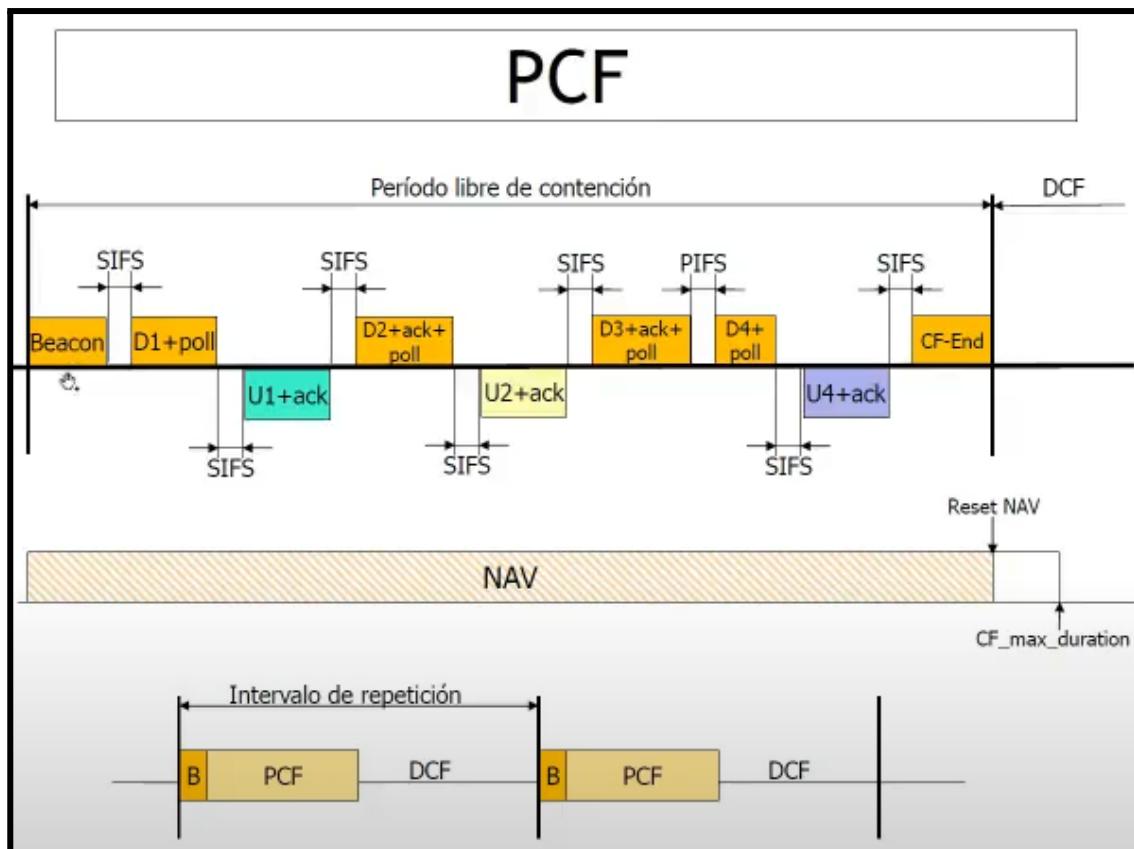
Access point: En wireless es desde y hacia el AP, 2 dispositivos hablan a través de él.

- **PCF - Función de control puntual:**

Se implementó alternadamente para darle certeza, que no sea 100% aleatorio. Para acotar la incertidumbre del cliente para que pueda transmitir.

Se fragmenta el intervalo total de tiempo entre DCF y PCF donde en PCF el access point toma el control total de la celda de tiempo y le asigna a un equipo el turno de transmisión, durante este período los equipos no escuchan para transmitir si no que el AP los designa, cuando se conecta un dispositivo avisa si quiere participar en la lista de dispositivos para transmitir en PCF o no. Luego se vuelve a DCF donde nuevamente compiten por la oportunidad de transmitir.

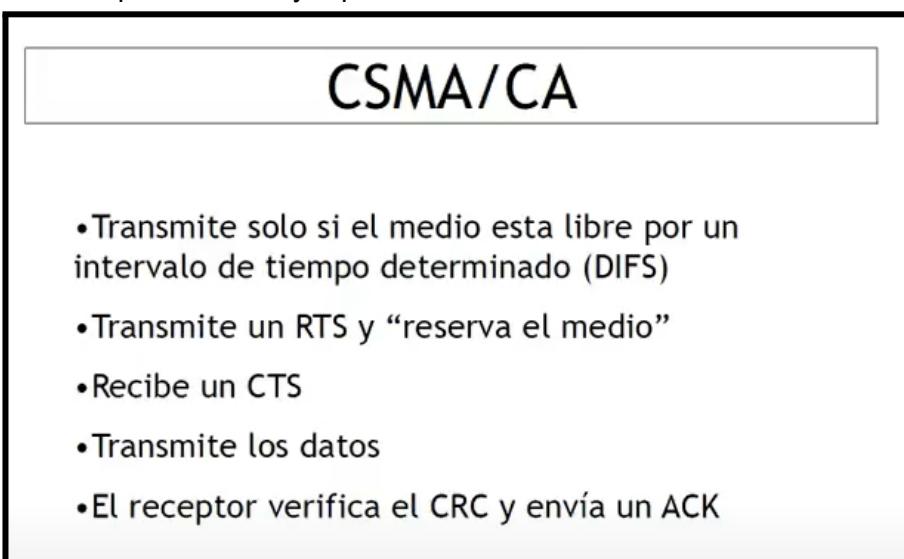
Durante PCF se va preguntando a los de la lista si tienen algo para transmitir, mientras a la par les envía datos en la trama, y sondea al siguiente.



- **Beacon:** trama administrativa que emite el access point, 1 cada 100 ms, 10 por segundo. Anuncia el comienzo de intervalo PCF
- **CF-End:** indica el fin del periodo PCF.

CSMA/CA:

Se suele usar MÁS que cableado ya que el link es más inestable.



Se debe usar cuando:

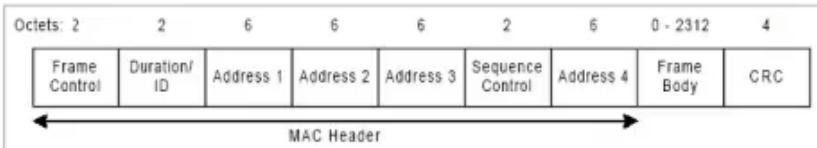
CSMA/CA

El algoritmo exponencial binario debe ejecutarse en cada uno de los siguientes casos:

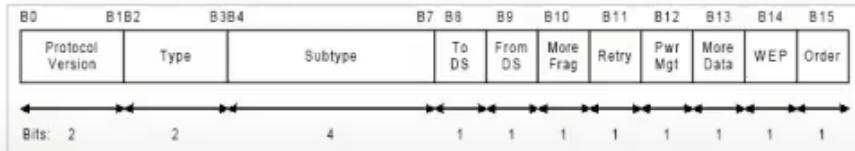
- Cuando escucha el medio y este está ocupado
- Despues de una retransmisión
- Despues de una transmisión exitosa

Koval: SOLO HACE
FALTA APRENDER
TRAMA
ETHERNET,
campos origen,
destino, type, datos,
el resto no. ESTA
SIGUIENTE
TRAMA no hace
falta aprenderla.

Trama



El campo Frame Control :



Frame Control

Las tramas pueden ser :

- Management : Petición/confirmación de asociación, autenticación, Beacon.
- Control : RTS, CTS, ACK
- Datos

ToDS: vale 1 cuando la trama se envía al AP con destino a una estación en DS.

FromDS: vale 1 cuando la trama viene del DS.

MF: indica que hay más fragmentos pertenecientes a la misma trama

Retry: Indica que esta trama ya ha sido transmitida. Sirve para descartar duplicados en caso que se pierda el ACK.

Power Mgmt: indica en que modo estará la estación luego de transmitir esta trama.

More Data: idem, el AP indica a la estación que tiene mas fragmentos para ella.

WEP: indica que el campo de datos está encriptado.

Power mgmt: indica si entra en modo ahorro de energía post transmitir trama la estación. Si es así, el AP almacena tramas que vayan hacia ese destino. En el beacon le indica a los destinos si tienen información pendiente. Si las hay, envía trama al AP (poll) solicitando lo pendiente.

More Data: No te duermas que tengo más para entregarte.

Duration/ID: intervalo de tiempo utilizado para calcular el NAV time.

Address Fields: ver cuadro.

Sequence Ctrl: identifica el fragmento.

CRC: 32-bit

| To DS | From DS | Address 1 | Address 2 | Address 3 | Address 4 |
|-------|---------|-----------|-----------|-----------|-----------|
| 0 | 0 | DA | SA | SSID | N/A |
| 0 | 1 | DA | BSSID | SA | N/A |
| 1 | 0 | BSSID | SA | DA | N/A |
| 1 | 1 | RA | TA | DA | SA |

- Address 1: destino (MAC)
- Address 2: remitente (MAC)
- BSSID: MAC del Access Point.

Diferencias con Ethernet:

- Fragmentación y reensamblado que no hay en ethernet:

Debido al alto BER en capa 1 por las distancias y atenuación (rayitas de señal en tu celular por ejemplo) que no ocurren en cableado estructurado pero acá sí por las antenas, conviene dividir los 1500 bytes de ethernet en fragmentos menores, así en caso de error solo transmitís el fragmento con error.

Wireless LAN - conectarse a un AP:

Para incorporarse a una celda (AP), la estación debe:

1. Sincronizarse: por medio de beacons enviados por AP
 - a. Modo pasivo: Escucha, el disp. escanea todos los canales de las bandas 2.4 y 5.0 buscando el beacon del access point (SSID).
 - b. Modo activo: Envía un prompt (trama) donde la estación pregunta al AP si está disponible.

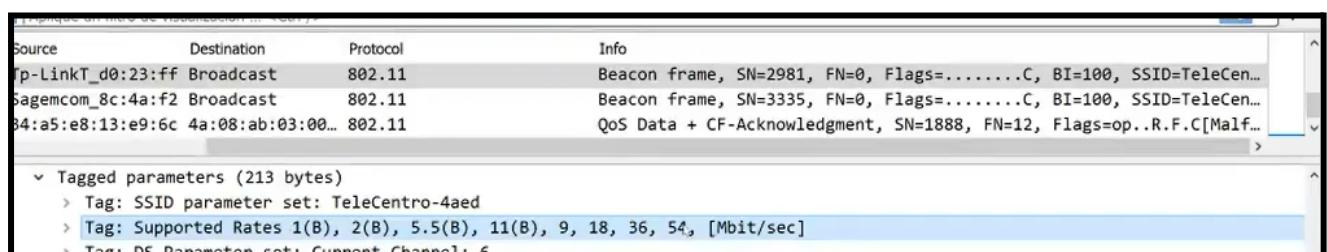
2. Autenticarse: intercambio de clave entre AP/terminal.
3. Asociarse: se vincula el equipo al AP permitiendo la transmisión.

Protección:

El estándar define WEP como protección, hoy día hay muchas mejores.

Diferentes normas para Wireless LAN Standard:

| 802.11 Wireless LAN Standards | | | | | |
|-------------------------------|---------|------|------------|-----------|---------------------------------|
| | 802.11a | B | G | N | AC |
| Velocidad (Mbps) | 54 Mbps | 11 | 54 | Hasta 600 | 433 /867 /1.69 Max 3.39 Gbps |
| Frecuencia de operación (GHz) | 5 | 2.4 | 2.4 | 5 & 2.4 | 5 & 2.4 |
| Modulación | QFDM | DSSS | OFDM, DSSS | OFDM | OFDM |



Si te vas alejando y pierdes señal vas a necesitas usar una técnica de modulación peor que sea menos sensible al ruido pero que transita menor información. Vas cambiando la técnica de modulación según la distancia.

Si quiero poner distintos AP en una misma banda, debo usar distintos canales (1,6,11) porque si no se pisarían sobre el mismo ancho de banda.

03/09

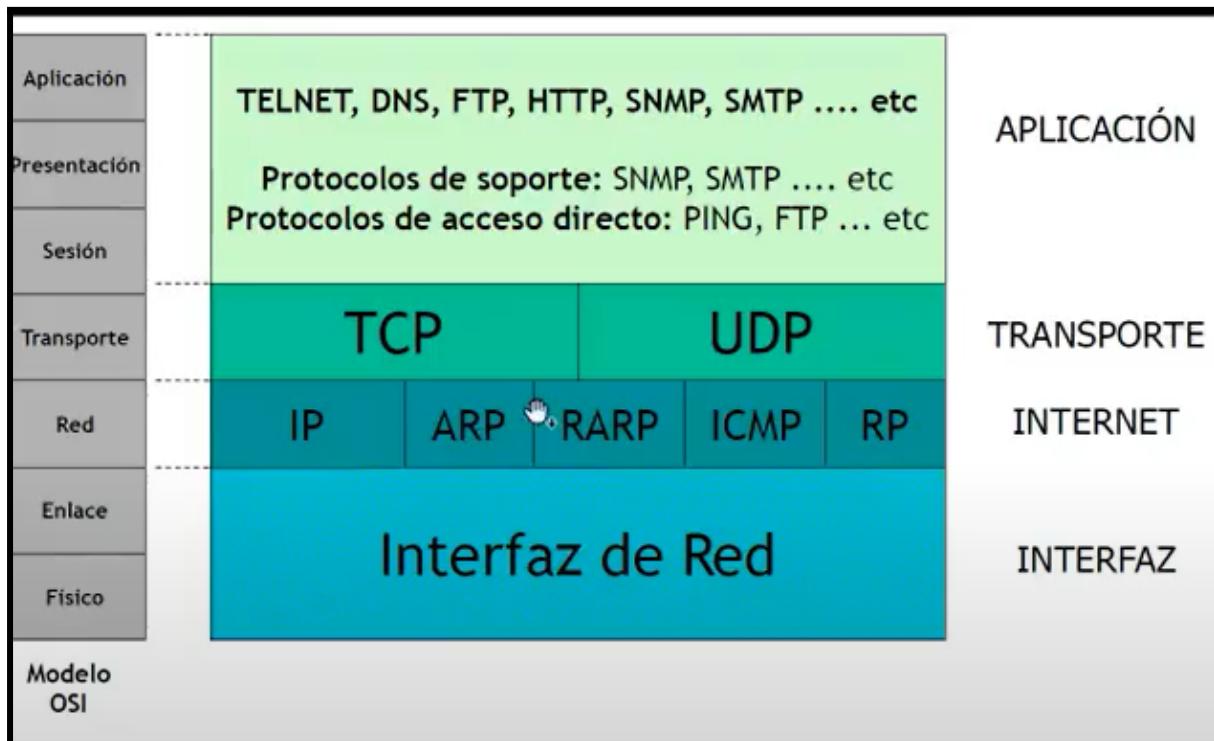
Arquitectura TCP/IP

El modelo comienza de capa 3 para arriba. Lo que hay abajo la arquitectura lo llama "interfaz de red" y no está especificado, es lo que cada uno quiera.

Todo tiene 32 bits de ancho. Contando palabras de 32 bits.

Suite TCP/IP:

Conjunto de protocolos compuestos esencialmente por TCP e IP pero hay muchos más



802.3 y 802.11 → Protocolo IP puede correr sobre ellos.

TCP: Cumple funciones de transporte y algunas de la capa de sesión.

Hay un único protocolo de aplicación que son las 3 capas del modelo OSI (aplicación, presentación y sesión).

Filosofía de internet:

- Servicios de aplicación que corren o no sobre TCP
- Servicio de transporte confiable y todo corre sobre TCP
- Servicio de entrega de paquetes connectionless (no orientado a la conexión).

Protocolos orientados a la conexión

cuando se identifican 3 etapas:

1. Establecimiento de la conexión
2. Intercambio de información
3. Liberación o desconexión

Ej: llamada telefónica.

Protocolos orientados a la NO conexión

Una única etapa. Intercambio de la información

Ej: Ethernet, 802.3, Wireless LAN.

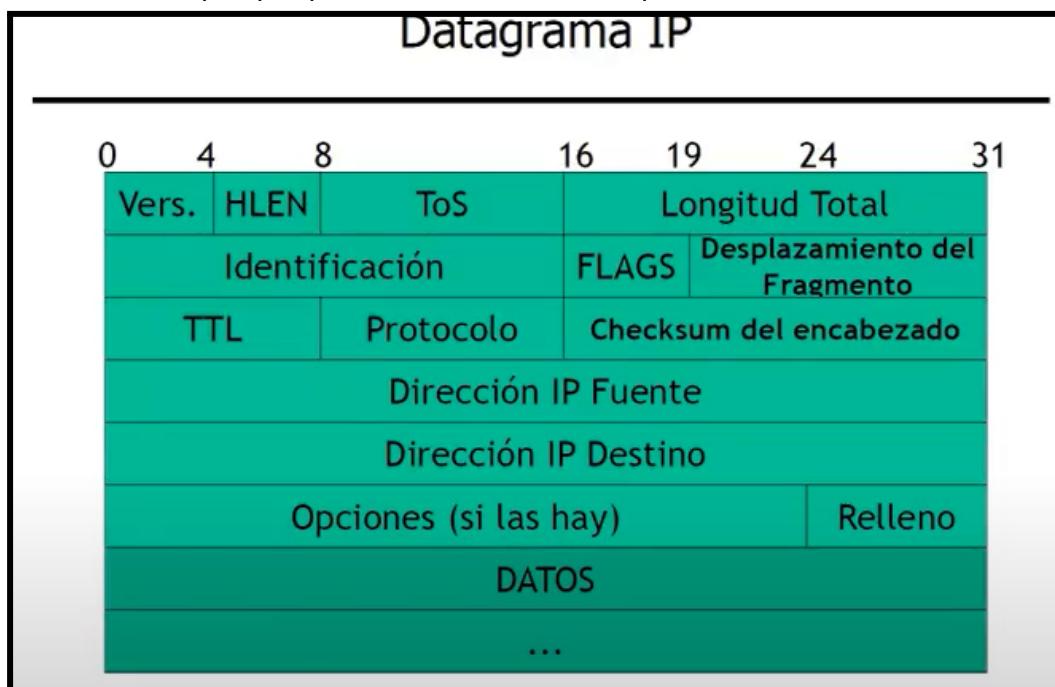
IP = Servicio “Connectionless”

- No confiable: Los paquetes pueden ser

- perdidos: La capa de red no ofrece confirmación y no avisa que se perdió. La capa de transporte tiene esa responsabilidad
 - duplicados: La capa de transporte tiene la responsabilidad de chequear y descartar.
 - desordenados
 - demorados.
- Connectionless: Paquetes tratados independientemente. No existe un estado en los routers acerca de cómo fueron tratados los paquetes anteriores, ni qué contenían. Aunque se separen en varios paquetes/datagramas en orden una trama, a cada paquete/datagrama el conmutador lo trata como independiente y le calcula la mejor ruta. Es una ventaja porque a pesar de los errores de la red todos los paquetes siguen.
 - Entrega Best-effort: El software realiza un serio intento por entregar el paquete (sin garantía)

Conformación de cabecera:

Tiene más campos porque ofrece más servicios que ethernet.



1. Versión: Ej: IPv4 : 0100.
2. HLEN: Header length. Indica cuantas palabras de 32 bits ocupa la cabecera. Pero hay una cabecera mínima de 5 palabras, 20 bytes (hasta dirección IP destino en la foto).
Valor máximo 15, 60 bytes.
3. ToS: Type of service. De los 8 solo usan 4, solo 1 de estos bits puede estar encendido, a grandes rasgos establecer que tráfico es más prioritario que otro. 1)Minimize delay, 2) Maximize throughput, 3)Maximize reliability, 4) Minimize cost. Tenía sentido con redes chicas al inicio, fuera de nuestro entorno de red pierden sentido porque depende del proveedor de servicios

también.

| Valores recomendados de ToS (RFC1349) | | | | |
|---------------------------------------|----------------|---------------------|----------------------|---------------|
| APLICACION | Minimize Delay | Maximize throughput | Maximize Reliability | Minimize Cost |
| Telnet / Rlogin | 1 | 0 | 0 | 0 |
| FTP - Control | 1 | 0 | 0 | 0 |
| DNS Query (UDP) | 1 | 0 | 0 | 0 |
| FTP - Data | 0 | 1 | 0 | 0 |
| ICMP | 0 | 0 | 0 | 0 |

4. Longitud total: Longitud máxima de IPV4 64K
5. Identificación: Cada datagrama IP sale de un host con un campo de identificación aleatorio distinto. 2^{16} posibilidades.
6. FLAGS: El primer bit no está declarado, el segundo es no fragmentar, el tercero más fragmentos.
7. Desplazamiento del fragmento
8. TTL: Time to live, el tiempo de vida del datagrama. Como el servicio es no orientado a la conexión, el mensaje se maneja por la red router por router, si no hubiera destino el paquete entra en loop de ruteo. Para que no quede indefinidamente por la red. En segundos, máximo 255. Cuando llega a cero ese paquete no puede seguir reenviándose. Decrementado por cada router en 1 unidad. Al fin y al cabo indica cuántos saltos y no segundos.
9. Protocolo: Nos indica qué hay en el campo de datos. Igual al campo ethertype. ICMP = 1 ,TCP = 6, UDP = 17.
10. Checksum del encabezado: Suma de comprobación solo para cabecera..
11. Dirección IP fuente.
12. Dirección Destino
13. Opciones: si las hubiera + relleno
14. Datos: Longitud variable

Al analizar o parsear una secuencia de IPV4 sabemos que comienza con 4 y sigue usualmente un 5.

Fragmentación y reensamblado:

Si se tiene un mensaje mayor que el máximo, no se puede encapsular, ejemplo: más de 1500 bytes en una trama ethernet. Para poder hacer eso sin exponer los datos a la capa superior, IP ofrece este servicio, le oculta a la capa de transporte los detalles de la interfaz de red, encargándose de fragmentar el mensaje si fuese necesario.

IP toma el mensaje de la capa de transporte lo encapsula en un datagrama IP y luego fragmenta ese datagrama para asegurarse que entre dentro de la trama.

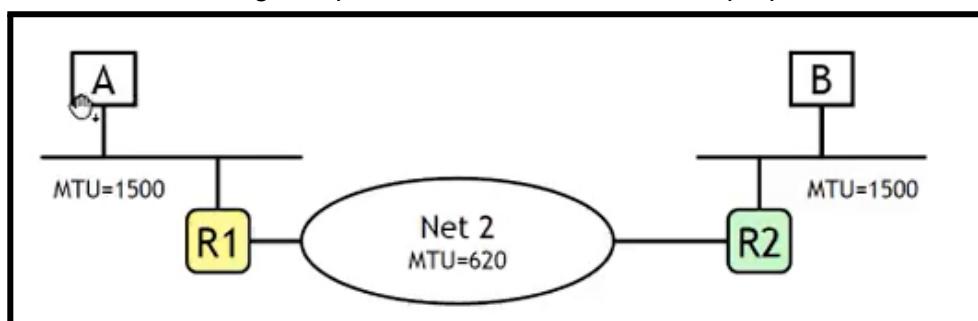
- **MTU: Maximum transfer Unit:**

Cada tecnología de conmutación de paquetes, fija un límite máximo para la cantidad de datos que pueden transmitirse en una única trama.

| <i>Red</i> | <i>MTU (Bytes)</i> |
|--------------------|--------------------|
| Token Ring 16 Mbps | 17914 |
| IEEE 802.3 | 1500 |
| X.25 | 576 |

Ejemplo:

Los routers están conectados con una tecnología de capa 2 con un MTU < que el MTU de ethernet. El router1 recibe el mensaje de 1500 bytes sin problema, lee la cabecera y lo debe conmutar por la red de menor MTU, para eso lo fragmenta, los recibe el router2, y se los envía al destino fragmentados, este destino debe reensamblarlo. Router2 no reensambla ya que generaría una sobrecarga de procesamiento, sólo conmuta paquetes.



-IP oculta los detalles de la tecnología subyacente. Divide los datagramas en fragmentos.

-Los fragmentos deben ser reensamblados en Destino.

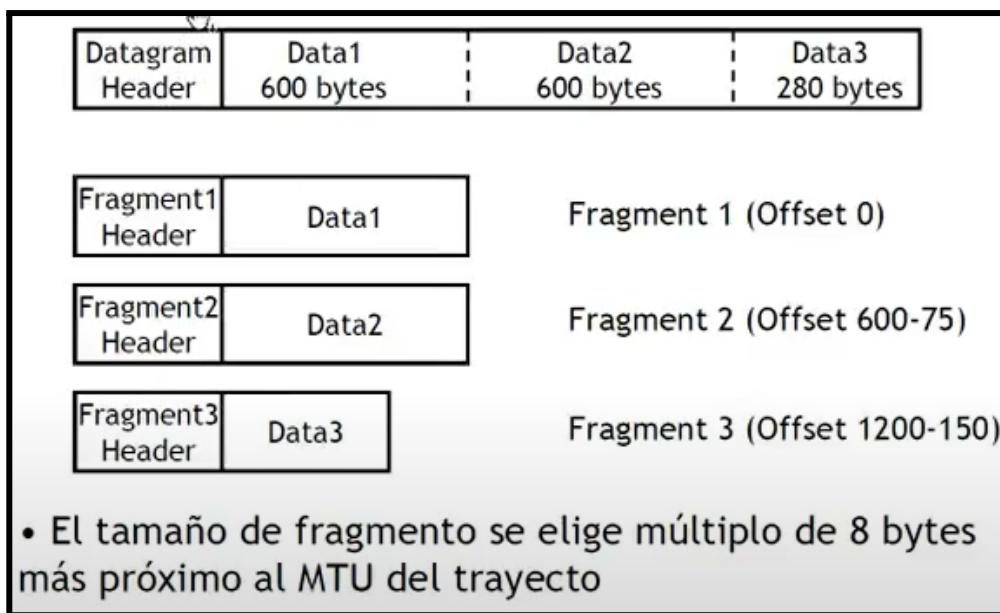
Datos del header modificados y estables ante fragmentación:

Cada fragmento tiene una copia del header con algunos datos modificados para indicar que es un fragmento.

- offset (desplazamiento del fragmento respecto al datagrama original) expresado en bloques de 8 bytes porque es pequeño y no acepta nros muy grandes. Un valor * 8 = desplazamiento real.
- Flag “más fragmentos” = 0 si es el último o el único.
- Flag “no fragmentar” impide fragmentación, envía mensaje de error al origen notificando que descarta y notifica el MTU del salto que no pudo realizar así el origen sabe la manera correctiva para que se realice la comunicación.

Campos de la cabecera que permanecen invariables:

- Mismo origen y destino
- identificador: Todos los fragmentos tienen el mismo.



Desventajas de fragmentación:

Muchas apps prefieren poner bit de no fragmentar encendido. Era necesario al comienzo del uso del protocolo para adaptarlo con otros. Ahora otros protocolos tienen la posibilidad de adaptarse al tamaño del datagrama.

1. Duplica la probabilidad de pérdida de un datagrama:
Da error en la transmisión, no llega a destino aunque llega una parte y ocupé recursos de red en transmitir medio mensaje que no sirve para nada. Ya que el origen no es consciente de la fragmentación y no va a retransmitir.
2. Genera mayor carga de procesamiento en los routers:
Tiene que partir el mensaje, generar nuevas cabeceras, calcular nuevos checksums, reenviar mensajes, etc.
3. Puede producir excesivas retransmisiones si hay pérdida de paquetes:
Si tengo una tasa de pérdida de paquetes elevada y además hay fragmentación el resultado es caótico.

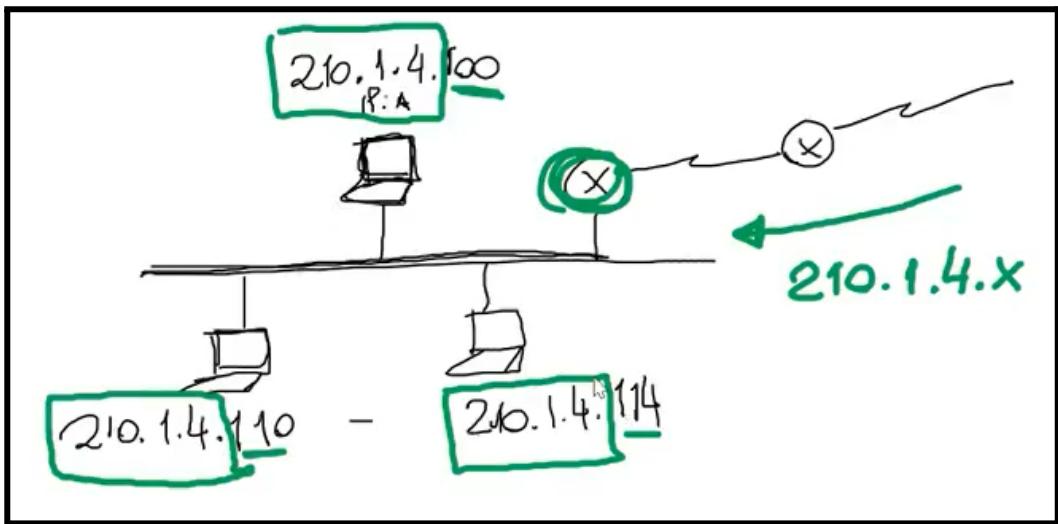
Direccionamiento:

- Dirección única en Internet
- 32 bits de longitud
- 4 bytes separados por “.” con notación decimal.
- La máscara identifica RED/HOST: Una parte identifica a la red y otra al host (disp) puntual. Se configura.

-El origen y destino tienen una dirección IP única

-El router cumple la función de encaminamiento: Tienen info almacenada que permite encontrar el camino al destino.

Necesito el protocolo IP (de capa 3, protocolo de red) para hablar con alguien fuera de mi red



Máscara de subred:

- Es un parámetro de configuración local del dispositivo con función de ruteo. Me permite saber quién soy, quiénes son mis vecinos (puedo usar tramas ethernet porque es de mi red) o sino usar el router para sacarme de la red y buscar la dirección destino en otra red.
- Es obligatoria junto a la asignación de IP porque permite determinar que parte de la IP representa la red y qué parte al host.
- No viaja en el datagrama IP

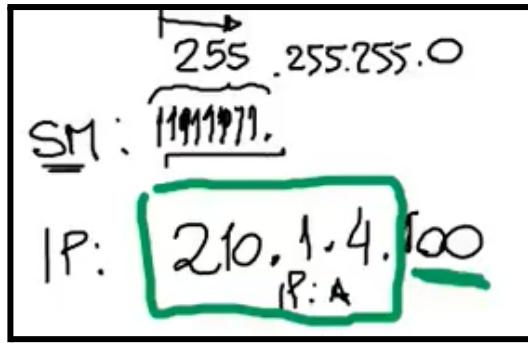
- Un 1 en la SM indican que el bit correspondiente en la dirección IP forma parte del identificador de red
- Un 0 en la SM indica que el bit correspondiente en la dirección IP identifica al host.

Los 1 deben ser consecutivos comenzando por la izquierda. (no puede haber 101010).

Ejemplo:

Los primeros 24 bits identifican a la red entonces corresponde 1 en la SM

$$\begin{aligned} \text{IP} &= 210.1.4.100 \rightarrow \text{máscara de red} \\ \text{SM} &= 11111111 . 11111111 . 11111111 . 00000000 \\ &= 255.255.255.0 \end{aligned}$$

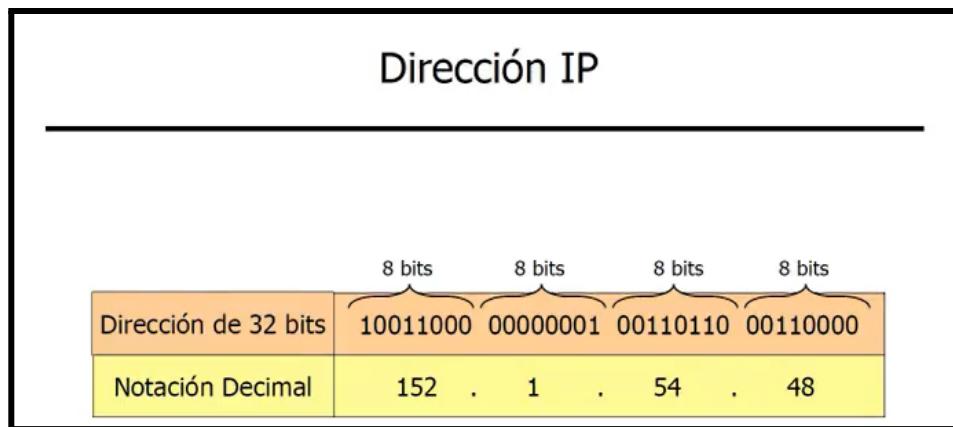


- Prefijo identificador de red
- Sufijo identificador de host: Longitud de prefijo.

Aclaraciones:

-Para el host de 8 bits solo puedo tener 254 combinaciones de dispositivos diferentes (el 00000000 identifica a la red y el 11111111 identifica al broadcast por lo tanto no pueden usarse)

-El datagrama solo tiene la dirección IP origen y IP destino. La máscara no existe, es local en la configuración del dispositivo. Es el dispositivo el que aplica la máscara a una dirección IP para comparar si la dirección IP destino comparte prefijo con la dirección IP propia del dispositivo y se trata de un vecino de red o está por fuera.



IANA:

-Organismo de asignación de números en internet.

2^{32} combinaciones de direcciones.

-Toma la tabla y reparte las direcciones IP para que no se repita y sea única. Además de ser bien administrado.

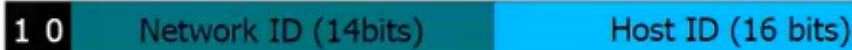
-Particiona el espacio de direcciones en bloques, formando redes muy grandes, redes medianas y pequeñas para asignar de a bloques.

Clases de direcciones

CLASE A



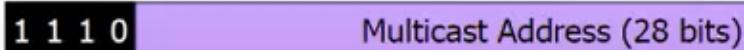
CLASE B



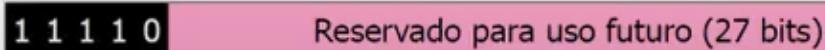
CLASE C



CLASE D



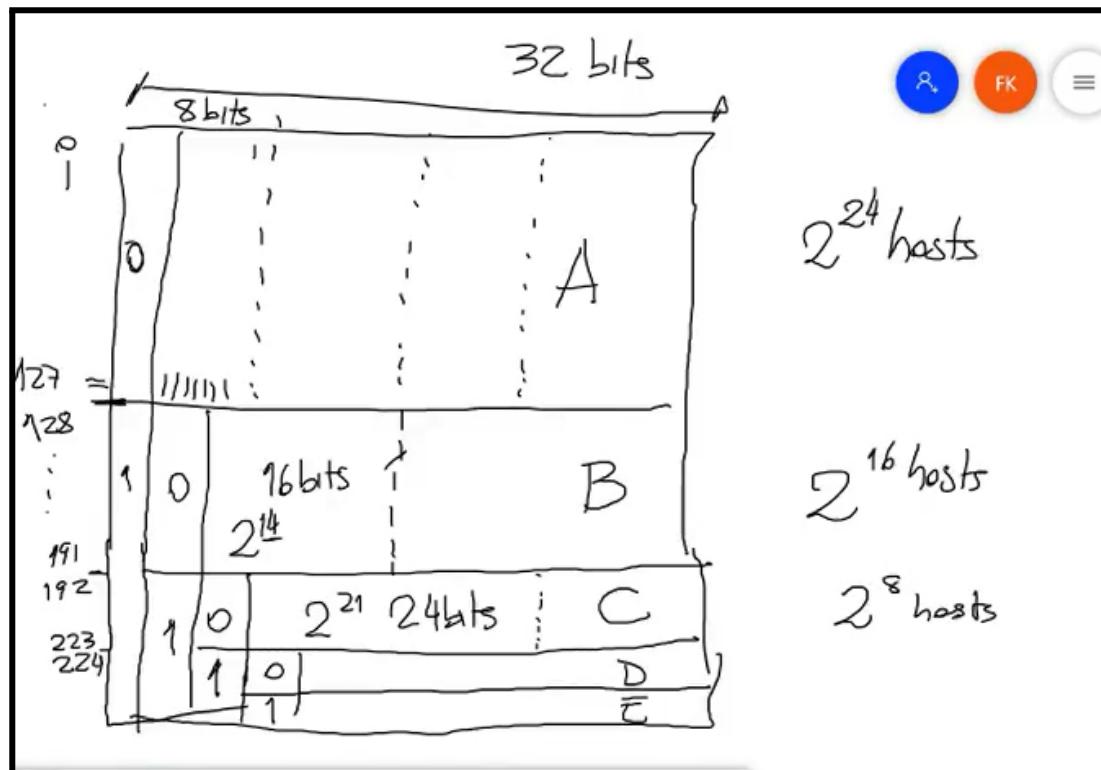
CLASE E



| Clase | Cantidad de redes | Red más baja | Red más alta | Cantidad de hosts por red |
|-------|-------------------|--------------|---------------|---------------------------|
| A | $2^7(128)$ | 1.0.0.0 | 126.0.0.0 | $2^{24}(16M)$ |
| B | $2^{14}(16K)$ | 128.1.0.0 | 191.255.0.0 | $2^{16}(64K)$ |
| C | $2^{21}(2M)$ | 192.0.1.0 | 223.255.255.0 | $2^8(256)$ |

- Las redes clase A:
 - Tienen prefijo de red de 8bits (primer octeto), los 3 octetos restantes son identificador de host.
 - Tiene 2^{24} hosts
 - Comienzan en 0 y terminan en 127
 - Solo 127 combinaciones (Diferentes redes)
- Las redes clase B:
 - Tienen prefijo de 16 bits
 - Tienen 2^{16} hosts.
 - Comienzan en 128 hasta 191

- 2^{14} combinaciones
- Las redes clase C:
 - Tienen prefijo de 24 bits.
 - Tienen 2^8 hosts
 - Comienzan en 192 hasta 223.
 - Solo hay 2^{21} combinaciones.
- Las redes clase D:
 - Comienzan en 224



La máscara natural para cada clase es 8 bits para clase a, 16 bits para clase b y 24 bits para clase c.

Direccionamiento privado:

Se reserva un bloque clase a, clase b y c para uso privado en una red hogareña y se pueden comunicar todos los dispositivos en la red LAN. Las puede usar quien quiera sin pedirle permiso a la IANA. Están repetidas y no se pueden usar para navegar en internet. Son la clase 10 de A, los bloques 172.16 a 31 clase b, y el bloque 192.168.0.0 clase c.

- Direccionamiento Privado (definido en RFC1918)
 - 10.0.0.0 - 10.255.255.255 (10/8 prefix)
 - 172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
 - 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)
- 100.64.0.0/10 – Shared Address Space (RFC 6598)
- VLSM & CIDR
 - Variable-Length subnet Mask (192.168.2.64/28)
 - Classless Interdomain Routing (192.168.0.0/16)

La primera dirección IP del rango se suele asignar al router.

El proveedor de internet le asigna una IP pública a mi router hogareño. Mi pc, mi celular, televisor, etc. Todos tenemos nuestro mismo IP origen (se llama NAT, network address translation, mecanismo que se creó a partir de escasez de IPs públicas) → para ingresar a internet desde una IP privada me debo traducir a una IP pública.

10/09

Clase A:

- Máscara natural: prefijo de 8 bits
- Máscara de subred: 255.0.0.0

Dependiendo del número del primer octeto que ponga me sugiere una máscara. (hasta 128 me sugiere 255.0.0.0 , hasta 192 me sugiere 255.255.0.0, luego me sugiere 255.255.255.0)

Shared Address Space (RFC 6598):

100.64.0.0/10

Rango de direcciones reservado para mecanismos de proveedores de internet, carrier grade NAT, por la escasez de ips V4. Es el mismo mecanismo que el router hogareño usa para repartir direcciones privadas y luego traducir a pública, para el proveedor es a nivel carrier con todos los abonados, el router hogareño en vez de tener una IP pública tiene una dirección de este rango, además del NAT del router, el proveedor hace un segundo NAT. La verdadera dirección pública está en el dispositivo central que hace el segundo NAT.

Classless Interdomain routing:

Se deja sin efecto el esquema de clases con el paso del tiempo, cada red va a tener un prefijo que puede ser variable, no necesariamente un octeto. Hay máscaras que no son sólo de 8,16 o 24 bits sino que pueden ser de 12, 23,25 con asignación más eficiente de IPS.

Le podés dar un bloque sin necesariamente darle toda una clase.

Notación CIR:

PREFIJO / [LONGITUD DEL PREFIJO]

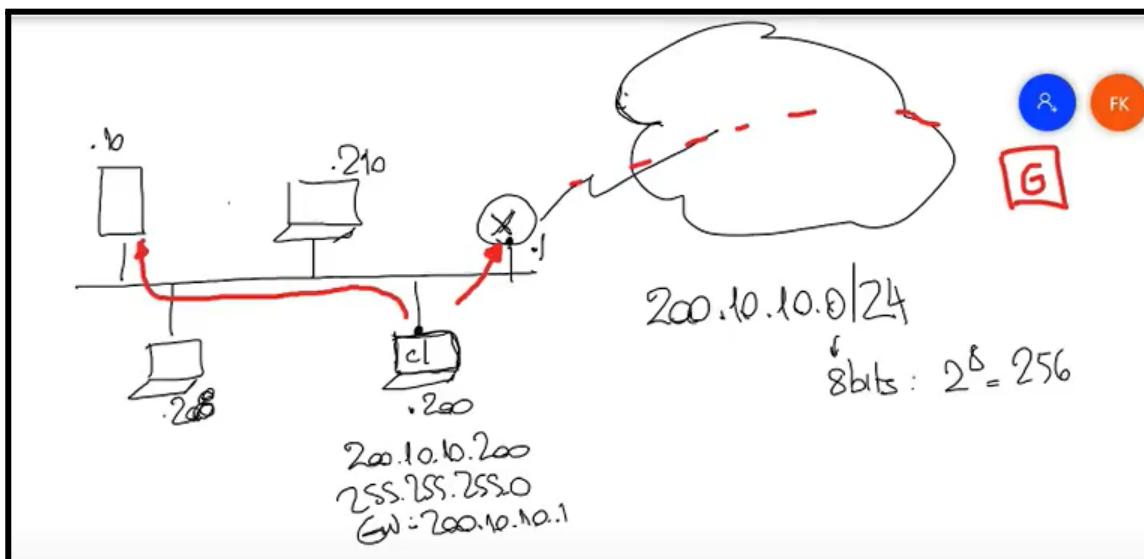
Ej: 192.168.0.0/16 que indica red 192.168 con máscara 255.255.0.0 que con esquema de clases 192 es clase C pero máscara 255.255.0.0 es clase B.

Variable-length subnet Mask:

Máscaras de subred de longitud variable que permiten salirse del esquema rígido de octetos.

Ej: 19.168.2.24/28 → indica que sólo los 4 bits del último octeto son del host. El 64 pasado a decimal no me permite ver en bits, pero si lo paso a binario 0100 0000, me permite 16 combinaciones, aunque son sólo 14.

Escenario al enviar un mensaje a otra estación (dentro o fuera de la red):



Si el destinatario es parte de tu red (misma máscara, mismo prefijo) se manda el mensaje como unicast, toma el datagrama IP, se encapsula en trama ethernet, mac origen y mac destino y lo envía. Para obtener el MAC destino desde la IP lo resuelve el protocolo ARP. En cambio, si el destino está fuera de la red, el host se da cuenta por la máscara y va a la gateway por defecto que se puso manualmente (el router).

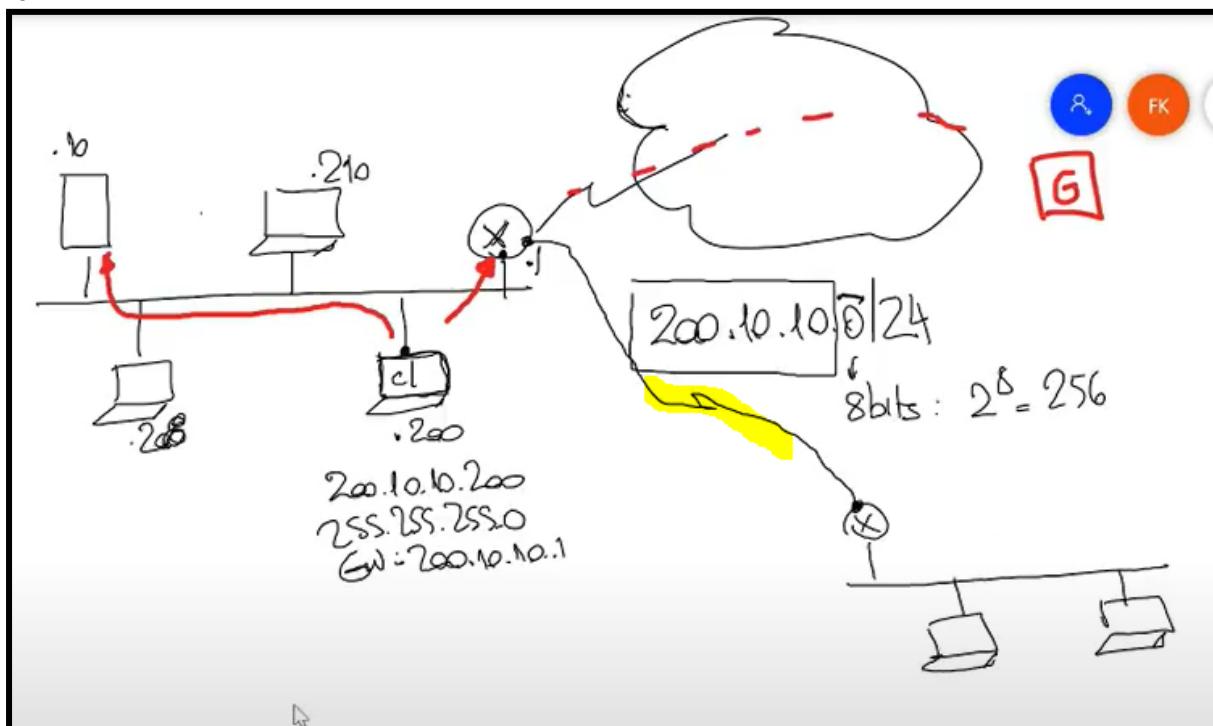
Subredes - Subnetting:

Se abre una nueva sede y se tiene que usar el mismo bloque de la primera red, necesitamos partir la red en subredes. El prefijo no se toca. Veo de modificar los bits de host.

Conceptos importantes:

- Si tengo 2 redes diferentes separadas pero conectadas debo tener redes IP diferentes (ejemplo en foto) por el dominio broadcast así no se ven como vecinos.

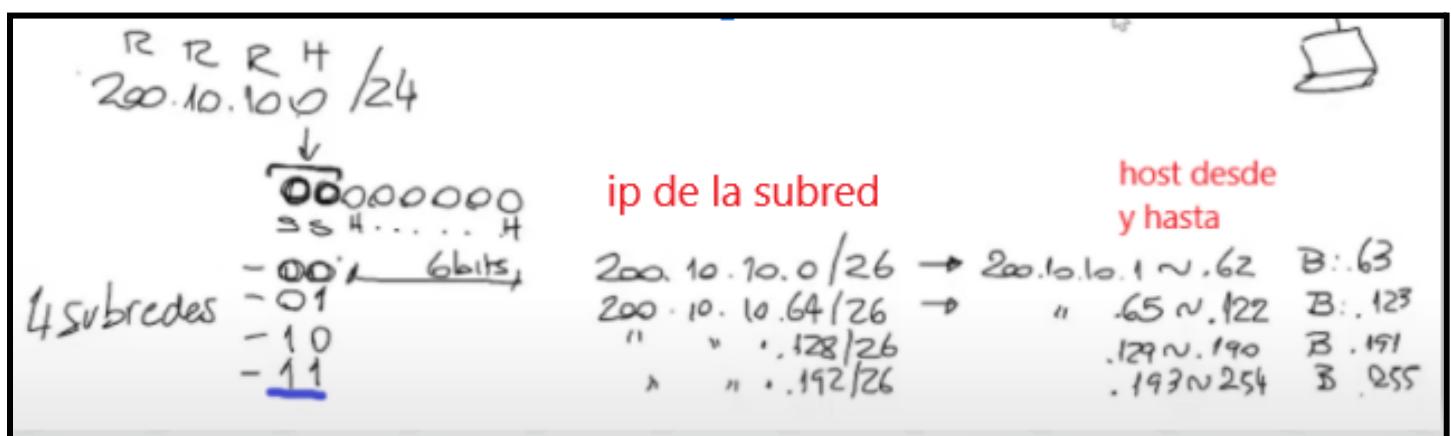
Ejemplo:



Del ejemplo, tomo los 8 bits del host, tomo bits que voy a ressignificar. Cualquiera, los tomo para identificar subredes y los restantes para host.

Parto la IP en = RED + SUBRED + HOST.

RED + SUBRED = son todo el mismo prefijo de red.

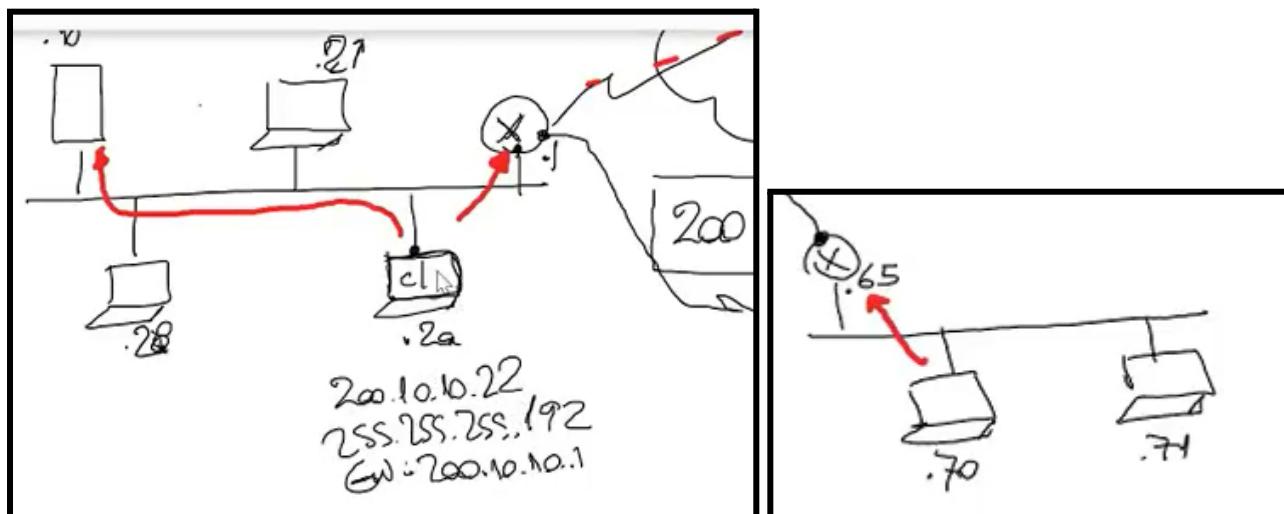


Aclaraciones:

- Subred 1 =
 - 200.10.10.0 **/26** → 2 bits más que /24 de la red central.
 - Host van desde 200.10.10.**1** a 200.10.10.**62** (porque la de todos unos,63, es el broadcast).
- Subred 2 =

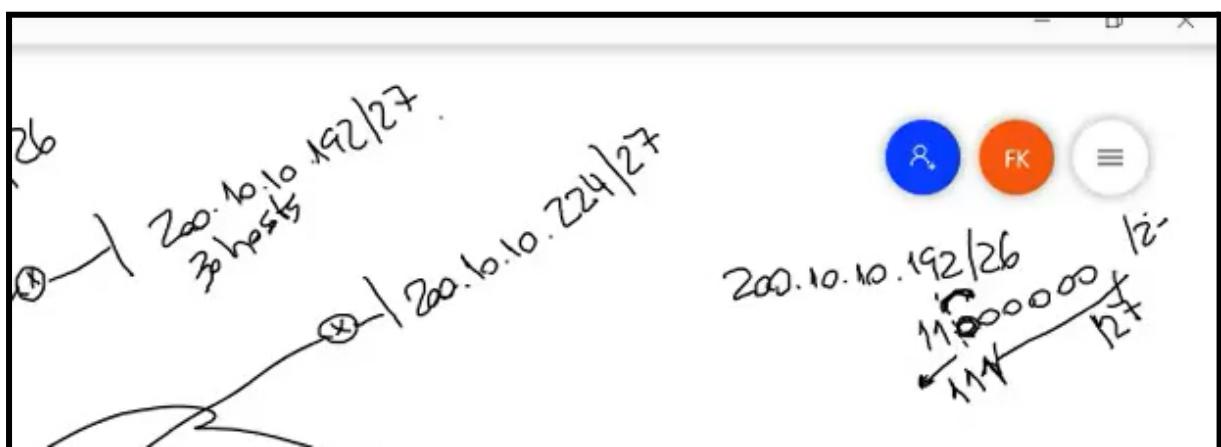
- 200.10.10.64 /26
- Host van desde 200.10.10.**64** a 200.10.10.**122** (porque la de todos unos,123, es el broadcast).
- etc con las 2 redes restantes (imagen).
- Se tomaron 2 bits como mínimo por reglas viejas que el todo 0 y todo 1 de subred no se podían tomar, pero ahora ya no está vigente, y podría tomarse 1 bit y las subredes extremo.

Se deben modificar los host y las máscaras de la primer red del ejemplo porque ya no es una red /24 sino /26 entonces la máscara es 255.255.255.192 (11000000) y te indica que tus vecinos van del 1 al 62, si querés un valor mayor de host tenes que buscarlo en otra subred.



Ahora se agregaron otras 2 sedes más y como tienen pocas estaciones tomo OTRO bit para otra subred /27. Con 110 = 192 y 111 = 224 de prefijos de subred. Y cada una con 30 host combinatorios.

1. 200.10.10.192 / 27
2. 200.10.10.224 / 27



Ejercicios de la guía práctica:

1)

DIRECCIONES IP:

- 5.4.1. Su red utiliza la dirección IP 172.30.0.0/16. Inicialmente existen 25 subredes. Con un mínimo de 1000 hosts por subred. Se proyecta un crecimiento en los próximos años de un total de 55 subredes. ¿Qué máscara de subred se deberá utilizar?

5.4.2. Una red tiene una dirección de 100.10.10.100 y una máscara de 255.255.255.0.

De la IP dada podemos analizar que:

- Es clase B por el 172 y además por que los primeros 2 octetos son de red. La máscara natural (16) con lo planteado.
- Es privada porque la 172.30 está reservada para uso privado.

Tengo un prefijo que no se puede tocar, los primeros 2 bytes pero puedo jugar con los 2 últimos bytes.

Solución:

Puedo tomar el 10.0 /8 o 10.0 /16 me da flexibilidad pero me limita mucho cuando me quiero interconectar con otro en cloud, existen colisiones de IPs.

Tomo 6 bits = $2^6 = 64$ subredes diferentes.

Cada una tiene los últimos 10 bits para hosts $2^{10} = 1024$ host diferentes - 2 (reservados a red y broadcast) como mínimo.

11111111 | 11111111 | 1111 1100 | 0000 0000

Máscara = 255.255.252.0

2)

- 5.4.3. Una red está dividida en 8 subredes de una clase B. ¿Que máscara de subred se deberá utilizar si se pretende tener 2500 hosts por subred?

Analizo:

Por decirme que es clase B asumo que son 16 bits para red.

Para tener 2500 host como mínimo necesito al menos 12 bits. $2^{12} = 4096$ hosts

Solución:

Máscara de clase B normal sin subredes = 255.255.0.0

Parto del mínimo de host y no de las redes.

8 subredes = $2^3 = 3$ bits.

Si bien puede usarse 3 bits, como necesito 12 para host me sobran 4 bits para subred.

$2^4 = 16$ subredes posibles.

máscara con 8 subredes = 11111111 | 11111111 | 1110 0000 | 0000 0000 → ip con /19

máscara con 16 subredes = 11111111 | 11111111 | 1111 0000 | 0000 0000 → ip con /20

Máscara con 8 subredes = 255.255.224.0

Máscara con 16 subredes = 255.255.240.0 → está bien por el profe.

3)

5.4.4. ¿Cuáles de las siguientes subredes no pertenece a la misma red si se ha utilizado la mascara de subred 255.255.224.0?

- A.172.16.66.24
- B.172.16.65.33
- C.172.16.64.42
- D.172.16.63.51

Analizo:

Si comparten prefijo = comparten red.

El límite está en el tercer octeto porque 172.16 está fijo.

Solución:

El prefijo de la subred son los primeros 3 bits del tercer octeto

.63 = 0 0 1 1 1 1 1 → está fuera de la subred.

.64 = 0 1 0 0 0 0 0

.65 = 0 1 0 0 0 0 1

.66 = 0 1 0 0 0 0 1 0

4)

Cuáles de los siguientes son direccionamientos válidos de clase B

- a.10011001.01111000.01101101.11111000
- b. 01011001.11001010.11100001.01100111
- c. 10111001.11001000.00110111.01001100
- d. 11011001.01001010.01101001.00110011
- e. 10011111.01001011.00111111.00101011

Solución:

- a. Comienza con 10, es clase b
- b. Comienza con 0, es clase a
- c. Comienza con 10, es clase b
- d. Comienza con 110 es clase c
- e. Comienza con 10, es clase b

Sino:

Paso a decimal para evaluar los valores con respecto a la tabla de clases:

a = 153. → CLASE B

b = 89. → CLASE A

c = 185. → CLASE B
d = 217. → CLASE C
e = 159. → CLASE B

5)

5.4.6. Se tiene una dirección IP 172.17.111.0 máscara 255.255.254.0, ¿cuántas subredes y cuántos host válidos habrá por subred?

La máscara indica que es de “clase B” y el 254 los bits adicionales para las subredes.
 $254 = 11111110 \rightarrow 2^7 = 128$ subredes distintas.

De la máscara = 1111111 | 1111111 | 1111110 | 00000000
Me sobran 9 ceros $\rightarrow 2^9 = 512$ hosts distintos por cada subred.

Aclaraciones importantes:

- Si me dan una IP y una máscara me dan un bloque de direcciones, una red, un fragmento.
Ej: IP 172.17.111.0 Y 255.255.254.0 → primero fue una clase B ahora es una subred, dejamos de pensar en la clase, sino en bloque de direcciones (/23)
Pero cuando hablamos de redes y subredes volvemos al esquema de clases, donde las redes marcan clases y las subredes cualquier modificación que hice de la máscara original.
- Error de enunciado en 5.4.7 → máscara 255.255.255.254 → no tiene host ya que el 0 es para la red y el 1 para broadcast y como la cantidad de hosts es 2^1 no hay.

ARP - Address resolution protocol

RFC 825

Su funcionalidad es mapear direcciones IP de alto nivel lógicas a direcciones MAC físicas.

Ya que sé la dirección IP destino de un mensaje pero no la MAC destino.

Se encapsula directamente sobre ethernet.

- Las aplicaciones de alto nivel solo trabajan con direcciones IP
 - Ilusión de una única red virtual: Pongo la dirección IP y llego al host de una manera u otra (si es mi vecino con mismo prefijo de red y mensaje directo, o con host fuera de mi red pasando por mi router y luego afuera.)
 - La comunicación es realizada por redes físicas reales:
- Los datagramas IP (capa 3) son encapsulados en tramas MAC (capa 2, con protocolo ethernet o 802.3 con MAC origen y destino) → se necesitan direcciones de hardware MAC.

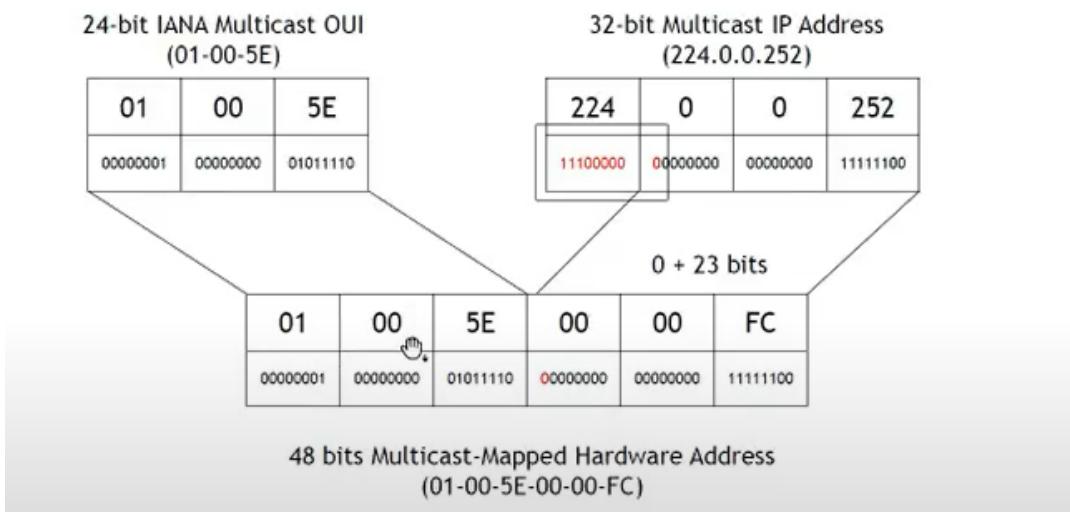
- Si la MAC destino es mi vecino entonces la respuesta del host destino es unicast porque solo le contesto a quien preguntó (solo le basta poner la dirección MAC destino del servidor en la trama ethernet) con ARP.
- La request del MAC es un broadcast, la respuesta es unicast.
- Entrega directa necesito ejecutar ARP

- La dirección MAC de broadcast YA LA CONOZCO y es todos 1, no necesito mandar ARP preguntando. Mapeo un broadcast de capa 3 (192.168.0.255) a un broadcast de capa 2 (ff. ff. ff. ff. ff. ff)

```
Interface: 192.168.0.14 --- 0x17
Internet Address      Physical Address      Type
192.168.0.1            6c-99-61-f7-cc-ef    dynamic
192.168.0.6            d0-ab-d5-3e-df-00    dynamic
192.168.0.105           56-ff-1b-ab-2e-fd    dynamic
192.168.0.255           ff-ff-ff-ff-ff-ff   static
224.0.0.22              01-00-5e-00-00-16    static
224.0.0.251             01-00-5e-00-00-fb    static
239.255.255.250         01-00-5e-7f-ff-fa    static
255.255.255.255         ff-ff-ff-ff-ff-ff   static
```

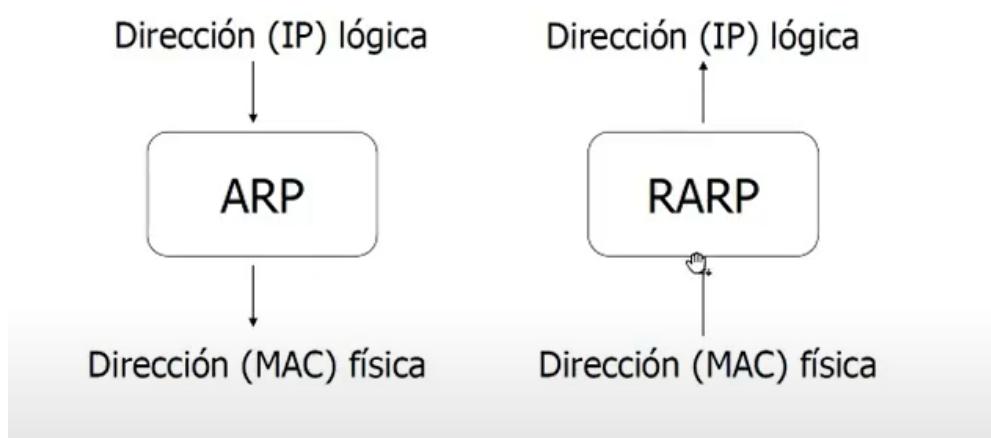
- Multicast va de un origen a un grupo de destinatarios, el grupo multicast tiene una dirección IP de clase D (comienza con 224). Por lo tanto hay una dirección MAC que identifica a un grupo multicast. Tampoco necesito hacer un ARP porque ya lo conozco.
- Ej: En la imagen de arriba la dirección IP 224.0.0.22 es una dirección multicast.
Existe un OUI (01-00-05) reservado para tráfico multicast en IP v4

ARP – Multicast mapping



- En resumen, para broadcast y para multicast no hago ARP, ya conozco las MAC. Para entrega directa sí.
- Existe ARP request y ARP reply
- En la tabla de ruteo gateway = On-Link es ARP.

ARP vs RARP



Reverse ARP:

Evolucionó a Boot B, permite a estaciones sin disco rígido poder obtener el sistema operativo al iniciar, que evolucionó a DHCP (Asignación dinámica de direcciones).

Cómo funciona:

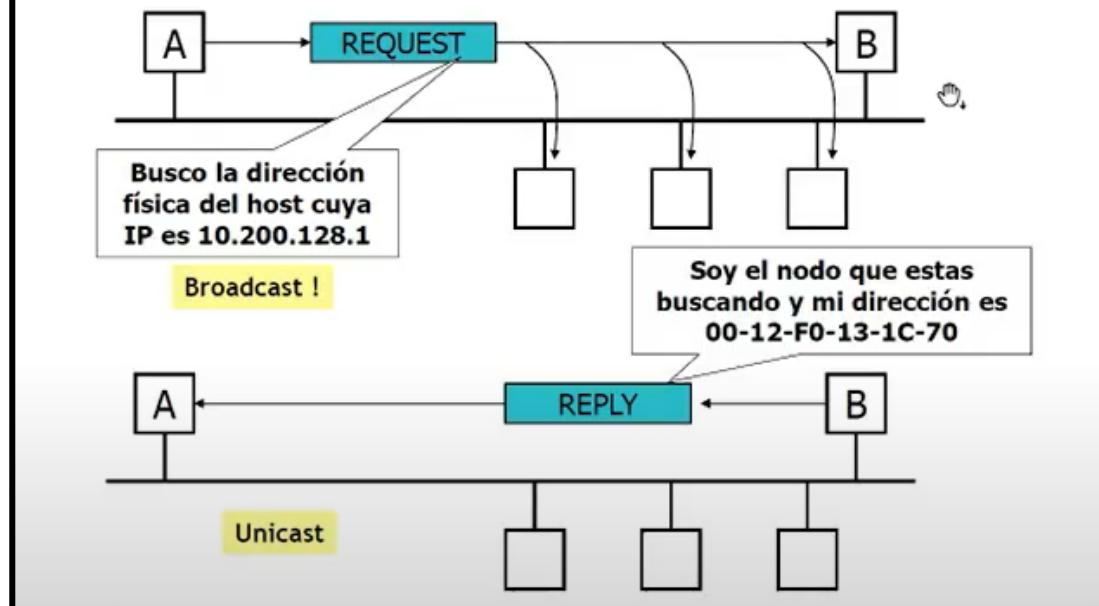
1. El host pregunta a todos quién tiene esa dirección IP mediante una trama ethernet broadcast con dirección MAC destino todos unos.
2. El host destino responde con su dirección MAC.
3. El host que pregunta al obtener la respuesta crea una entrada en la caché de ARP asociando la IP con la MAC, esta entrada es efímera y caduca, periódicamente hay que hacer el ARP para que la asociación sea dinámica y pueda ir cambiando.

En caso de mensajear con un host fuera de la red:

Tengo que hacer el salto del router afuera así que hago ARP como los puntos de arriba

1. Mando ARP con la IP del router.
2. El router me contesta
3. Guardo en la caché la MAC

ARP



Msj- datagrama ARP:

Formato del datagrama

Cant. Octetos

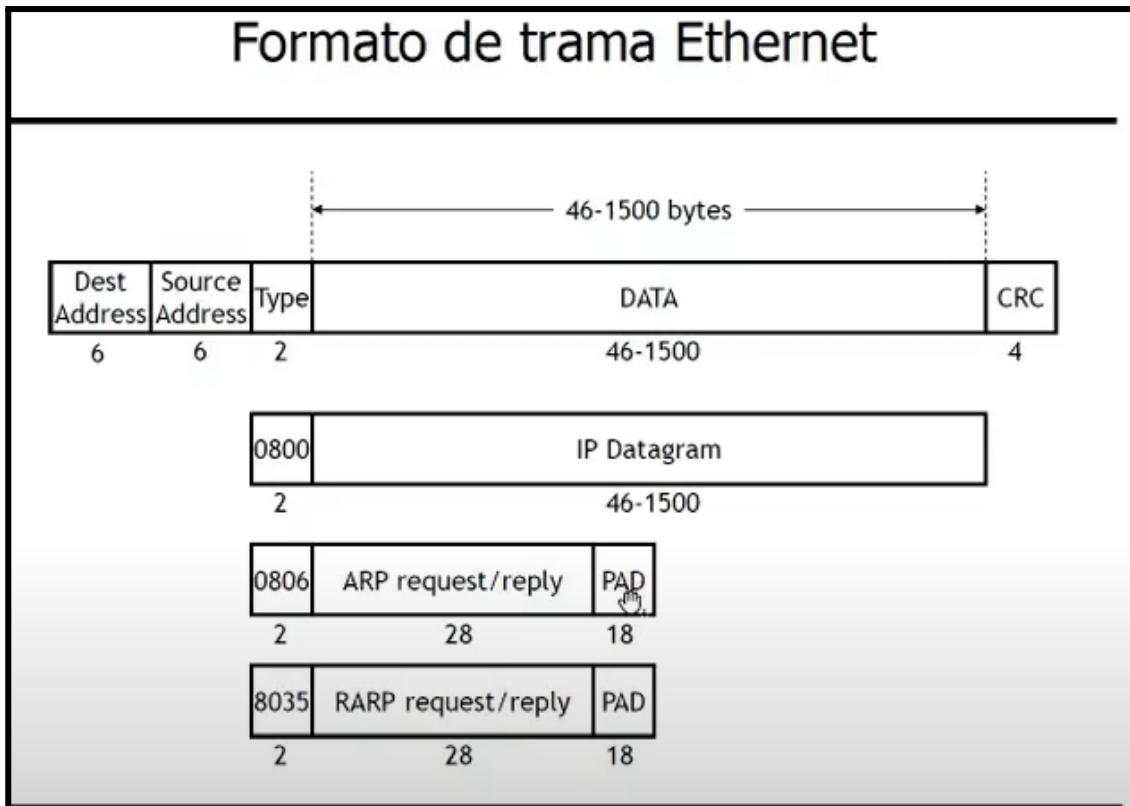
| | | |
|---|----------------------------------|----------------|
| 2 | HARDWARE TYPE | Ethernet=1 |
| 2 | PROTOCOL TYPE | IP=0x800 |
| 1 | LONG. DIRECCION FISICA (en Oct.) | 6 for Ethernet |
| 1 | LONG. DIRECCION LOGICA (en Oct.) | 4 for IP |
| 2 | OPERACION | Ver cuadro |
| 6 | DIRECCION FISICA DEL EMISOR | |
| 4 | DIRECCION LOGICA DEL EMISOR | |
| 6 | DIRECCION FISICA DEL DESTINO | |
| 4 | DIRECCION LOGICA DEL DESTINO | |

1=ARP Request
2=ARP reply
3=RARP request
4=RARP reply

Ethernet II frame type = 0x0806

Aclaraciones:

- Dirección física del destino: MAC address, en la request está en cero porque es lo que busco averiguar
- 28 bytes de largo del datagrama de la imagen, si lo encapsulamos en una trama ethernet no llegamos al mínimo de 64 bytes. Por ende hay que agregarle el PAD de relleno



ARP gratuito:

- Es un ARP que envía un host preguntando por sí mismo (al broadcast)
- Se envía cuando un host quiere asegurarse que una dirección IP está libre, permite detectar conflictos en IP.
- Cuando pongo una dirección IP en una interfaz (Ej. fastethernet 0/2), antes de activarla el host manda un ARP gratuito, si nadie responde está libre, y si alguien responde la IP está en uso.
- Sucede a cada cambio de estado de la interfaz. Cuando apagás y prendés una interfaz hace un ARP gratuito.
- Informan a los switches el MAC del cliente conectado.

| Time | Source | Destination | Protocol | Info |
|-------------|-------------------|-------------------|----------|---|
| 5 0.293818 | HewlettP_be:4a:66 | Broadcast | ARP | ARP Announcement for 10.200.127.34 |
| 6 0.622563 | Cisco_00:17:99 | Broadcast | ARP | Who has 10.200.127.50? Tell 10.200.127.1 |
| 7 1.268651 | HewlettP_be:4a:66 | Broadcast | ARP | ARP Announcement for 10.200.127.34 |
| 9 2.268643 | HewlettP_be:4a:66 | Broadcast | ARP | ARP Announcement for 10.200.127.34 |
| 14 3.414402 | HewlettP_be:4a:66 | Broadcast | ARP | Who has 10.200.127.16? Tell 10.200.127.34 |
| 15 3.414553 | HewlettP_d9:34:50 | HewlettP_be:4a... | ARP | 10.200.127.16 is at 00:50:8b:d9:34:50 |
| 18 3.419527 | HewlettP_be:4a:66 | Broadcast | ARP | Who has 10.200.127.1? Tell 10.200.127.34 |

Address Resolution Protocol (ARP Announcement)

```

Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (1)
[Is gratuitous: True]
[Is announcement: True]
Sender MAC address: HewlettP_be:4a:66 (00:12:79:be:4a:66)
Sender IP address: 10.200.127.34
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
Target IP address: 10.200.127.34

```

17/09

ICMP - Internet Control Message Protocol

(RFC 792).

Como en el protocolo IP los paquetes se retardan, duplican, da inseguridad en el protocolo y poca visibilidad sobre lo que sucede

-El ICMP llena los huecos que deja IP con mecanismos.

- Comunica Errores a nivel de red
- Informa acerca de eventos inesperados
- Informa acerca de la red, en respuesta a consultas.
- Solo informa el error, no especifica qué acción correctiva tomar.

-Se encapsula sobre IP.

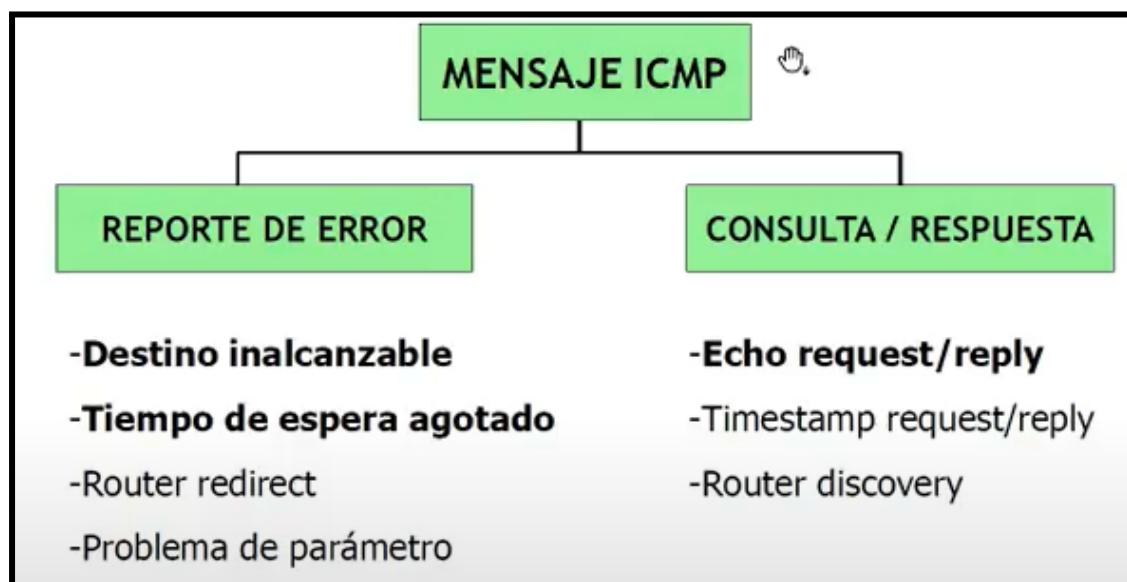
Formato del datagrama ICMP:

Protocolo número 1. 8001



- Checksum: Mismo mecanismo que IP cubre todo el datagrama
- Datos: Generalmente contiene
 - Encabezado IP del datagrama que causó el error.
 - Primeros 8 bytes de datos del datagrama erróneo.
 - Información necesaria para identificar la raíz del error.

Grupos de datagrama ICMP:



Reporte de error:

1) Destino inalcanzable:

Ante la imposibilidad de conmutar/entregar un datagrama el router envía un mensaje ICMP antes de descartarlo.

Motivos:

-Network unreachable: No sabe como llegar a una red. Es decir, al querer conmutar el router no encuentra una ruta y tampoco hay un default route.

-Host unreachable: Sigue cuando el router llegó a la red destino pero no puede entregarlo a la IP destino (nadie contesta).

-Protocol (TCP-UDP) not enabled

-Port not bound to a service: No hay ningún servicio corriendo.

-Fragmentation needed, but DF flag set: Cuando el mensaje que se quiere mandar es más grande que el MTU de la interfaz en el próximo salto.

-Source route failed: En caso que el origen indique cómo tiene que comutar y llegar al próximo salto pero este router no tiene forma de hacerlo, le notifica al origen el error.

2) Tiempo de espera agotado

Motivos:

- El router detecta que el campo TTL debe decrementarse a 0.
- El host destino ha desistido a la espera de un fragmento.

Consulta - respuesta:

3) Echo Request / Reply:

Utilizado para conocer si la interfaz destino es alcanzable y está funcionando.

Si recibo reply significa que existe una ruta destino y tengo visibilidad a nivel 3, no me garantiza que me pueda conectar al HOST pero si la ruta.

-Echo request: Envía un identificador y un número de secuencia para contrastar requests y replies.

-Echo reply: La respuesta no es obligatoria. Debe responder incluyendo los datos recibidos en el request.

Hacer un ping a una IP es un echo request.

| 0 | 8 | 16 | 31 |
|---------------|---------------------|---------------|----|
| Tipo | Código | ICMP Checksum | |
| Identificador | Número de secuencia | | |

PING (Packet InterNet Gopher)

```
C:\>ping www.yahoo.com
```

```
Pinging www.yahoo-ht3.akadns.net [69.147.114.210] with 32 bytes of data:
```

```
Reply from 69.147.114.210: bytes=32 time=183ms TTL=55
```

```
Reply from 69.147.114.210: bytes=32 time=184ms TTL=55
```

```
Reply from 69.147.114.210: bytes=32 time=184ms TTL=55
```

```
Reply from 69.147.114.210: bytes=32 time=218ms TTL=55
```



```
Ping statistics for 69.147.114.210:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 183ms, Maximum = 218ms, Average = 192ms
```

Round trip time: Lo que demoró en ir el request y volver el reply.

Traceroute

```
C:\>tracert www.yahoo.com
```

```
Tracing route to www.yahoo-ht3.akadns.net [69.147.114.210] over a maximum of 30 hops:
```

```
1  1 ms  1 ms  1 ms 200.26.65.5
```

```
2  57 ms  6 ms  4 ms 200.69.225.150
```

```
3  23 ms  4 ms  4 ms customer68-110-190.iplannetworks.net [200.68.110.190]
```

```
4  27 ms  4 ms  3 ms customer191-1.iplannetworks.net [200.61.191.1]
```

```
5  19 ms  5 ms  5 ms 227.54.3.200.telecom.net.ar [200.3.54.227]
```

```
6  *      *      * Request timed out.
```

```
7  *      *      * Request timed out.
```

```
8  225 ms 204 ms 212 ms nota.bas2.dce.yahoo.com [198.32.124.115]
```

```
9  201 ms 214 ms 217 ms ge-2-0-8.p417.pat1.dce.yahoo.com [216.115.97.128]
```

```
10 204 ms 177 ms 190 ms ge-2-1-0-p140.msr1.re1.yahoo.com [216.115.108.17]
```

```
11 188 ms 159 ms 199 ms gi1-23.bas-a2.re3.yahoo.com [66.196.112.55]
```

```
12 171 ms 170 ms 174 ms f1.www.vip.re3.yahoo.com [69.147.114.210]
```



```
Trace complete. Minimum = 183ms, Maximum = 218ms, Average = 192ms
```

Aclaraciones:

- En cada salto de la imagen (del 1 al 2 y así) el origen envió 3 veces el mensaje y 3 veces el router 1 lo rechazó (poner el TTL en 0). En la segunda línea, pasó por el 1 pero el 2 lo rechazó tres veces y así.
- Si el trace se completa, indica visibilidad (que hay ruta) y cuál fue ese camino. También me garantiza que voy a ese dispositivo.
- En la imagen: Los Timed out del 6 y 7 pueden ser por dos razones:
 - Está configurado para que no genere reportes de error por consumo de recursos
 - Porque el dispositivo intermedio no sabe como volver al origen.
- Antes de saber si vas a llegar mandas TTLs cortos. para ir marcando la traza porque queres saber si hay un error de ruteo y donde.
- El reporte de error lo genera el router y es diferente a conmutar paquetes, el reporte tiene menor prioridad.

Variante de Trace route:

El TTL se debe generar para cualquier datagrama IP, en linux se usa datagrama UDP hacia un puerto raro. Pero ambos producen lo mismo, descarte y generación de reporte de error ICMP. Y el procedimiento incrementa en 1 el TTL y avanza un salto, pero finaliza con “puerto no vinculado a ningún servicio” y no con un echo reply y el reporte de error generado indica el fin de la trama.

Ruteo y Routing Protocols:

Ocurre en la capa 3 (porque se cumple el encaminamiento y direccionamiento, posibilidad de direccionar y referencia dispositivos)

Un router para enviar los datagramas al siguiente “Hop” o salto realiza 2 funciones:

1. Determinar el mejor camino a destino
2. Comutar el datagrama: Encapsularlo en el protocolo de capa 2 del siguiente link que lo lleva a destino

Tabla de ruteo:

Se almacena la información de topología, la información que tiene el host acerca de la red.

- Hay un peor match, (0.0.0.0), en esa entrada el dato coincide SIEMPRE y va al default gateway que es el router.

- Las entradas que indican mi DHCP server (ej: 192.168.0.1), mi IP de la pc, la del router, la del broadcast, la del default gateway, etc las genera mi router.

Lista de interfaces:

Todas las interfaces que tiene la pc para el sistema operativo.

Como si la pc tuviera múltiples placas de red (la ethernet, la wireless, virtual)

Tipos de ruteo:

Depende de si la información en la tabla de ruteo es estática (configuración que el administrador hace y queda aunque haya cambios en la red) o dinámica.

- **Estático:** Funciona siempre y cuando tenga un solo camino y sea la única forma de llegar. Si hay varios caminos el ruteo estático está en desventaja.
Si configuraste 2 caminos en un router se fija la prioridad y por más que uno se caiga si era el prioritario tampoco llega.
- **Dinámico:** Bueno para cuando tenes varias rutas, si una se cae puedo ir por la otra. Tengo que activar el protocolo de ruteo del o los routers (en cada uno) y se va a comunicar con los vecinos para contarle las redes que conoce.
Un router elige el camino mejor y lo incorpora en la tabla de ruteo, la otra posibilidad queda guardada en la base de datos del routing protocol, y si se cae el primer link el router va a revisar esas rutas de la tabla de ruteo y las va a sacar, además se activa la ruta alternativa.

Ventajas del ruteo dinámico: Intercambio automático de dirección sin intervención y la posibilidad de usar múltiples caminos de forma dinámica.

Los protocolos de ruteo son siempre dinámicos.

Otras clasificaciones:

1. **Single-Path:** Sólo pueden elegir un camino una vez, eligen el que creen mejor y el otro queda a la espera
 2. **Multi-Path:** Capacidad de usar múltiples caminos
-
1. **Plano:** El mismo proceso va a correr en todos los routers, y todos intercambian información entre sí. Es una dificultad a la escalabilidad.
 2. **Jerárquico:** Permite hacer áreas entre una cantidad de routers, estas áreas se conectan, hay procesos de ruteo reducidos, y entre áreas se pasan información consolidada. No todos hablan con todos, se puede escalar.
-
1. **Interior:** Un protocolo que corro dentro de mi red interna.

2. **Exterior:** Cuando me quiero conectar con otra compañía, por fuera, quiero intercambiar información más consolidada, intercambiar menos detalle porque no quiero que conozca toda mi topología.

Dos formas de hacer el cálculo para saber cual es el mejor camino.

1. **Vector Distancia:** Calculan la distancia, cantidad de saltos, menor cantidad de saltos es mejor ruta
2. **Estado del enlace:** Toman en cuenta la distancia pero además características del enlace (capacidad, retardo, confiabilidad, etc). Es más complejo que vector distancia.

Ejemplos:

| Routing Protocol | Static Dynamic | Single-Path Multi-Path | Flat Hierarchical | Interior Exterior | Link State Distance Vector |
|------------------|-------------------|---------------------------|----------------------|----------------------|-------------------------------|
| RIP | Dynamic | Single-Path | Flat | Interior | Distance Vector |
| IGRP | Dynamic | Multi-Path | Flat | Interior | Distance Vector |
| OSPF | Dynamic | Multi-Path | Hierarchical | Interior | Link State |
| EIGRP | Dynamic | Multi-Path | Flat | Interior | Adv. Dist. Vector |

Objetivos de un Routing Protocol:

- Flexible: Posibilidad de configurarlo, capacidad de adaptarse a cambios en las variables de red como ancho de banda y retardo.
Rápida adaptación a los cambios en la topología de la red.
- Óptimo: A partir de una topología dada encuentre el mejor camino para ir de A a B
- Rápida convergencia: Converger = cambios de topología por desapariciones de links toma tiempo. Se busca reconfigurar los caminos de la red de la forma más rápida.
- Robusto
- Simple

RIP:

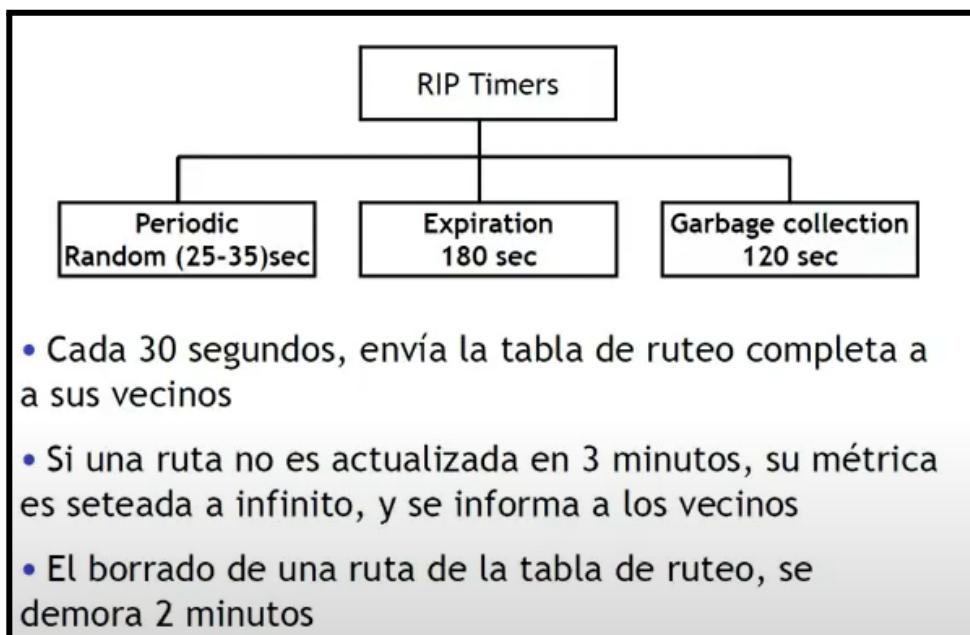
- RFC 1058 (version 1) 2453 (version 2)
- Distance-vector, interior gateway protocol
 - con “split horizon” y “poison inverse”
- Optimiza la métrica
 - hop count (máximo 15 hops, tamaño de red limitado)
- Encapsulado en datagramas UDP
 - puerto 520
 - entrega“no confiable”

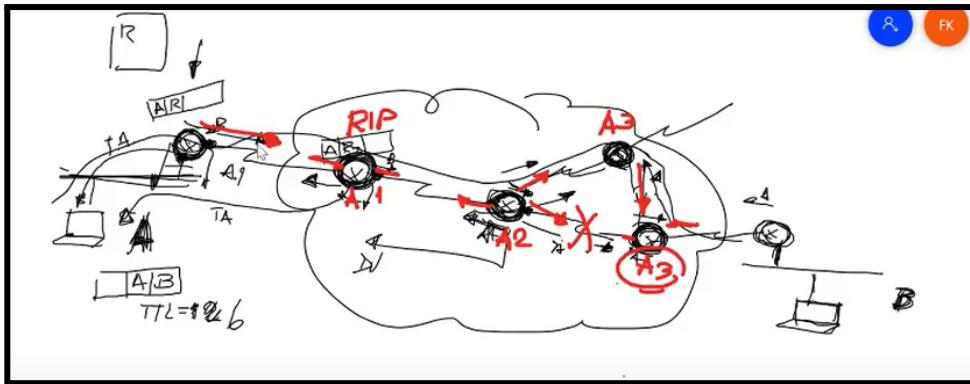
La versión 1 es classful (corresponde una clase y una máscara natural). Ej: Un router conoce la red 10 aunque implementaste la 10.1.1 /24

La versión 2 es classless. Ej: un router conoce la red 10.1.1 /24 otras no.

Actualmente no se considera como protocolo de ruteo en redes corporativas.

Cómo funciona RIP:





Una entrada en la tabla de un router es válida siempre y cuando el router envíe updates periódicamente.

Si agrego routers corriendo RIP sólo puedo llegar a 15 saltos. Tengo un límite en el tamaño de mi red si implemento RIP.

Inicialización

Envía un request a todos los vecinos (broadcast) solicitando sus tablas de ruteo completas.

No realiza Neighbor Discovery, envía broadcasts y no recibe confirmación

Confiabilidad



Se basa en la retransmisión periódica de toda la información

Cada vecino toma lo que informa el broadcast, lo asume como cierta y adopta, no chequea que sea su vecino realmente.

Subredes (Version 2)

Incluye información de subred en la tabla de ruteo y la informa en las actualizaciones a sus vecinos

Seguridad (Version 2)

Password Opcional de 16 bytes (cleartext). Evita la existencia de black-holes (routers que informan todas las redes con métrica 0)

Fácil de quebrar

OSPF:

OSPF

Open (no-proprietario) Shortest Path First

- RFC 2328 (OSPF version 2)
- Protocolo interior recomendado para TCP/IP
 - link state utiliza el algoritmo de Dijkstra
- Ventajas
 - converge más rápido que RIP
 - intercambia menos información que RIP
- Corre directamente sobre IP (no UDP/TCP); protocolo número 89

Aclaraciones:

- Cada router que corre OSPF arma un grafo completo de toda la red, establece relaciones entre los vecinos.
- Converge más rápido que RIP porque no tiene estos timers
- Intercambia menos información que RIP porque no envía cada 30 segundos nada, sólo cuando cambia la topología de la red.

Métrica optimizada

- hop-count, delay, throughput, etc.

Balanceo de carga

Cuando existen dos rutas con la misma métrica, puede enviar tráfico por ambas rutas

Confiabilidad

- Realiza Flooding, con confirmación de los vecinos
- Checksum de los mensajes

Flooding: Cuando hay cambio de topología lo notifico a todos a la vez porque lo tengo en mi grafo.

Subnets

Diseñado para trabajar con VLSM y CIDR

Seguridad

Contraseña simple cleartext

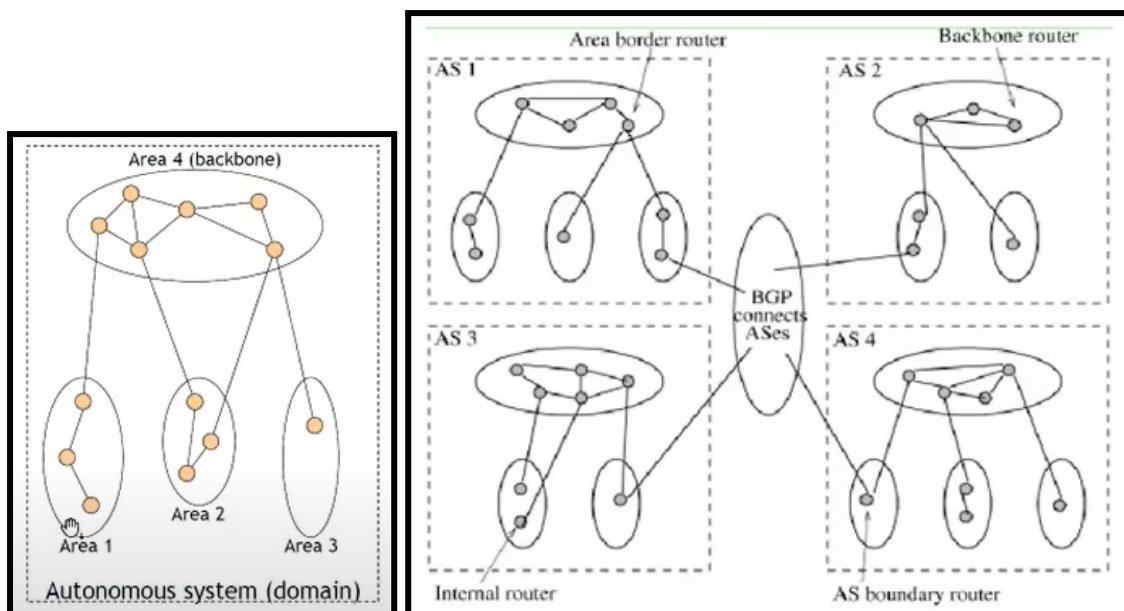
MD5 - preshared key

Jerarquía en OSPF - Sistemas autónomos

El dominio de ruteo se divide en “Areas”

- Backbone y areas conectadas
- Jerarquía de 2 niveles. Permite mantener pequeñas las bases de SPF
- Cada área corre una copia del Link-State Protocol
- Los routers de borde realizan summarización de rutas e intercambian menos información

Puedo tener routers de diferentes tamaños y no necesariamente todos grandes como RIP.



DHCP - Dynamic Host Configuration Protocol

Definido en RFC 2131

- Es un protocolo cliente servidor perteneciente a la capa de aplicación.
- Derivado del BOOTP, protocolo que permite la inicialización de computadoras sin disco rígido. (Hoy está en todos los dispositivos).
- Centraliza y administra la asignación de direcciones IP (Indicamos el pool de direcciones y le da al cliente que le pide y guarda registro de ip y host).
- Mantiene un registro de la IP asignada a cada cliente.

Tipos de asignación de direcciones IP

1. Estática
2. Dinámica

Todo host debe tener:

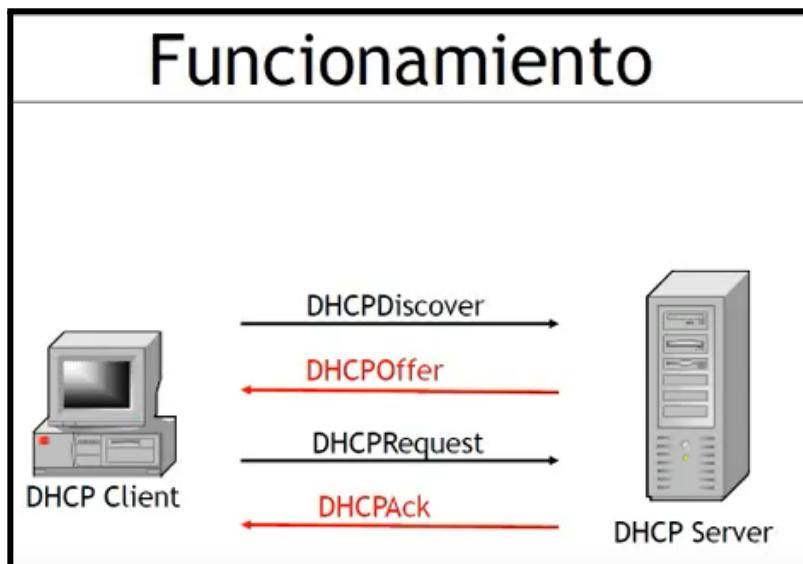
Si quiero salir de mi red y navegar en internet.

1. Dirección IP única
2. Máscara de subred
3. Default Gateway: Si quiero salir de mi red para internet.
4. Servidor DNS

Asignación dinámica - ventajas:

- Elimina la necesidad de llevar un registro de direcciones asignadas (facilita la administración): El DHCP lo hace automáticamente, no lo tiene que gestionar el administrador de la red.
- Facilita la modificación del espacio de direcciones de una red: En subnetting si quería hacer subredes tenía que modificar la máscara en todos los dispositivos, si uso DHCP lo cambio solo en el servidor y todas las pcs luego de reiniciarlas toman una nueva IP con la configuración.
- Permite la utilización eficiente de un espacio de direcciones reducido (más hosts que direcciones IP disponibles - ISP): Si tengo menos direcciones IP que potenciales clientes, DHCP me permite reutilizar, o sea prestar y asignar a otro cliente un instante más tarde.
- Elimina la existencia de errores en la configuración: Elimina errores manuales ejemplo poniendo mal la ip o la máscara.
- Permite asignar a cada host todos los parámetros de configuración junto con la dirección IP: No solo da una ip, sino máscara, default gateway, etc.

Cómo funciona:



1. El cliente envía un **DHCPDiscover** a una dirección especial y se convierte en un broadcast de red LAN, porque el cliente está iniciando y no tiene una dirección IP, sino que quiere obtener una.
2. 1 o varios servidores DHCP le hacen una oferta de ip (**DHCPOffer**)
3. El cliente selecciona una (**DHCPRequest**)
4. El servidor le confirma la asignación de la IP. (**DHCPACK**)

Lo paradójico del servicio: Es una aplicación (protocolo de la capa de aplicación) que corre sobre TCP/IP y se utiliza para repartir direcciones IP, es decir, yo encapsulo el mensaje DHCP en un protocolo de capa de transporte en un datagrama IP cuya dirección origen y destino no tengo ni conozco.

El cliente envía el mensaje con:

- La dirección IP origen 32 todos ceros (red)
- La dirección IP destino 32 todos unos (broadcast)

Ambas direcciones especiales se usan para esto y nada más.

-En el mensaje **DHCPDiscover** no envío solo a un broadcast de LAN sino a un broadcast a todos porque no sé en qué red estoy parada.

-En el mensaje de oferta **DHCPOffer** el servidor si sabe su IP pero no le puede contestar al cliente porque el cliente no la tiene. La respuesta está encapsulada en un datagrama con dirección IP destino todos unos (Broadcast).

- NO se puede usar TCP para este servicio DHCP porque TCP es un protocolo de capa de transporte orientado a la conexión y necesita abrir una conexión antes de intercambiar info (imposible con DHCP porque el cliente aún no tiene IP). Para este servicio DHCP se usa el protocolo de transporte UDP.

DHCP Discover:

-El cliente inicializa una versión limitada de TCP/IP y envía un pedido de dirección IP a los servidores DHCP.

-El mensaje posee dirección origen 0.0.0.0 y dirección destino 255.255.255.255

-Posee la dirección física del cliente y el nombre del host: O sea la MAC Address. Así cuando el server le hace una oferta identifica quién se lo está pidiendo

Se realiza cuando:

- Se enciende por primera vez el DHCP client
- El DHCP server rechaza un pedido de dirección IP específica.
- El cliente con dirección IP asignada, decide liberarla y solicitar otra.

DHCP Offer:

Todos los DHCP Servers que reciben el request responden con una oferta con:

1. Dirección de hardware del cliente: Identificar claramente a qué cliente le estoy haciendo la oferta.
2. Dirección IP destino 0.0.0.0
3. Una dirección IP ofrecida.
4. La máscara de subred
5. Duración de la asignación (Lease): Por cuanto tiempo se la estamos prestando
6. Una identificación del servidor (dirección IP)

El DHCP Server reserva la dirección IP ofrecida a la espera que el cliente confirme que la quiere. El cliente DHCP selecciona la dirección IP de la primera oferta recibida. (en la práctica hay muchos más datos del fabricante que el cliente elige, no solo la primera)

IP Lease Selection o DHCPRequest:

Después de recibir al menos 1 oferta el cliente envía un broadcast a todos los servers indicando la oferta aceptada.

El mensaje se envía como un Request, indicando dirección IP del servidor oferente aceptado.

Todos los DHCP servers rechazados recuperan la dirección ofrecida y queda disponible para responder una nueva oferta.

IP Lease Acknowledge:

El DHCP Server cuya oferta fue aceptada, envía una confirmación positiva al cliente (DHCPACK).

Este mensaje contiene la dirección IP asignada y otros valores de configuración. (Ej: cuánto tiempo es válida).

Cuando el cliente recibe la confirmación, TCP/IP está completamente inicializado y puede comunicarse en la red.

El cliente al recibir el ACK antes de activar la IP va a hacer un ARP gratuito para asegurarse que nadie lo tiene.

Puede suceder que el cliente reciba una confirmación negativa (NACK).

IP Lease NACK:

Sucede por 2 razones:

1. El cliente intenta renovar una asignación anterior y la IP ya no está disponible.
2. La dirección IP es inválida porque el cliente se ha movido físicamente de subred.

Otros mensajes relacionados con protocolo DHCP:

- **DHCPDecline:** El cliente indica al servidor que la dirección está en uso (ARP gratuito). “No quiero la dirección porque alguien la está usando”.
- **DHCPRelease:** El cliente libera la asignación, cancelando el lease. Se hacía antes cuando el protocolo era inestable.
- **DHCPIinform:** El cliente solicita sólo los parámetros de configuración adicionales.

Intento de renovación:

Todo cliente DHCP intenta renovar su asignación (Lease) cuando ha pasado la mitad (50%) del tiempo de asignación.

Envía directamente el mensaje (DHCPRequest) directamente al Server que le otorgó la dirección.

Si el DHCP Server está disponible, envía un ACK.

Cuando el DHCP Client se inicializa, intenta obtener la misma dirección IP, del mismo Server.

Si el server no contesta cuando el cliente solicita el 50% del tiempo, sigue usando la ip hasta la mitad del nuevo intervalo, o sea el 25% del 50% restante de tiempo y luego vuelve a intentar la renovación y si sucede lo mismo vuelve a intentar por tercera vez.

Una vez excedido el 87.5% (tercer intento) del tiempo sin respuesta, o recibido un NACK, se inicializa el proceso DHCP de nuevo, descartando la IP dada anteriormente asumiendo que ocurrió un error.

-Tener en cuenta el lease/tiempo de vigencia, si es corto se van a tener renovaciones de dirección IP y si es demasiado largo las IPS no se van a liberar y en algún momento me voy a quedar sin IPS libres.

-Tener en cuenta en entornos corporativos otorgar siempre la misma IP para que el administrador de red sepa qué IP corresponde a qué persona y pueda otorgar permisos en el firewall. Siempre que yo como cliente pida la IP me va a dar la mía, mientras yo no la pida DHCP se la presta a otro y ese va a tener mis mismos permisos.

Consideraciones de diseño:

No son tan vigentes ahora porque la aplicación de DHCP son sólidas.

- Es común dividir el espacio de direcciones disponibles en 2 DHCP Servers para aumentar la disponibilidad (más ofertas). Si un cliente no recibe una IP, no puede operar la red.
- El pool de direcciones que reparte el DHCP Server excluye un rango IP reservado para asignación estática (routers, servidores, impresoras, etc).
- Es necesario configurar los routers para permitir el paso de DHCP Request (broadcast).

Opciones adicionales del DHCP:

Son infinitas, porque el protocolo es flexible, y los fabricantes las utilizan.

DHCP Options (RFC 2132)

- 1-Subnet mask
- 3-Router Option
- 6-Domain Name Server
- 33-Static Route Option (Default gateway)

Ejemplos de tramas DHCP:

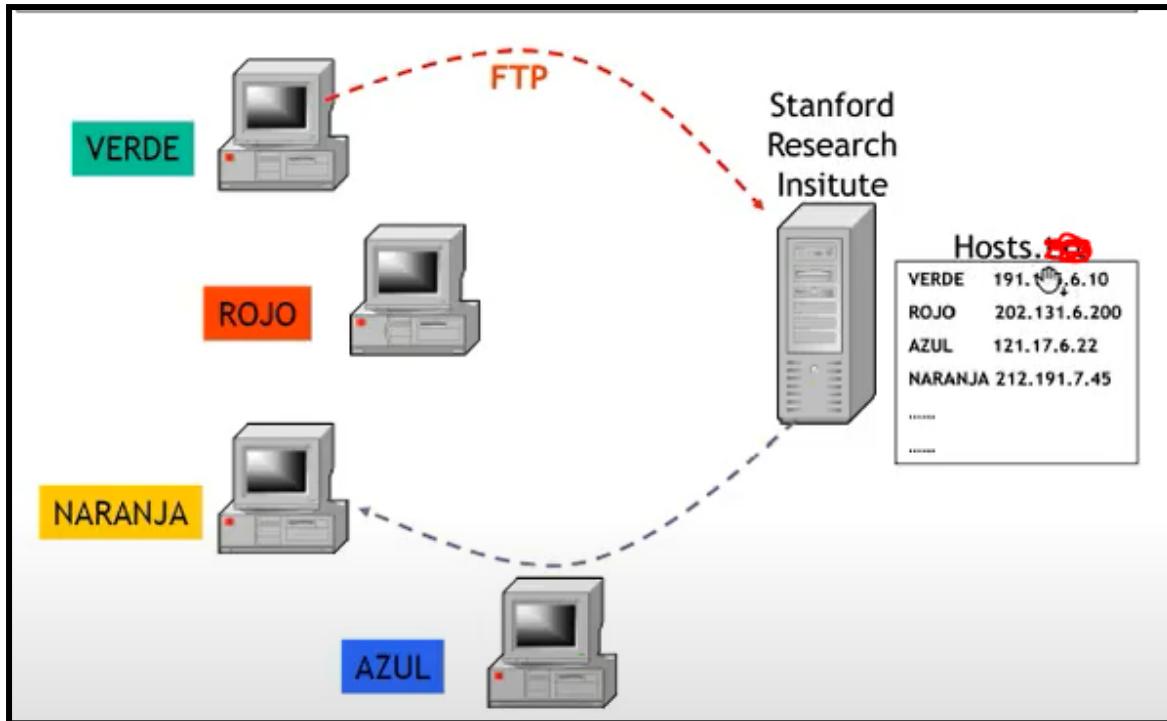
| Time | Source | Destination | Protocol | Info |
|------------|---------------|-----------------|----------|--|
| 3 0.259998 | 0.0.0.0 | 255.255.255.255 | DHCP | DHCP Request - Transaction ID 0x8071d9b5 |
| 4 0.280627 | 10.200.127.14 | 255.255.255.255 | DHCP | DHCP ACK - Transaction ID 0x8071d9b5 |

DNS - Domain Name System

RFC 1034 y 1035

-Sistemas de redes de dominio

-Utilizamos un nombre para no recordar una dirección IP (las comunicaciones se realizan con protocolos IP)



Host File:

Previo al DNS

Mecanismo dentro de un dispositivo donde un archivo de texto permite asociar un nombre con una dirección IP. No es recomendable porque es un archivo estático y poco extensible.

El host estaba en Stanford, cuando empezó a hacerse intensivo el uso del hostfile, hubieron problemas que originaron la creación de DNS

- El archivo es plano entonces si alguien registra cierto nombre ya nadie más lo puede usar.
- El archivo se tornó muy grande
- Era un único punto de falla para toda la red
- Los dispositivos se tenían que conectar varias veces al día para obtener la última versión y eso saturaba los recursos del servidor de Stanford.

DNS:

Es un servicio cliente servidor.

Es una base de datos distribuida y un espacio de nombres jerárquico.

Compuesto por:

1. Resolvers: Cliente, corre en cada dispositivo. Envía el pedido de resolución entre la aplicación y el servidor de nombres.
2. Name Servers: Reciben el pedido y resuelven el nombre de Host a una dirección IP. Responden las consultas de los clientes.

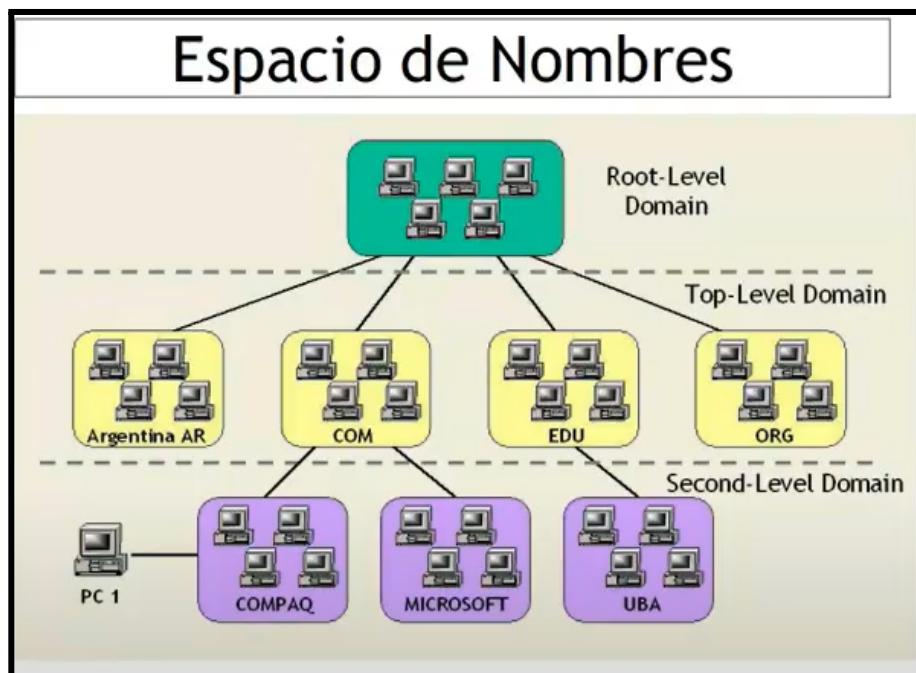
Espacio de nombres:

- Organización lógica, cómo se construyen y delegan los dominios
- Diseño jerárquico y formato árbol.

-Levels

1. La raíz (root level) se identifica lógicamente por un punto (.). Lo más alto de la jerarquía. No utiliza etiquetas.
2. Los top level domain identificados por .com .edu .org .gov .[país] (códigos ISO)
3. Second level o dominios inferiores: Pueden contener hosts u otros dominios llamados sub-dominios.

Por ejemplo el dominio Microsoft.com puede poseer hosts:
[ftp.microsoft.com](ftp://ftp.microsoft.com) o sub-dominios como dev.microsoft.com



Host names: Se agregan al comienzo del nombre de dominio. Generalmente se los identifica con el “Fully Qualified Domain Name” (FQDN) cuando se tiene el nombre del host con todos los dominios y subdominios hasta el root e incluso el .

Implementación:

Hay un servidor DNS encargado de conocer esa IP, responsable de traducir los nombres que terminan con ese dominio.

El proveedor da el servidor DNS físico que hostea mi dominio.

ZONAS (implementación):

1. **Zona de autoridad:** Porción de dominio por la cual un servidor es responsable.
El DNS Server responsable de la zona posee el archivo de la zona (tiene asociación nombre + IP para ese dominio)

Un único Server puede mantener múltiples zonas (lo que pasa en la realidad que contratamos servicios de dominio).

Es un archivo de zona hosteado en un servidor DNS.

Roles de servidores:

1. Tiene la copia “principal”
2. Recibe una copia para agregar redundancia al servicio.
3. -
4. Servidores que no hostean ninguna zona sino que están puestos deliberadamente para resolver consultas de los clientes.

Servidores

Roles de los servidores:

Primary Name Server

- Los archivos de información de la zona se almacenan localmente

Secondary Name Server

- Obtiene la información de zona del Master Name Server

Master Name Server

- Fuente de información para un Secondary Server. Pueden ser Primary o Secondary Servers

Caching Only

- No almacena información de zona

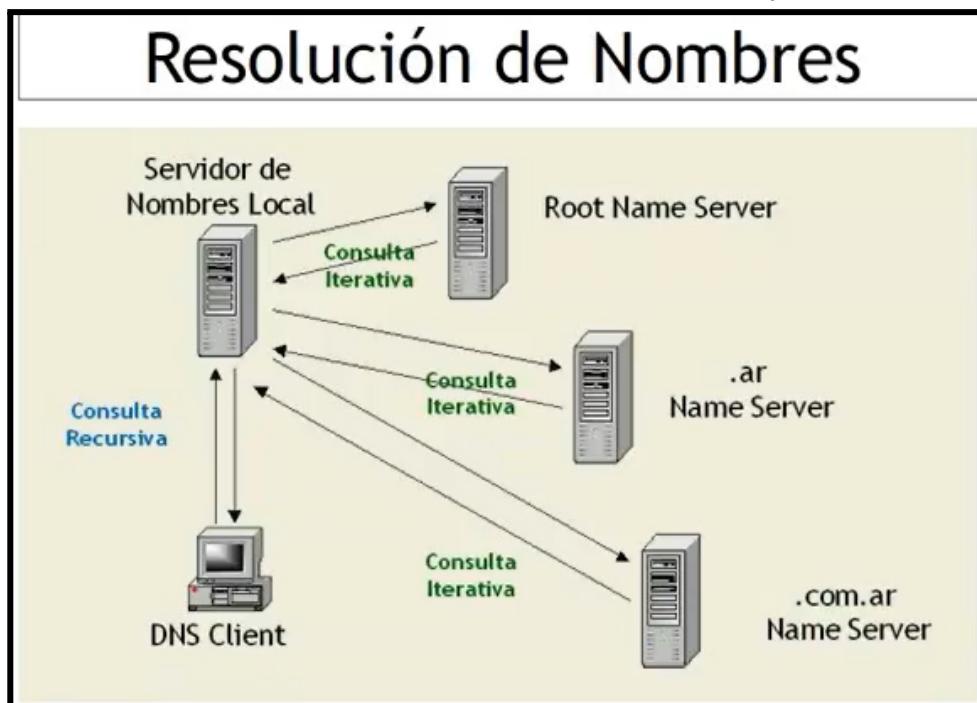
DNS servers configurados por el DHCP server, o sea, el router que configuró el proveedor de internet:

```
DNS Servers . . . . . : 2800:810:100::d  
2800:810:100:4:181:47:248:145  
2800:810:100::5  
2800:810:100:4:181:47:248:145  
181.45.64.77  
190.55.60.129  
181.47.2.4.164
```

Puede ser que sean solo de caché, para que el cliente pueda resolver nombres, o sea para que cuando busquemos un dominio (google) la pc va a ir en orden por cada uno de esos server y va a preguntar la dirección IP del dominio (google).

Resolución de Nombres:

- El cliente (resolver) tiene definido un dns server (los de la foto de arriba) y hace una consulta recursiva al servidor de dominio.
- El server busca en su caché una consulta similar y si no encuentra hace una búsqueda iterativa (el cliente comienza por preguntar en root servers, este delega a un siguiente servidor (iteración) y así. Hasta llegar al server que lo tenga)
- Este server va a dar la referencia a los servidores DNS donde está el archivo de zona donde UTN registró el dominio, donde dio sus IPS para continuar la resolución.
- La respuesta va a ser cacheada (el archivo de zona tiene su TTL seteado por el administrador del dominio) por el servidor de nombres local y se la da al cliente



Si quiero prepararme para hacer un cambio (cambiar la dirección IP de un server) tengo que reducir el TTL al mínimo para que tenga un impacto inmediato en los clientes, sino puede ocurrir que algunos por el TTL largo sigan apuntando a la vieja dirección IP y los nuevos a la nueva IP y sería un inconveniente.

El TTL no es el mismo TTL de los datagramas del router.

Tipos de consulta:

Los siguientes:

Consulta recursiva:

El servidor de nombres consultado está obligado a responder con los datos o con un error. La respuesta es la IP o “no existe”.

Consulta iterativa:

El servidor consultado responde con su mejor respuesta. Puede ser el nombre resuelto o una referencia a otro servidor de nombres, que pueda ser capaz de responder la consulta.

Consulta inversa:

El resolver solicita el nombre de Host asociado a una IP.

Preguntarle a un servidor por una IP cuál es el nombre asociado, al revés.

En traceRoute cuando alcanza el destino toma la IP para hacer una consulta inversa y obtener el nombre.

Cacheo de datos por servidores DNS:

Porque reciben un TTL que viene con cada respuesta.

Caching & TTL

- Los DNS Servers cachean las consultas iterativas
- Cada entrada en cache tiene asociado un tiempo de vida (TTL)
- Cuando éste expira, la entrada es borrada
- El TTL remanente es enviado al Resolver cuando se responde una consulta recursiva

Protocolos y puertos:

-Puerto 53 → DNS.

-El servicio corre sobre UDP para que la consulta del cliente sea ágil la respuesta, ya que UDP es no orientado a la conexión

-Hay una variante que usa TCP, si la respuesta del server al cliente es limitada en tamaño por retrocompatibilidad, si la respuesta no entra se indica en el mensaje el truncamiento.

-TCP se usa para hacer transferencias de zona, cuando un servidor secundario se conecta con el primario para hacer una copia del archivo de zona y mantenerlo actualizado

- El servicio DNS server escucha peticiones en el puerto 53, tanto de TCP como UDP
- La petición se realiza en UDP
- Si se recibe una respuesta truncada, se realiza nuevamente usando TCP

Resource Records - Registros de recursos:

- Están dentro del archivo de zona
- Son diferentes entradas
- Son de diferentes tipos:
 1. Tipo A: Asocia un nombre a una dirección IP. Lo básico
 2. Tipo MX (mail Exchange): Los mails/mensajes encolados del servidor de salida de correo, este servidor consulta en el DNS a través del tipo MX cuál es la dirección IP del servidor de email que hostea a ese dominio.com.

- **A Host Record** : asocia estáticamente un nombre de Host con una dirección IP. Comprende la mayor parte del archivo y lista todos los Hosts dentro de la zona

```
www    IN A    200.69.225.145
rhino  IN A    200.26.65.12
```

- **MX Mail Exchange** : asocia un dominio de email con la dirección de los servidores de correo

```
@      IN MX   [10] mailhost
@      IN MX   [20] mail1.infovia.com.ar
```

3. CNAME: Es un alias.

- **CNAME Canonical Name** : permiten asociar más de un nombre de Host a una única dirección IP (alias)

```
fileserver1    CNAME  Rhino
ftp           CNAME  Rhino
```

Sniffeando un ping a un dominio:

| dns | | | | | |
|----------------|--------------|--------------|----------|---|--|
| Time | Source | Destination | Protocol | Info | |
| 2156 12.918162 | 192.168.0.14 | 181.45.64.77 | DNS | Standard query 0xc5ca A sistemas.frba.utn.edu.ar | |
| 2157 12.918450 | 192.168.0.14 | 181.45.64.77 | DNS | Standard query 0xaaca AAAA sistemas.frba.utn.edu.ar | |
| 2158 12.929033 | 181.45.64.77 | 192.168.0.14 | DNS | Standard query response 0xc5ca A sistemas.frba.utn.edu.ar A 200.89. | |
| 2160 12.938821 | 181.45.64.77 | 192.168.0.14 | DNS | Standard query response 0xaaca AAAA sistemas.frba.utn.edu.ar SOA ns | |

El primer destino es el primero server DNS de la configuración (imagen más arriba), y este contesta la IP.

El sistema operativo consulta también por el récord **AAAA --> IPv6** y el A.

| | | | | |
|---|--------------|--|-----|---|
| 192.168.0.14 | DNS | Standard query response 0xc5ca A sistemas.frba.utn.edu.ar A 200.89.153.25 | | |
| 192.168.0.14 | DNS | Standard query response 0xaaca AAAA sistemas.frba.utn.edu.ar SOA ns1.frba.utn.edu.ar | | |
| Answers | | | | |
| sistemas.frba.utn.edu.ar: type A, class IN, addr 200.89.153.25 | | | | |
| Name: sistemas.frba.utn.edu.ar | | | | |
| Type: A (Host Address) (1) | | | | |
| Class: IN (0x0001) | | | | |
| Time to live: 58214 (16 hours, 10 minutes, 14 seconds) | | | | |
| Data length: 4 | | | | |
| Address: 200.89.153.25 | | | | |
| <i>[Request In: 2156]</i> | | | | |
| 1138 9.122074 | 192.168.0.14 | 181.45.64.77 | DNS | Standard query 0x6a28 A www.facebook.com |
| 1139 9.122394 | 192.168.0.14 | 181.45.64.77 | DNS | Standard query 0x4e24 AAAA www.facebook.com |
| 1141 9.137076 | 181.45.64.77 | 192.168.0.14 | DNS | Standard query response 0x4e24 AAAA www.facebook.com CNAME star-mi... |
| 1142 9.137457 | 181.45.64.77 | 192.168.0.14 | DNS | Standard query response 0x6a28 A www.facebook.com CNAME star-mini... |
| Authority RRs: 0 | | | | |
| Additional RRs: 0 | | | | |
| Queries | | | | |
| > www.facebook.com: type A, class IN | | | | |
| Answers | | | | |
| > www.facebook.com: type CNAME, class IN, cname star-mini.c10r.facebook.com | | | | |
| > star-mini.c10r.facebook.com: type A, class IN, addr 31.13.94.35 | | | | |
| <i>[Request In: 1138]</i> | | | | |
| [Time: 0.015383000 seconds] | | | | |

Facebook retorna un alias del servidor más cercano (CDN).

Registro de Nombres

La administración local del TLD .ar lo realiza
Cancillería (MRECIC). Para dar de alta un dominio,
basta con completar los datos en <http://www.nic.ar>

Para verificar si un dominio ya se encuentra registrado
www.nic.ar o www.nic.

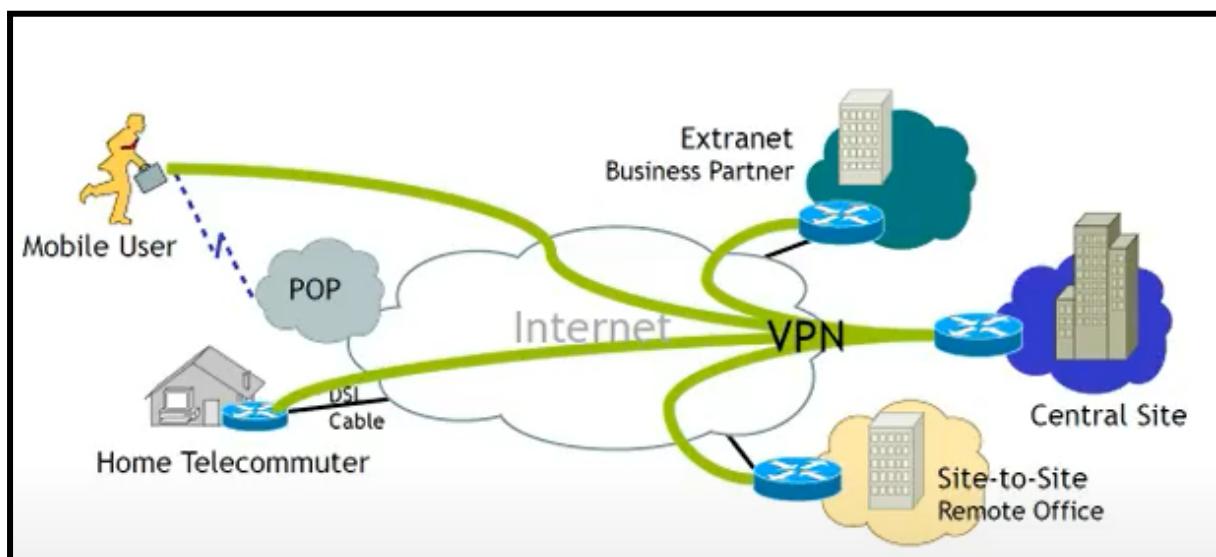
VPN - Virtual Private Networks:

Estándar IPSec Lab (para hacer túneles)

- Red privada virtual
- Armar una red que parezca privada sobre una red que esencialmente sea pública.
- Interconectando puntos de la red para que al ojo del usuario parezca “su” red.
- Lo que pasa por internet está “protegido” si usamos VPN.

Tipos de VPNS:

1. De acceso Remoto: De un usuario (ej: teletrabajador que se conecta a su oficina) conectándose a la través de un cliente de VPN creando un túnel a servicios privados.
2. Site to Site: 2 redes se conectan usando gateways a través de internet permitiendo que el tráfico intercambiado pase por la red pública pero sea protegido.



Variantes del protocolo:

Anecdóticas porque ahora el estándar es IPSec que resolvió casi todas las limitaciones de implementaciones de los otros.

La IPSec nace como opción de protección de conexiones en IPV6 de forma nativa. y se asocia luego a IPV4.

Variantes de protocolos:

•PPTP

Desarrollado por MS, no provee encripción (capa 2)

•L2TP

Desarrollado por Cisco y MS, mejora al PPTP incluyendo encripción a través de IPSec (ESP) (capa 2)

•IPSec

Considerado el estándar actual (capa3)

•SSL

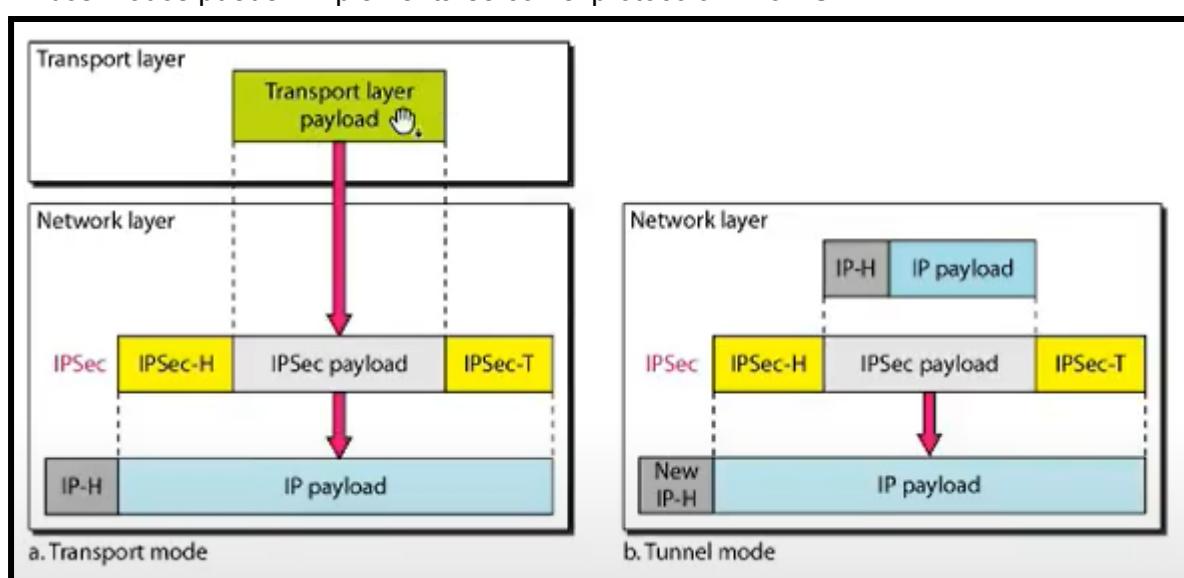
No requiere cliente de software (capa 4)

IPSec:

Suite de Protocolos destinados a proveer seguridad a nivel de red.

Modos de operación:

Ambos modos pueden implementarse con el protocolo AH o ESP



1. Modo transporte

- La conexión a nivel transporte está encriptada
- El segmento que está en capa 4 cuando pasa a capa 3 es encriptado y encapsulado en un datagrama IP.
- Está encriptado en datos pero se puede saber el origen y destino.
- Es entre 2 hosts. (A y B en dibujo, las PCS).

-No se suele usar en IPV4, pero todas las de IPV6 van encriptadas por este modo.

2. Modo túnel

-Se establece entre 2 gateways (no entre 2 hosts)

-Todo el tráfico que pasa está encriptado. Se usa mucho para encriptar todo el tráfico

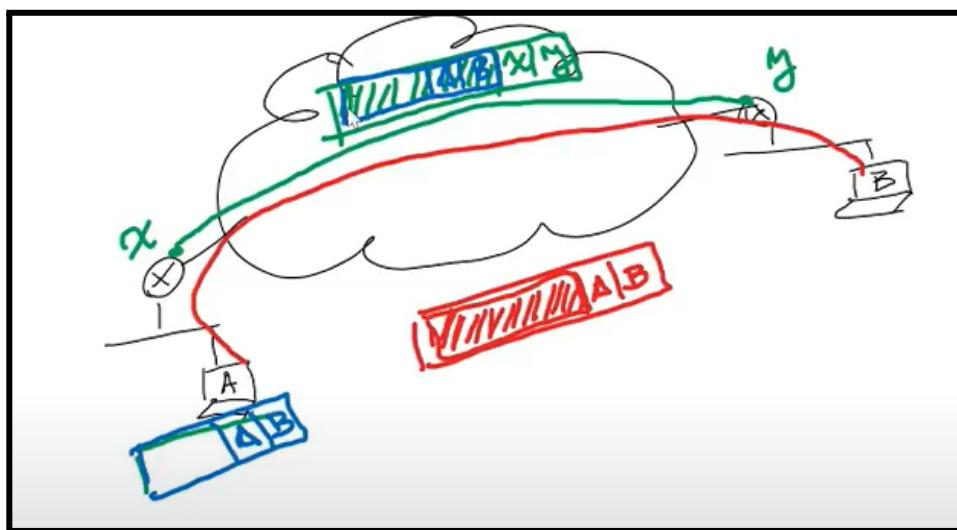
-Cuando se opera en este modo el origen y destino pueden ser direcciones IP privadas (que por internet no podrían ser. Ej: 192.168... o 10.1..).

-Si cuando la estación A envía un mensaje a la estación B sin encriptar y B está fuera de su red, A manda el mensaje al router, el router encapsula ese mensaje en otro, pone una cabecera y los datos son encriptados.

-Por punto anterior, en la comunicación no pueden saber quien es el origen y destino de un mensaje, porque el mensaje que se manda (adentro está el mensaje real encriptado) tiene destino y origen de routers. Si alguien hace sniffing no lo ve.

-En resumen, toma un datagrama completo y lo encripta, lo encapsula y lo manda. La dirección origen y destino son X e Y (routers) y no hosts.

-El túnel se establece cuando uno de los extremos quiera comunicar, si ninguno establece comunicación no se hace. Cuando un extremo recibe un paquete que debe mandarse por el túnel, el router manda un mensaje para establecer ese túnel con la clave. Ahí se establece la fase 1, luego la fase 2 indicando las IPs privadas. → Security association



Rojo: Modo transporte y Verde y azul: Modo túnel

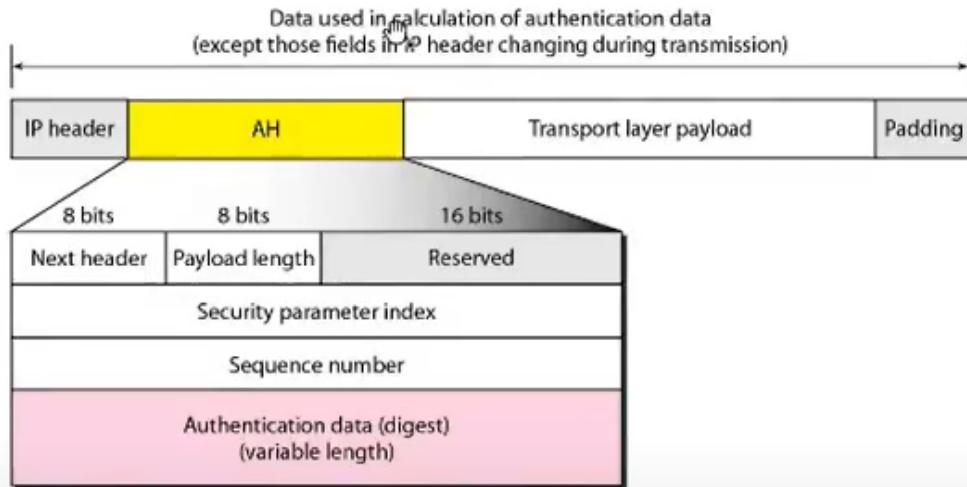
Protocolos:

1. AH protocol: Authentication Header - cabecera de autenticación.

Si lo uso garantizo autenticidad e integridad.

Me garantiza que salió de x a y pero no está encriptado (puedo leer el mensaje pero no alterarlo). AH no encripta.

AH en modo transporte



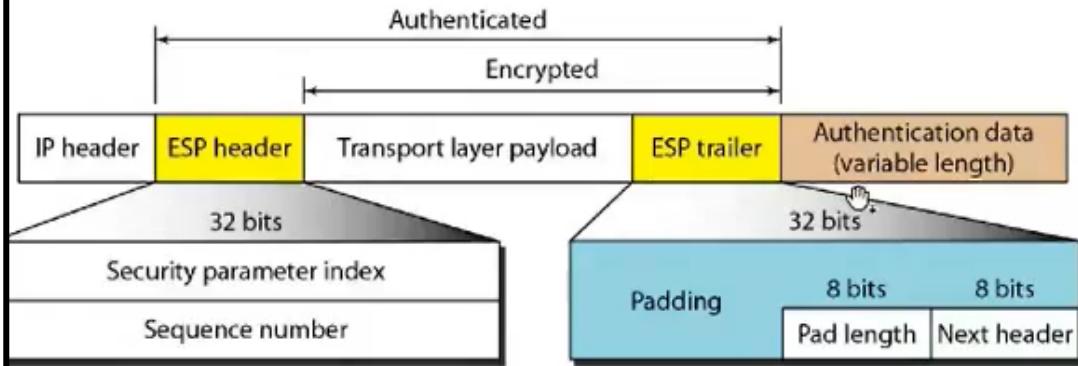
AH provee:

Autenticidad del origen, Integridad de los datos pero no privacidad

2. ESP protocol: Encrypted security payload - carga encriptada

Garantiza autenticidad, integridad y privacidad (lo encripta para que no se pueda ver)

ESP en modo transporte



ESP provee:

Autenticidad del origen, Integridad de los datos y
privacidad

Implementación:

- **Fase 1:** Protocolo de intercambio de claves

Configuración que nos va a permitir declarar los extremos del túnel, los gateways. Se crea una asociación entre la IP del otro extremo y la clave (palabra clave para conectarnos)

IKE Policy (Phase I)

```
crypto isakmp policy 1
    authentication pre-share
    hash sha
    encryption 3des
crypto isakmp key clave_privada address 192.168.3.1
```

Parámetros que necesito para configurar:

1. Dirección IP del extremo (de "y" en la imagen dibujada si me paro en X)
2. Forma de autenticación: Certificados o claves preacordado con los dos extremos.
3. Algoritmo de integridad: Hash, ej: Sha
4. Mecanismo de encripción: Ej. 3des, AES. Para encriptar el intercambio de información para establecer la relación.

- **Fase 2:** Defino configuración del túnel que acabo de establecer.

Definir qué va dentro del túnel, dominio de encripción

Agrego en cada extremo su configuración, en esta tengo una red local, IP privada, (la de la izq. en la foto) y una remota (la de la derecha)

Parámetros: Quién es el peer, qué algoritmos uso de encriptación de datos, integridad, cuál es el modo, define la access list (que flujo se permite) y finalmente pego el criptomap a la interfaz saliente.

Debe tener la misma configuración espejada porque cualquiera puede establecer la conexión y si están distintas el otro extremo la va a rechazar.

2 modos:

1. Rule base
2. Politics base: Más restrictivo

El set peer es la ip privada del otro extremo

Set transform-set es como se encriptan los datos

```
crypto ipsec transform-set ESP-3DES-SHA esp-3des esp-sha-hmac
access-list 102 permit ip X.X.X.X 0.0.0.255 Y.Y.Y.Y 0.0.0.255
crypto map IPSEC 20 ipsec-isakmp
    set peer 192.168.3.1
    match address 102
    set transform-set ESP-3DES-SHA
interface FastEthernet 0/0
    crypto map IPSEC
```

Access List:

Para describir flujos/tráfico.

Son entradas, que se busca match hit en orden de agregación, a la primera que coincide sale. Hay que tener cuidado que no se solapen las entradas, si no coincide ninguna va la default/implícita que dice "deny any any"

Wildcard:

La inversa de la máscara. Un 0 para la red y un 1 para el host.

Agrandar el túnel:

Agregar un registro nuevo a la access list

El túnel no queda establecido, se establece cuando haya necesidad de cruzar tráfico protegido. Mientras las dos redes no se conecten, el túnel no existe (es configuración lógica nada más).

Puede pasar que: Haya fase 1 pero no fase 2 (no se estableció el security association por mal dominio de encripción mal escrito. Fase 1 referido a túnel entre routers, fase dos a redes que pasan por el túnel.

Ejercicios de parcial:

Necesitas 8 bits para tener como máximo 256 (200 host de la red A).
 $10.200.0.0/24$

$255.255.254.0$

Necesitas al menos 6 bits para 64 hosts. (cumple red B y C)
 $10.200.0.0/24$
 $10.200.1.0/26$
 $10.200.1.128/26$

| | |
|--|--|
| Pregunta 7 Finalizado Puntúa 2,00 sobre 3,00 <input checked="" type="checkbox"/> Marcar pregunta | Se requiere particionar el bloque de direcciones IP: 10.200.0.0/23 para asignar a las siguientes redes: A - 220 hosts B - 60 hosts C - 55 hosts Proponga una solución indicando qué dirección de red y máscara (o longitud del prefijo) asigna a cada una. Por ejemplo: A - 10.200.x.y/z; B - 10.200.p.q/; etc. A - 10.200.0.0/24 B - 10.200.1.0/25 C - 10.200.1.64/25 |
| Comentario: B y C se solapan | |

Para la siguiente configuración de direccionamiento con clase:

Dirección IP: 192.168.1.120 Máscara: 255.255.255.224

- a. Indicar la cantidad total de subredes que se pueden asociar sin considerar las prohibidas.
- b. Cuál es el número de la subred a la que pertenece el host individualizado por la IP dato?
- c. Cuál es la máxima cantidad de hosts que puedo direccionar por subred?
- d. Cuál es el número del host individualizado por la IP dato?
- e. Cuál es la dirección de broadcast de la subred dato?
- f. Qué dirección habría que asignar al host nº 18 de la subred nº 5?

Contestar por línea con la letra y el número o dirección IP. Por ejemplo:

a. 20

.....

e. 150.30.40.5

a. 6

b. 96

c. 30

d. 24

e. 192.168.1.121

f. 192.168.1.178

127
145

a. $224 = 11100000 \rightarrow 3$ bits para subred $2^3 = 8$ redes.

b. 96 porque la ip en binario es 00000001.01111000 → los primeros 3 bits para la subred suman 96.

c. 2^5 host = $32 - 2 = 30$

- d. el número del host es 00000001.01111000 los bits del host, los últimos 5 → 11000 = 24
- e. La dirección del broadcast es la última y sino 00000001.01111111 = 127
- f. Subred 1 → de 0 a 31, subred 2 de 32 a 63, subred 3 de 64 a 95, red 4 de 96 a 127 y la 5 de 128 a 159, el dieciochoavo es 145.

22/10

TCP - Transmission Control Protocol

TFC 793/1122

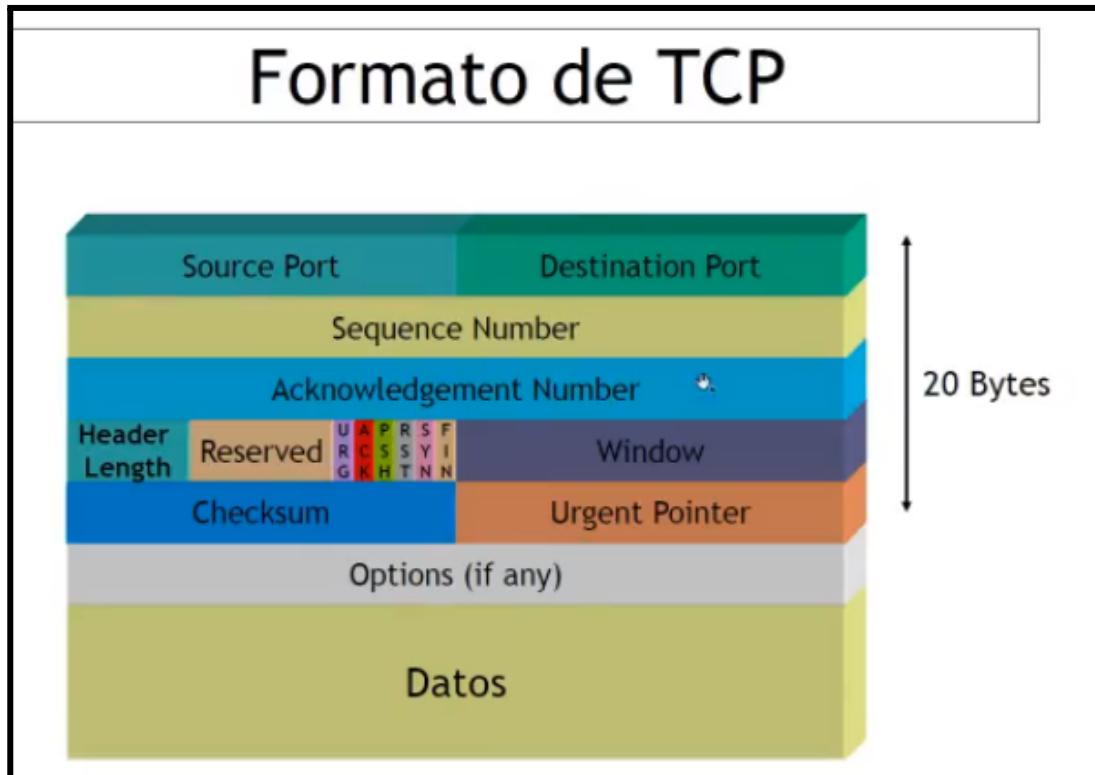
- Capa 4 de transporte.
- Ofrece muchos más servicios que UDP
- Se encapsula en IP
- TCP-IP todo tiene 32 bits de largo

Características:

- Protocolo punto a punto
- Orientado a la conexión: Exclusivamente para dos hosts. Estado mantenido por ambos extremos. Si implemento una conexión con más de dos extremos no será con TCP o convertir el servicio en n servicios de 2 extremos. Si requiero más de dos participantes si o si tengo que ir a UDP.
Cada extremo lleva la cuenta de cómo progresó la conexión (cuánta info se intercambió, próximo segmento a enviar y recibir,etc)
- Multicast y broadcast usan UDP sí o sí.
- Utiliza “segmentos” (se le dice así a la PDU de capa 4 para tcp) generalmente contenidos en un único datagrama IP (sin forzar la fragmentación de IP).
- Confiabilidad alcanzada mediante
 - Confirmaciones
 - Timeouts
 - Retransmisiones
 - Checksum de la cabecera y el cuerpo (mensaje completo): IP hace checksum solo de la cabecera y no garantizaba la integridad de los datos como si hace TCP.
- Podés determinar quién fue cliente y quién servidor porque en una conexión TCP el cliente conoce el puerto del servidor al que se quiere conectar. Y el puerto del cliente es cualquiera. (Ejemplo de consola más abajo).
- La cabecera TCP ocupa 20 bytes en tanto y en cuanto no tenga opciones e impacta en el campo de la cabecera “header length” también.

- En el formato de la trama TCP (imagen debajo) no tengo un campo de largo total, no hace falta, porque está encapsulado en IP que me entrega la totalidad de los packs.

Formato de TCP:



Las opciones se usan bastante en los sistemas operativos, no todas pero si, es lo opuesto a IP donde cayeron en desuso.

1. Source Port - puerto origen:

16 bits.

-TCP usa el “Puerto destino” para identificar el destino final: Dentro de un host (una dirección IP) cuál proceso es al que va dirigido el mensaje.

-Conexión: Cada par de endpoint (extremo)

-End point (extremo): Dirección IP + TCP port.

Esta combinación de 4 valores (IP y puerto origen e IP y puerto destino) es única.

-Un port TCP puede ser compartido por múltiples conexiones: Uno de los endpoints puede ser idéntico pero el otro debe ser inevitablemente diferente. (imagen debajo, uno de los extremos debe ser distinto).

-Podemos alcanzar 65535 (2^{16})

Ejemplo importante: Si abro 2 instancias de browser para conectarme a google tengo 2 conexiones distintas pero con misma IP origen (mía), misma IP destino (google), puerto destino (mismo servicio) pero se discrimina una sesión de la otra con el puerto origen

porque cuando abro una conexión mi sistema operativo me asigna un puerto origen, y otro diferente a la segunda conexión.

Ver en consola estado de conexiones:

Me muestra el protocolo, la dirección local, la dirección remota y el estado de la conexión.

| Active Connections | | | |
|--------------------|--------------------|--------------------|------------|
| Proto | Local Address | Foreign Address | State |
| TCP | 0.0.0.0:135 | 0.0.0.0:0 | LISTENING |
| TCP | 192.168.0.14:53804 | 52.157.250.241:443 | CLOSE_WAIT |
| TCP | 192.168.0.14:53805 | 52.157.250.241:443 | CLOSE_WAIT |
| TCP | 192.168.0.14:53806 | 52.157.250.241:443 | CLOSE_WAIT |
| TCP | 192.168.0.14:53807 | 52.157.250.241:443 | CLOSE_WAIT |
| TCP | 192.168.0.14:53808 | 52.157.250.241:443 | CLOSE_WAIT |
| TCP | 192.168.0.14:53809 | 52.157.250.241:443 | CLOSE_WAIT |
| TCP | 192.168.0.14:53812 | 20.190.160.96:443 | CLOSE_WAIT |

En el ejemplo se estableció de adentro hacia afuera. El local es el cliente.

2. Destination Port:

Same as source port.

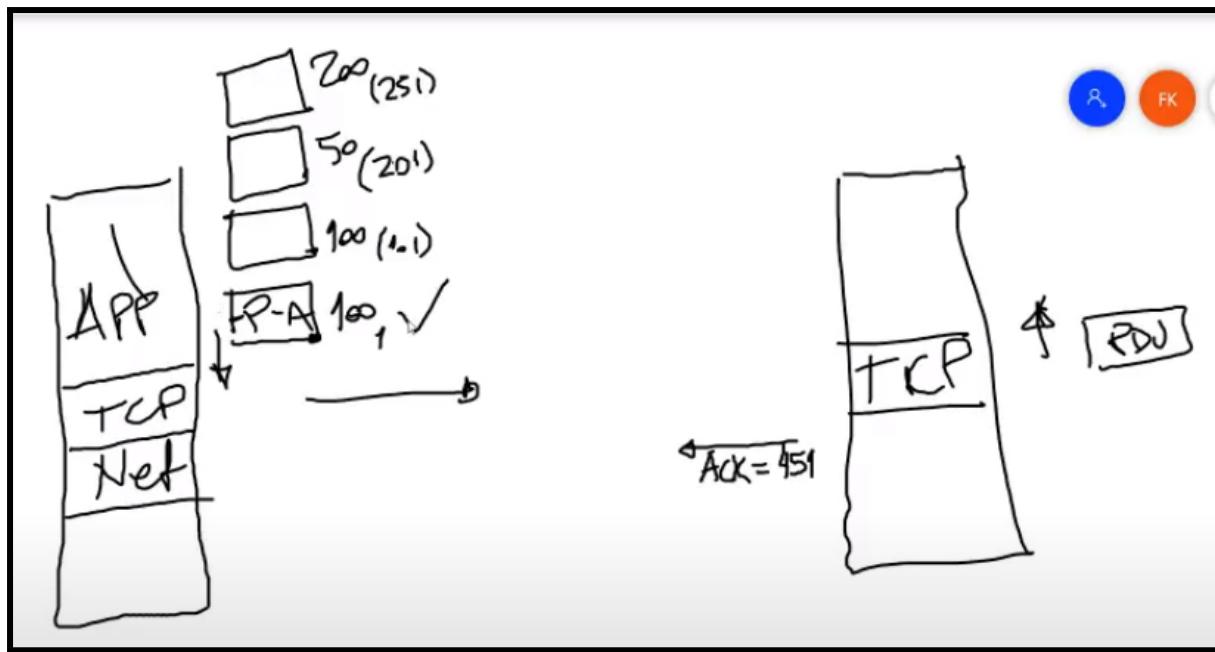
3. Sequence number:

-32 bits

-Número de secuencia del primer byte en este segmento. El primer byte en el campo de datos.

-Cada PDU (mensaje) que la aplicación le entrega a TCP para que transporte y entregue al destino. Si hay varios mensajes a mandar manda cada uno de a uno y en orden con la indicación del byte para que se reciban en ese orden. El receptor envía una confirmación ACK indicando que espera recibir el número indicado para que en origen no lo mantenga en un buffer en caso de pérdida y reenvío.

-Es el número de orden del primer byte en el mensaje cuando se mandan varios mensajes y en el envío se toma como un gran conjunto de bytes.



4. Acknowledgement Number:

- Confirmación
- 32 bits
- Próximo número de secuencia que el emisor de este segmento espera recibir.
- Válido solamente cuando flag ACK = 1
- Tiene que existir la combinación del flag ACK encendido y el valor de número de secuencia para confirmar la recepción de un segmento.

-Son acumulativas y secuenciales: Con un solo mensaje de confirmación es suficiente para garantizar la llegada de varios segmentos - mensajes. Hasta el número indicado es todo lo que recibió, si entre medio se perdió uno el siguiente a ese no lo puede confirmar ya que antes se perdió otro. Ej: recibió 1, 2, el 3 no, y el 4 sí, sólo puede confirmar el 1 y 2.

5. Header length:

- Cantidad de palabras de 32 bits en la cabecera.
- Igual que en IP. En TCP la mayoría de mensajes tiene 20 bytes de longitud por lo tanto suele ser 5 el valor.

6. Window Size:

- 16 bits
- Cantidad de bytes, comenzando por el indicado en el campo ACK (en el segundo y tercer paso de handshake), que el receptor está dispuesto a recibir.

7. Checksum:

- Código de detección de errores.
- 16 bits.

8. Urgent Pointer:

16 bits

-Es un puntero que apunta al último byte de la secuencia de datos urgentes. Me indica donde terminan los datos urgentes.

El siguiente byte que está indicado en el campo sequence number es el primer byte de datos de la PDU de aplicación (lo que se encapsuló) luego de los datos urgentes.

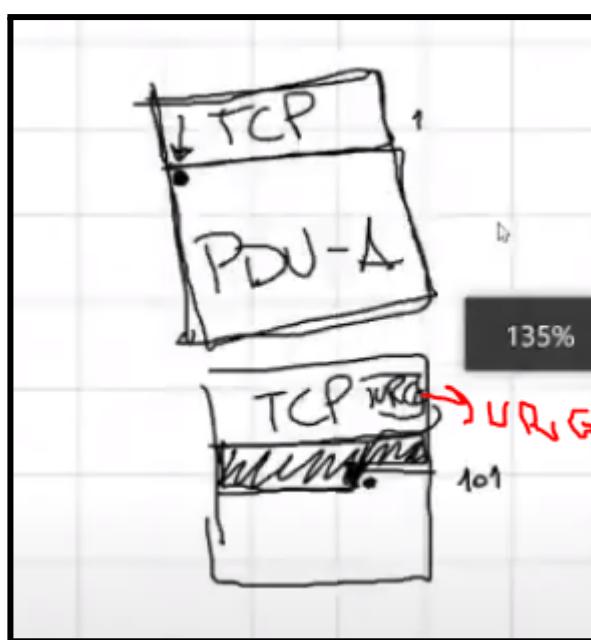
-Se utiliza cuando existen datos/mensajes urgentes que es como un mensaje aparte que no forma parte de la secuencia numerada de mensajes que se nombró antes

-Es un canal de comunicación paralelo para poder enviar estos datos juntos o mientras se envían los otros porque se deben analizar antes que otros.

-Cómo funciona: Existe un flag URG que debe estar activado

Ej: Si algo demora mucho tiempo y te da la opción de abortar todo, este mensaje debería ir antes que el resto.

-Diferencia entre un segmento normal y uno urgente:



9. Options:

-Variable y opcional.

-La opción más común es el MSS (maximum segment size), timestamp, window scale factor. Está en todas las conexiones TCP.

-Descripción más abajo del resumen.

Flags:

| Flag | Descripción |
|------|--|
| URG | El contenido de Urgent es válido |
| ACK | El contenido de ACK es válido |
| PSH | Push, procesar tan pronto como pueda |
| RST | Esta conexión debe reiniciarse |
| SYN | Sincronizar números de secuencia |
| FIN | El emisor no tiene más datos para enviar |

- **PSH:**

-El destino tiene un mecanismo para confirmar que recibió los mensajes-segmentos TCP, y es el ACK encendido para eso puede tener distintas políticas de confirmación:

1. Confirmar inmediatamente la recepción.
2. Piggybacking: Para hacer más eficiente el uso de la red. Espera a ver si la aplicación tiene algo para enviar en respuesta y aprovechar ese mensaje para confirmar la correcta recepción del anterior. Es decir confirmo por cada segmento o confirmo al final de varios segmentos recibidos. Ej: screen de los 4 segmentos recibidos.

-Para telnet y otros servicios se puede usar este flag para indicar que los datos deben entregarse inmediatamente. Lo recibe, lo confirma y lo entrega a la app.

Relacionados al establecimiento de la conexión:

- **RST:**

Lo envía un host cuando la conexión necesita restablecerse ya sea por rechazo de petición de conexión o porque uno de los dos extremos perdió el estado de la conexión cuántos bytes hay intercambiados. (servidor no acepta la conexión del cliente envía un segmento con el flag activado RST.)

Ej: Cuando hago telnet y recibo al toque la advertencia que no se pudo conectar es porque mandó un RST y no esperó un timeout

Cierre despropósito de la conexión.

- **SYN:**

Sincronizar.

Sincroniza los números de MTU/MSS máximos de cada host para determinar el menor de ambos y que no se necesite fragmentar con protocolo IP.

- **FIN:**

Para cortar conexiones entre cliente y servidor, lo envían ambos.

MSS Option:

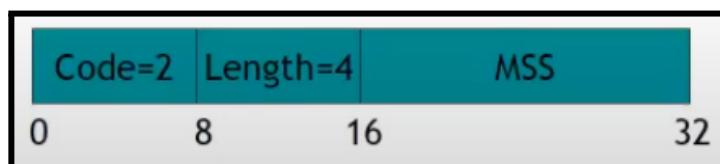
Maximum segment size.

Es declarada al establecimiento de la conexión, (Segmentos SYN). No puede modificarse durante el intercambio de segmentos. Es decir cada host envía al otro cuál es su MSS.

Determina el tamaño máximo del segmento de datos que es capaz de aceptar. Calcula a partir de la MTU de la interfaz cuál sería el tamaño.

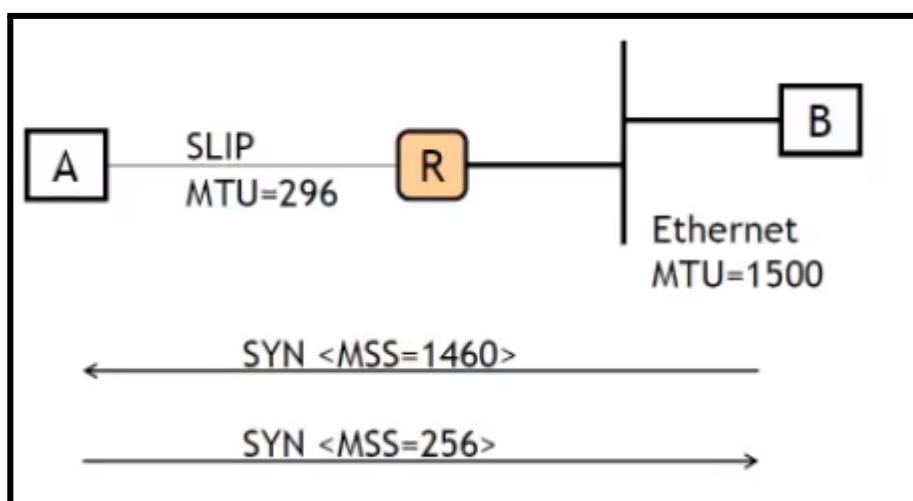
-Ejemplo con ethernet: Como ethernet tiene longitud máxima de 1500 bytes, el mensaje que es encapsulado en ethernet es más grande que 1500 bytes, IP empieza con el mecanismo de fragmentación que trae problemas de pérdida y las aplicaciones no desean. Si en el host origen se cual es el MTU del protocolo de capa 2, sé cuánto ocupa una cabecera IP y una cabecera TCP puedo calcular el tamaño máximo de lo que meta en el campo de datos de TCP. El MSS calcula a partir del MTU de la interfaz cuál sería ese tamaño.

Debe ser: MTU de la interfaz capa 2 - 40 bytes (20 bytes de la cabecera IP y 20 bytes de la cabecera TCP)



Ej: interfaz ethernet con MTU 1500 - 20 - 20 = 1460 MSS

Cada host envía al otro cuál es su MSS y usan el menor valor de ambos:



Establecimiento de la conexión - 3 handshake :

En estos pasos operan muchos mecanismos de seguridad como firewalls que permiten conexiones salientes. El firewall podría inspeccionar el tráfico y asegurar que los SYN vayan de adentro hacia afuera y no permitir al revés, los descarta. Chequea que los SYN+ACK tengan un SYN previo, etc.

Tiene 3 partes:

1. Primera parte:

Un cliente se conecta a un servidor a través de un mensaje de petición de conexión. Se le dice segmento SYN porque es un segmento vacío sin datos pero encendido con el flag SYN, además, el mensaje va a tener contenido en el campo número de secuencia (ISN) y uno en acknowledge pero éste es irrelevante porque el flag ACK está apagado.

El seq = X es el ISN (initial sequence number) que indica el primer valor de secuencia de los mensajes que envíe el host y que no debe ser 1 por prácticas de seguridad sino aleatorio, es el cero relativo. Donde se debe empezar a numerar.

2. Segunda parte:

El servidor envía un segmento SYN+ACK sin datos con el flag SYN y el flag ACK encendidos y contenido en acknowledge y seq. Confirmando la petición.

En Ack = X+1 indica que espera recibir del próximo mensaje su sequence number igual al siguiente del anterior indicado.

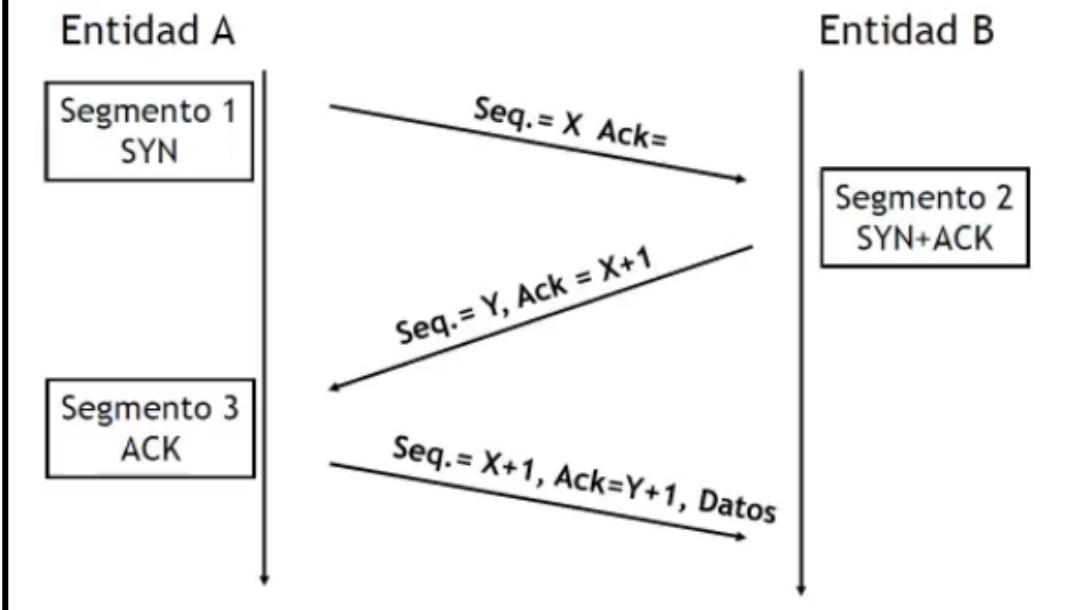
Envía también su propio ISN con valor Y (cualquier valor sin relación con X), es el cero relativo del servidor.

3. Tercera parte:

El cliente envía un segmento ACK. Puede o no llevar datos porque ya está establecida la conexión.

ACK = Y+1 indica el sequence number del servidor próximo que espera recibir y envía su sequence number asociados a los datos que pueden viajar o no también.

Three-way Handshake

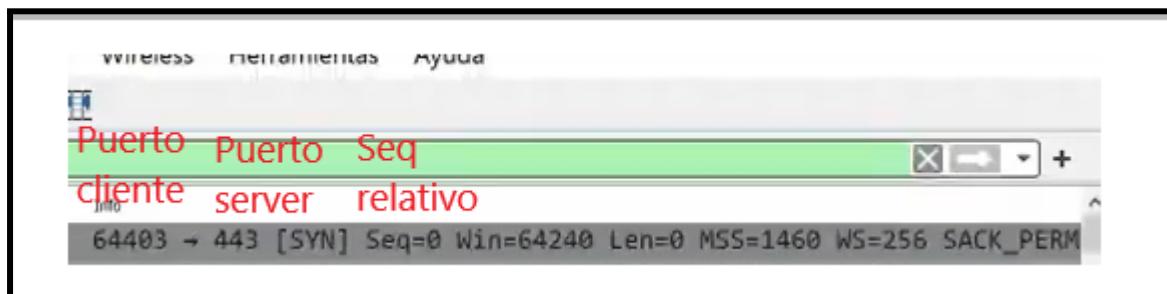


Ejemplos con tramas wireshark:

| Time | Source | Destination | Protocol | Info |
|--------------|---------------|---------------|----------|---|
| 235 1.648129 | 192.168.0.14 | 31.186.239.94 | TCP | 64403 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM |
| 247 1.907995 | 31.186.239.94 | 192.168.0.14 | TCP | 443 → 64403 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 |
| 248 1.908241 | 192.168.0.14 | 31.186.239.94 | TCP | 64403 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0 |

El seq = 0 y el Ack = 1 son relativos, si te parás sobre ellos en wireshark el número HEXA es otro valor.

El cliente envía el dato (hola) en otro mensaje pero podría haber estado en el ACK



| Protocolo | Info |
|-----------|---|
| TCP | 64403 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| TCP | 443 → 64403 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 |
| TCP | 64403 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0 |
| TLSv1.2 | Client Hello |

```

▼ Transmission Control Protocol, Src Port: 64403, Dst Port: 443, Seq: 0, Len: 0
  Source Port: 64403
  Destination Port: 443
  [Stream index: 1]
  [TCP Segment Len: 0]
  Sequence Number: 0      (relative sequence number)
  Sequence Number (raw): 530418923
  <
0000  6c 99 61 f7 cc ef 3c f0  11 34 c5 92 08 00 45 00  1-a...<..4...E
0010  00 34 ec 9b 40 00 80 06  3e 59 c0 a8 00 0e 1f ba  .4...@...>Y
0020  ef 5e fb 93 01 bb 1f 9d 8c eb 00 00 00 00 80 02  .^....[...].
0030  fa f0 fa 78 00 00 02 04  05 b4 01 03 03 08 01 01  ...x.....
0040  04 02

```

TLS da seguridad al protocolo TCP.

```

▼ Transmission Control Protocol, Src Port: 443, Dst Port: 64403, Seq: 0, Ack: 1
  Source Port: 443
  Destination Port: 64403
  [Stream index: 1]
  [TCP Segment Len: 0]
  Sequence Number: 0      (relative sequence number)
  Sequence Number (raw): 1130205933
  [Next Sequence Number: 1      (relative sequence number)]
  Acknowledgment Number: 1      (relative ack number)
  Acknowledgment number (raw): 530418924
  0110 .... = Header Length: 24 bytes (6)
  <
0000  3c f0 11 34 c5 92 6c 99  61 f7 cc ef 08 00 45 00  <..4..1..a....E
0010  00 2c 00 00 40 00 31 06  79 fd 1f ba ef 5e c0 a8  .,.@..1..y....^..
0020  00 0e 01 bb fb 93 43 5d  92 ed 1f 9d 8c eb 60 12  .....C]....[...].
0030  72 10 d6 13 00 00 02 04  05 b4 00 00 00

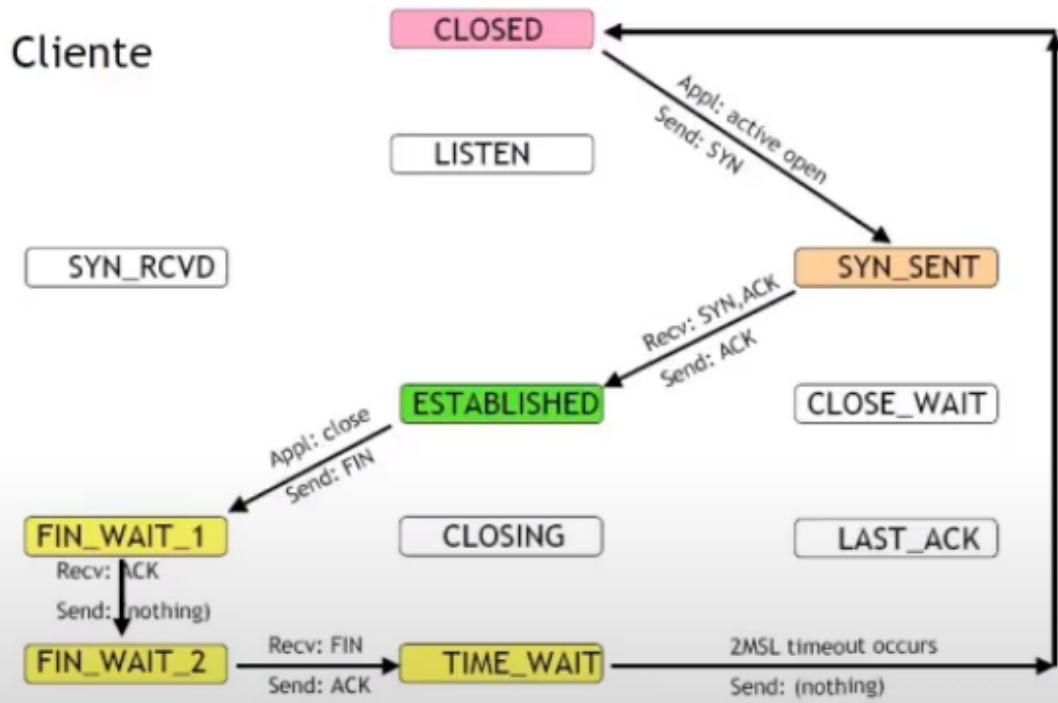
```

Diagrama de estados de una conexión:

1. Desde el punto de vista del cliente:

Cómo ve el progreso de la conexión.

Apertura y cierre normal



Estado CLOSED → Estado SYN_SENT:

Desde un estado inicial cerrado (ninguna conexión) la app hace una apertura activa (le dice a TCP que se conecte a un destino en tal puerto), TCP arma un SYN con el puerto destino y puerto origen, un número de secuencia inicial (NSI) y el flag SYN activo.

Estado SYN_SENT → Estado ESTABLISHED:

Espera el SYN+ACK y envía el ACK. Se establece la conexión. e intercambian los datos.

Estado ESTABLISHED → Estado FIN_WAIT_1:

La app considera que no tiene nada más que intercambiar. Lo hace el cliente o el servidor. El cliente cierra la mitad de la conexión y envía un FIN

Estado FIN_WAIT_1 → FIN_WAIT_2:

El cliente recibe un ACK del servidor para cortar la conexión.

FIN_WAIT_2 → TIME_WAIT:

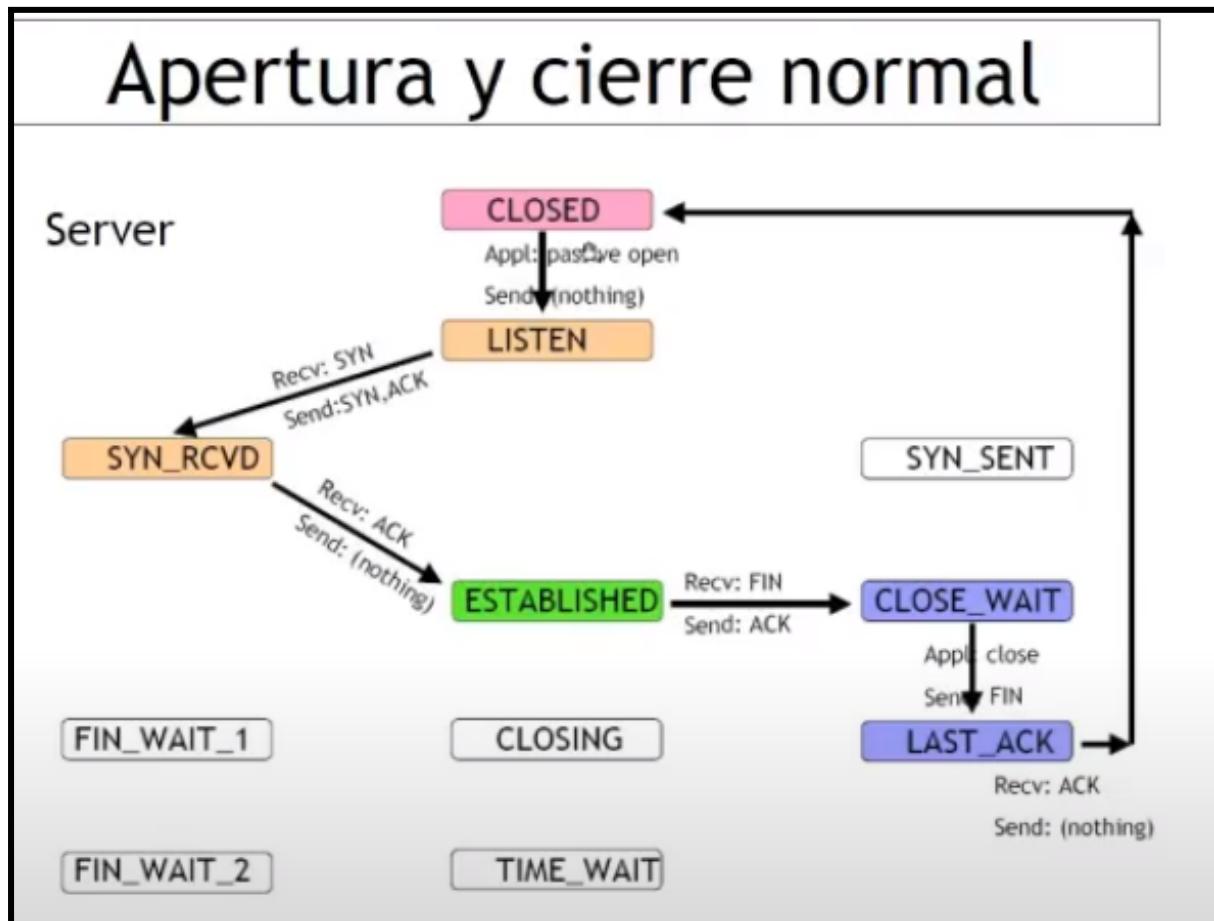
El cliente recibe un FIN del servidor y envía un ACK y se cierra la otra mitad del server.

Cierre de la conexión:

Se hace en 4 pasos (y no 3 como apertura), cierre de la mitad del cliente y cierre de la mitad del servidor. Cierre de las dos mitades por separado.

El cliente manda FIN, espera ACK, recibe ACK y espera cierre de servidor, servidor envía FIN y cliente responde ACK.

2. Desde el punto de vista del servidor:



Apertura pasiva:

Se activa TCP sobre un puerto cuando uno inicia el servidor y no mandando una petición al remoto como en la apertura del cliente. El servidor queda escuchando.

Estado CLOSED → Estado LISTEN:

El servidor queda escuchando a posibles peticiones.

Estado LISTEN → Estado SYN_RCVD:

Al recibir un SYN de un cliente envía un SYN ACK.

Estado SYN_RCVD → Estado ESTABLISHED:

Recibe un ACK del cliente.

Realiza el intercambio de datos hasta que el cliente le envíe un FIN.

Estado ESTABLISHED → Estado CLOSE_WAIT:

El cliente envía un FIN de la conexión que el servidor envía ACK de confirmación.

Estado CLOSE_WAIT → Estado LAST_ACK:

El servidor envía el FIN y el cliente retorna ACK de este último FIN enviado por el server.

Diferencia entre cliente y servidor:

El cliente para una conexión se abre y se cierra, y desde el punto de vista del servidor es la conexión de un cliente pero se queda escuchando por futuras conexiones de futuros clientes y sólo se desactiva si se desactiva el proceso que está escuchando.

Control de flujo:

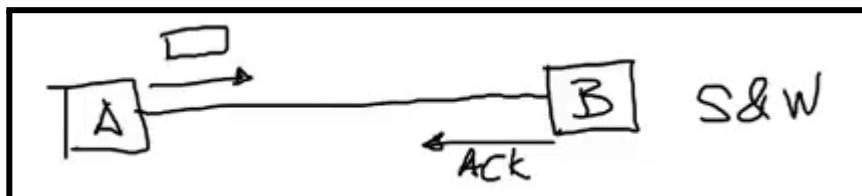
En la conexión de dos host, el host destino podría no tener la capacidad necesaria para procesar todos los mensajes que le envían, si el protocolo de capa par implementa una herramienta de control de flujo le da al host una forma de control de la cantidad de información que puede tomar.

Cómo se realiza el control de flujo en general:

Lo vimos en comunicaciones.

1. 1 Stop and Wait

Mientras B no confirma el ACK, A no puede enviarle más mensajes, así B controla el flujo.



Inconvenientes de Stop and Wait:

- Es ineficiente en el uso del enlace de datos. Se envía solo un mensaje a la vez con una conexión tipo half-duplex. O A envía y B confirma o viceversa.

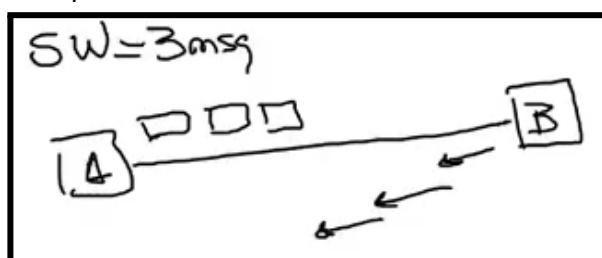
2. 2 Sliding Window - Ventana deslizante

Aumenta la cantidad de mensajes que A le puede enviar a B sin que B le confirme.

B confirma 1,2 y 3 y habilita a que A le envíe un cuarto mensaje. 3 enviados 3 confirmados y así.

Se parece más a full-duplex de enviar y recibir a la vez. Cuando A envía el 1, ya envía la confirmación y va enviando el 2, y así puedo enviar el 4 por el ACK anterior.

Lo usan la gran mayoría de protocolos. TCP tiene una variante de este mecanismo.



Control de flujo en TCP:

- El protocolo utiliza un mecanismo de la forma de “ventana deslizante” tal como HDLC
- A diferencia de HDLC separa la confirmación de datos recibidos del permiso para enviar más. (win + ack y en HDLC es N(S) con bits fijos, ventana fija y ese campo confirma y da permisos)
- Cada octeto de datos se considera que tiene un número de secuencia

-Esquema de Otorgamiento de Créditos:

Mecanismo donde uno de los extremos le indica al otro cuál es el crédito que le otorga, es decir, cuántos bytes le deja enviar antes de requerir una confirmación. Los bytes están asociados al tamaño del buffer asignado a la conexión.

(Es un poco distinto a sliding Window porque éste último es rígido con cantidad de mensajes y TCP con créditos dinámicos)

Este crédito inicial que los clientes van utilizando en cada mensaje, y con cada confirmación tengo un nuevo valor de ventana. Esta ventana se actualiza dinámicamente.

TCP confirma la recepción de datos con un mensaje (ACK) de la forma [A (acknowledge)= i, W (window) = j] donde:

- Se confirma la recepción de todos los octetos hasta $i-1$, se espera recibir i
- Se permite enviar una nueva ventana de datos ($W= j$ octetos). Esto es: desde i hasta $i+j-1$

Ej: En dibujo, indicas que el siguiente a recibir es 101, y $W = 10$ bytes entonces esperas hasta el 110

En este mecanismo está presente el campo window está en la cabecera TCP. Y en el establecimiento de la conexión se intercambia el crédito inicial. A le indica a B cuantos bytes le deja enviar y B le indica a A cuántos bytes le permite transmitir.

El servidor le indica su WIN, a cada mensaje del cliente se le resta la WIN:

```

48129 192.168.0.14      31.186.239.94    TCP      64403 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
97995 31.186.239.94     192.168.0.14     TCP      443 → 64403 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
08241 192.168.0.14      31.186.239.94    TCP      64403 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
08466 192.168.0.14      31.186.239.94    TLSv1.2 Client Hello
68290 31.186.239.94     192.168.0.14     TLSv1.2 Server Hello
88685 31.186.239.94     192.168.0.14     TCP      443 → 64403 [ACK] Seq=1461 Ack=518 Win=30016 Len=1460 TCP segment o
<

Sequence Number (raw): 1130205934
[Next Sequence Number: 1461 (relative sequence number)]
Acknowledgment Number: 518 (relative ack number)
Acknowledgment number (raw): 530419441
0101 .... = Header Length: 20 bytes (5)
> Flags: 0x010 (ACK)
Window: 30016
[Calculated window size: 30016]

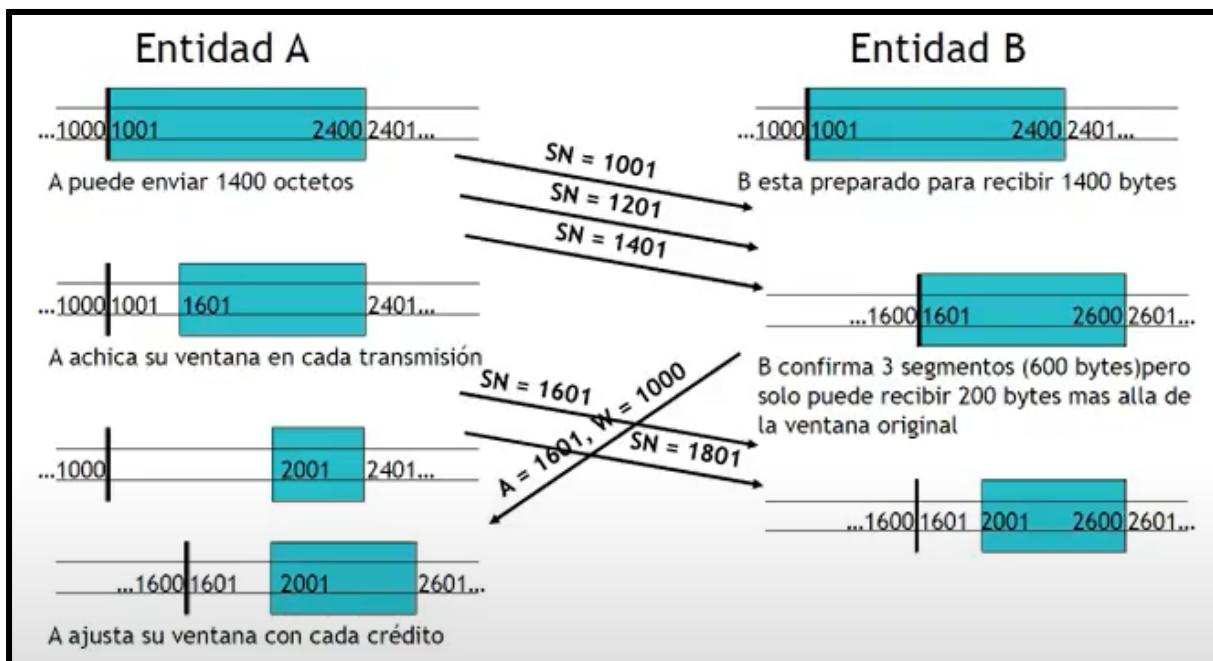
```

0030 75 40 c4 0f 00 00 16 03 03 00 62 02 00 00 5e 03

Ejemplo de la imagen:

El server le indica que puede mandar 29.200, el cliente envía 517, el server contesta que espera a partir del 518 y que a partir de ese momento la WIN cambió y es 30016 más. Cada extremo decide que hacer, podría haber ocurrido que el servidor decremente de los 29200 iniciales a los 517 que envió el cliente o le da más pero es administración de recursos del sistema operativo.

Ejemplo entre dos extremos y mensajes con créditos:



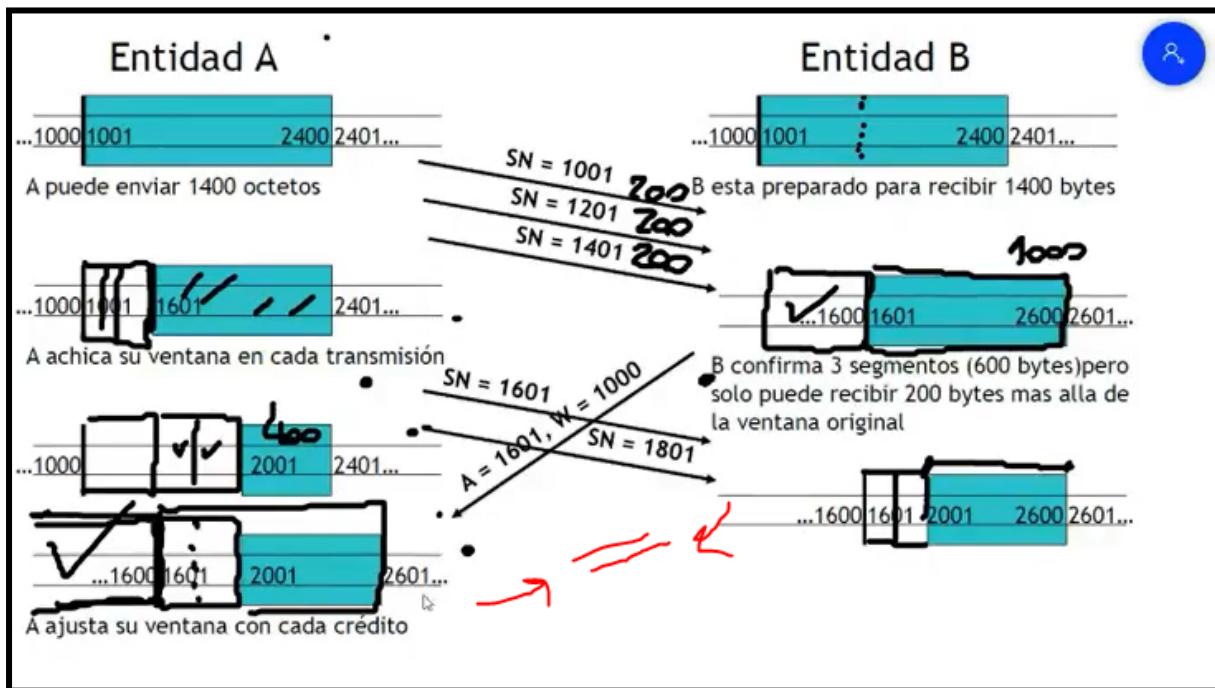
Cada “línea” horizontal es un punto en el tiempo.

Primero A envía 2 mensajes y le sobra crédito, B recibe y confirma los 3 (la confirmación llega después de que A vuelva a enviar 2 mensajes más con el crédito sobrante). Al recibir el ACK de B se modifica la ventana y se agranda pero de esos 1000 de crédito nuevos ya

consumí 400 de los dos mensajes enviados antes. En la última línea el estado del cliente es 600 bytes confirmados, 400 bytes enviados sin confirmar y 600 bytes más de crédito sobrantes para utilizar.

Este mismo mecanismo ocurre en ambos sentidos pero por simplicidad no sé muestra la ventana que A le da a B.

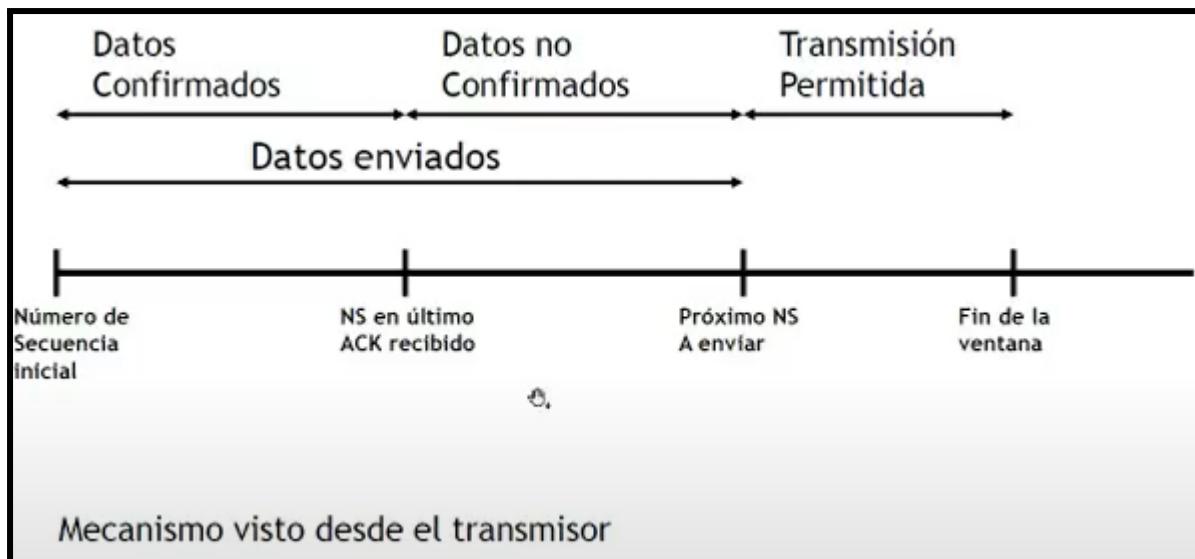
La línea vertical de A es el inicio de la ventana, lo anterior está confirmado por B -(se ve en gráfico de vista desde el transmisor).



Vista desde el transmisor:

Explicado arriba, tenés datos que ya enviaste y recibiste ACK (datos confirmados), tenés datos que están en tránsito sin ACK (datos no confirmados) y tenes un remanente de crédito hasta el fin de la ventana que es la transmisión permitida. Cada extremo lleva la cuenta de esta situación a través de las variables de estado número de secuencia inicial (INS), un último ACK recibido, puede haber bytes no confirmados (NS próximo - ACK último), etc.

De los dos lados A y B en ambos sentidos



Control de errores en TCP:

Estrategia de retransmisión.

- Como en los protocolos de capa 2, TCP incluye un mecanismo de control de errores: En OSI hay control de errores en capa 2 y capa 4, TCP es capa 4.
- No existe en TCP una confirmación de rechazo, tal como REJ o SREJ en HDLC.
- TCP se basa en la confirmación positiva de la recepción y retransmite cuando la confirmación no llega dentro de un período determinado RTO (retransmisión time out). Si llega la confirmación (ACK) no se retransmite.
- Método de corrección de errores en TCP es ARQ petición de retransmisión automática, el host que envía automáticamente va a retransmitir ante la ausencia de una confirmación. El tiempo que va a esperar es el timeout de retransmisión

Valor de RTO:

El valor del RTO indicado debe estar en el orden del Round trip time (Time en cmd) en milisegundos, es decir, lo que se tarda en ir y volver, que no va a ser lo mismo de tu pc al router, de tu pc a un servidor en argentina y de tu pc a los servidores de google. Puedo esperar más pero no mucho más, y no puedo esperar menos.

Los problemas que puede traer tener un RTO menor es que retransmite muchas veces antes de que me llegue el ACK. Y si es muy grande y el mensaje tuvo error voy a tardar bastante en darme cuenta que no llegó

```

C:\Users\fkoval>ping amazon.es
Pinging amazon.es [54.239.33.90] with 32 bytes of data:
Reply from 54.239.33.90: bytes=32 time=263ms TTL=218
Reply from 54.239.33.90: bytes=32 time=259ms TTL=218
Reply from 54.239.33.90: bytes=32 time=260ms TTL=218
Reply from 54.239.33.90: bytes=32 time=257ms TTL=218

Ping statistics for 54.239.33.90:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 257ms, Maximum = 263ms, Average = 259ms

C:\Users\fkoval>ping frba,utn.edu.ar
Ping request could not find host frba,utn.edu.ar. Please check

C:\Users\fkoval>ping frba.utn.edu.ar

Pinging frba.utn.edu.ar [181.30.38.43] with 32 bytes of data:
Reply from 181.30.38.43: bytes=32 time=16ms TTL=53
Reply from 181.30.38.43: bytes=32 time=12ms TTL=53
Reply from 181.30.38.43: bytes=32 time=16ms TTL=53
Reply from 181.30.38.43: bytes=32 time=11ms TTL=53

```

Manejo de la ventana de congestión:

Mecanismo relacionado con el control de la congestión. Al estar por alcanzar el umbral de congestión se ejecutan mecanismos para que no suceda.

Congestión de la red: Más información que lo que puede manejar la red, la carga excede la capacidad y se satura. Los mensajes se encolan en el buffer o pueden sobrepasar y son directamente descartados.

Descarte de un mensaje = retransmisión del mismo, la condición se agrava de manera exponencial.

Umbral de congestión: Antes de alcanzarlo la eficiencia en la utilización de un recurso es alta, al pasarse la eficiencia empieza a caer.

Al comienzo de la transmisión, no se tiene información acerca del estado de la red. Es necesario determinar cuántos segmentos pueden enviarse. Para ello se define la ventana de congestión **cnwd**.

Ventana permitida:

Mínimo entre la ventana de congestión y el crédito.

$$anwd = \min [cnwd, credit]$$

anwd =

anwd : Ventana permitida, en segmentos. Cantidad de segmentos que se pueden enviar ahora, sin esperar ACK.

cnwd : Ventana de congestión, en segmentos. Usada por TCP en el comienzo y durante períodos de congestión

credit : Cantidad de bytes permitidos por el destino, en segmentos (= Window / Tamaño de segmento)

Credit = Divido la cantidad de datos que tengo de crédito / tamaño máximo de segmento

Ej: En wireshark Win: 29200 / 1500 (ethernet) = 20 tramas

1. Comienzo lento - slow start:

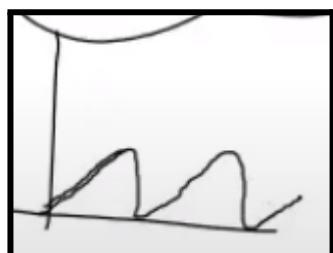
Cuando se inicia una nueva conexión, se inicializa cnwd = 1. Cada vez que se recibe una confirmación, se incrementa en 1.

Cuando un segmento se pierde (caduca RTO), cnwd vuelve a valor 1 y comienza nuevamente.

Básicamente en el ejemplo del dibujo donde A quiere mandar 4 mensajes a B, primero tiene una cnwd de 1 y un crédito de 30, así que envía 1 mensaje, recibe el ACK, su cnwd ahora vale 2 su crédito se mantiene en 30 se reduce o se incrementa y ahora puede mandar 2 mensajes en vez de 1. Luego su cnwd vale 4 por las dos confirmaciones previas y así sucesivamente, a menos que haya un error y el cnwd vuelve a 1.

Problema con el mecanismo slow start:

Si todos los nodos implementan este mecanismo, todos envían tráfico, se empieza a llenar el buffer y se desborda y se empiezan a rechazar paquetes (RTO) si se descarta un mensaje de cada uno de los nodos dejan de transmitir y vuelven a transmitir devuelta y así en ciclo.



2. Fast retransmit:

No tiene que ver con la solución a slow start, es otro mecanismo y no gestiona el control de la congestión sino es para manejo de la ventana.

Envias 2 segmentos por ejemplo y recibís 2 veces ACK del primero.

Cuando la fuente recibe un ACK duplicado significa:

- El segmento (2) fue demorado - pero finalmente llegará
- El segmento (2) se perdió - deberá retransmitirse.

En lugar de esperar a que caduque el RTO, si se reciben 3 ACK duplicados, se retransmite el segmento perdido.

Transmito el segmento A, luego el segmento B, recibo 3 ACK para el segmento A, en el momento que recibo la tercera ACK retransmito B sin importarme si caducó el RTO o no.

Esta mejora del mecanismo analiza que si recibiste 3 ACK duplicados es porque la red no está congestionada porque fluyen los paquetes sino que el segmento B se perdió y debo retransmitir.

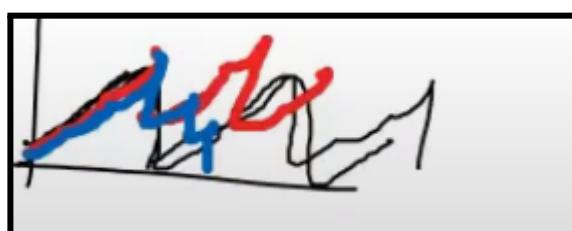
3. Fast recovery:

Viene a solucionar los problemas del slow start

Esta variante permite al transmisor evitar volver al slow-start en caso de perderse un segmento. (de forma esporádica y no por congestión de red).

Cuando se recibe el 3er ACK duplicado, se setea cwnd = cwnd / 2.

Se setea la ventana de congestión a la mitad de su valor y no a 1, porque puede ser un paquete que se perdió de manera esporádica y puedo retomar mi ritmo de transmisión más rápido porque la red no está congestionada. Si la red efectivamente está congestionada van a bajar progresivamente y aliviar la congestión sin problema.



Rojo: Con pérdida de paquete pero sin congestión.

Azul: Con pérdida de paquete por congestión.

Well-known ports:

Los puertos del servidor deben ser bien conocidos así los clientes se conectan al que escucha.

Puertos bien conocidos del servidor: Definidos del 1 al 1024.

Un cliente usa un puerto alto > 1024.

Estos valores están agotados hace años, cualquier protocolo nuevo tiene que escuchar en un puerto mayor.

| Port | Servicio |
|------|-----------------|
| 20 | FTP-Data |
| 53 | DNS |
| 21 | FTP - Command |
| 23 | Telnet |
| 80 | HTTP |
| 110 | POP - Version 3 |
| 25 | SMTP |
| 1720 | H.323 |

UDP - User Datagram Protocol

RFC 768

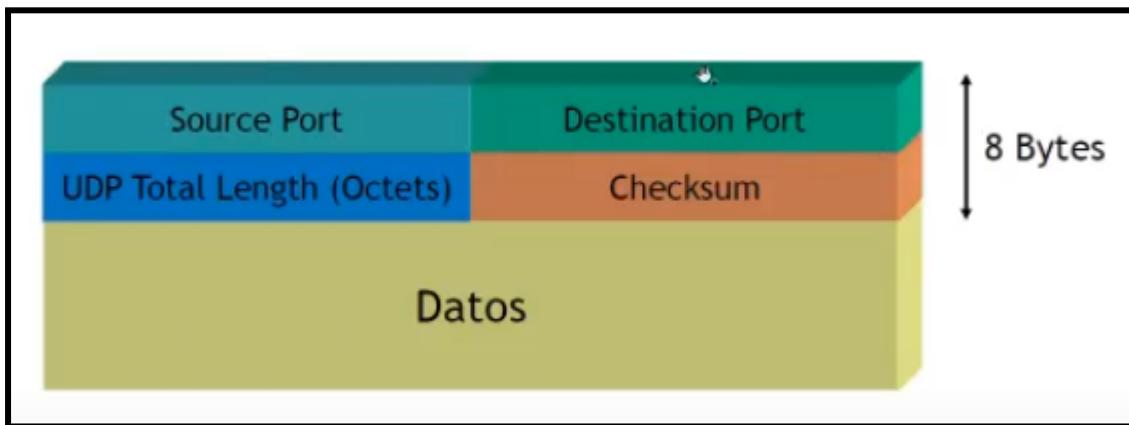
Servicio de datagramas no orientado a la conexión

Características:

- Brinda servicios similares a IP
 - No orientado a la conexión, no mantiene un estado.
 - No confiable, no envía notificaciones en caso de descartes, no reordena: Porque no tiene un número de secuencia ni orden como TCP
 - No realiza control de flujo: Porque no tiene un campo ventana como TCP
- UDP agrega (comparado con IP):
 - Puerto origen y destino para identificar el servicio
 - Checksum de la parte de datos, opcional. IP sólo tiene checksum de la cabecera. (control de errores)
- Es el preferido para servicios como:
 - Procesos simples de petición/respuesta (aplicaciones no críticas, sin necesidad de control de errores/flujo).
Ej: DNS que corre sobre UDP, el cliente pide la dirección IP de una página y DNS le responde.
 - Ej: NTP, network time protocol donde el cliente pregunta al servidor que hora es.
 - Multicast y broadcast
 - Streaming de audio y video de forma eficiente: Porque son procesos sensibles al retardo e isócronos, uso UDP que no hace ni chequeo ni retransmisión ni nada. Ej: Enviar un cuadro de video a todos, lo envío solo

una vez y no tiene sentido establecer una conexión con cada espectador y controlar si lo recibió, si retransmito, etc.

Formato de UDP:



1. Source y destination port = mismo significado que en TCP. identifica los procesos cliente y servidor
2. UDP total length (octetos).
3. Checksum: Es opcional.

Cómo funciona:

El sistema destino recibe el datagrama. Verifica el puerto destino con los puertos activos en ese momento.

- Si no coincide, envía un ICMP “destino inalcanzable” (puerto no vinculado a ningún servicio).
- Si coincide y hay lugar en el buffer lo encola.
- Si no hay lugar lo descarta. No envía mensaje de error

Ventajas:

ágil, rápido y sin sobrecarga para procesos que se ajustan a estas características.

Ejemplos de well-known ports de tipo UDP:

| Port | Servicio |
|------|---|
| 7 | ECHO |
| 53 | DNS |
| 69 | Trivial File Transfer Protocol (TFTP) |
| 123 | Network Time Protocol (NTP) |
| 161 | Simple Network Management Protocol (SNMP) |

HDLC - High-level digital link control

- Protocolo de capa 2 orientado a la conexión
- Protocolo orientado al bit (todo lo que vimos en la materia son de este tipo).
- Se usa en entornos punto a punto nada más. Ej: routers cisco con enlace serial
- Permite una transmisión “transparente” (independiente del código)
- Orientado a la conexión
- Formato único de trama
- Confirmación por ventana deslizante (más sencillo que TCP).
- Protocolo de WAN
- Diseñado para operar en configuración equilibrada y la que no (muy antigua).
- Utiliza un “flag” para la delimitación de comienzo y fin de tramas: “01111110” (7E)
Usa ese flag al comienzo, y al final.

Inserción de ceros:

Mecanismo que garantiza la transparencia del protocolo.

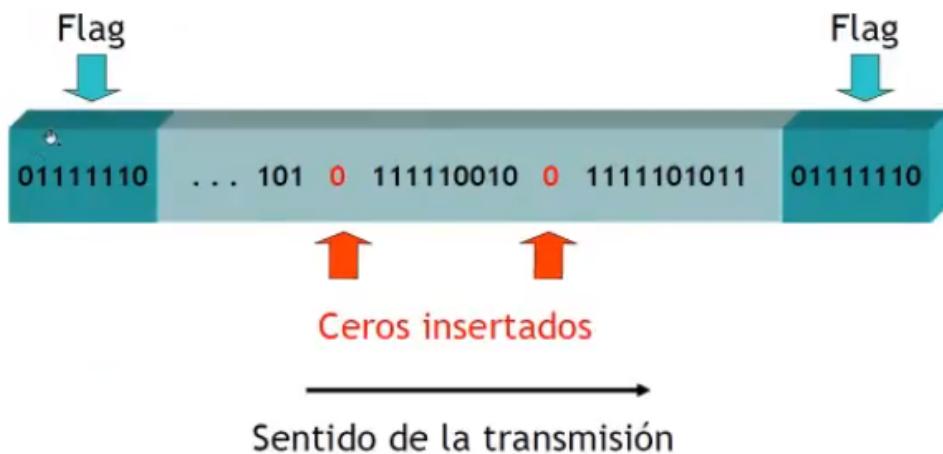
Consiste en asegurar que no existirá en el campo de datos una secuencia “01111110” (el flag) porque sino el receptor al leer esto truncará el mensaje pensando que llegó al fin y no es así.

Para realizarlo, inmediatamente después de la aparición de la secuencia 11111 (5 unos), se inserta un 0 sin importar qué bit sigue a continuación.

El receptor es el encargado de “retirar” ese 0, luego de recibir una secuencia 11111

Cuando el transmisor recibe la PDU de la capa superior, la encapsula en la trama de HDLC, la parsea (escanea los bits y hace lo de arriba) y se asegura de que no aparezca esta secuencia. Se pone de un lado y se quita del otro con parseo.

Mecanismo de Inserción de Ceros



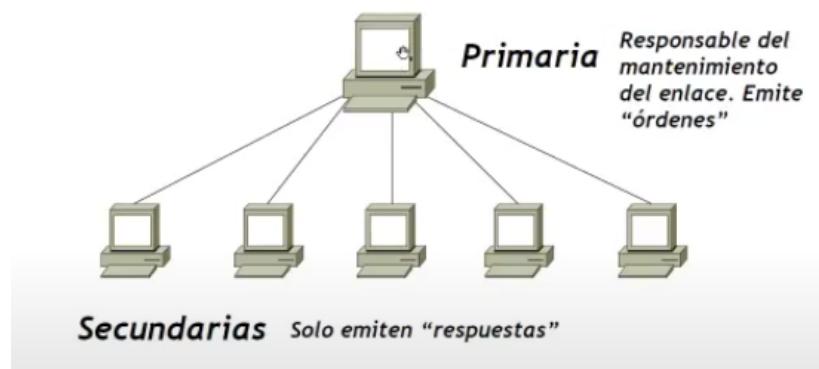
Estaciones:

Pueden ser:

- Primarias:
Controlan el enlace de datos
Transmiten “órdenes” a las estaciones secundarias y reciben “respuestas” de éstas.
- Secundarias:
Actúan como esclavas respondiendo a las órdenes.
No tienen responsabilidad en el mantenimiento del enlace.
- Combinadas:
Transmiten órdenes y respuestas, reciben órdenes y respuestas.
Mantienen sesiones con otras estaciones combinadas.

Configuración no-equilibrada:

Configuración No-equilibrada:

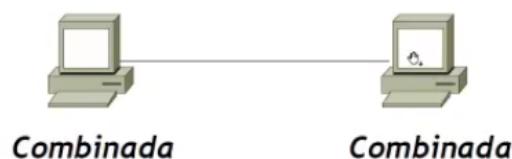


Cuando se crea el HDLC era muy popular el master - slave con primarias y secundarias
Sólo en los bancos hoy día, una estación procesador (host) y terminales remotas que no procesan.

Configuración equilibrada:

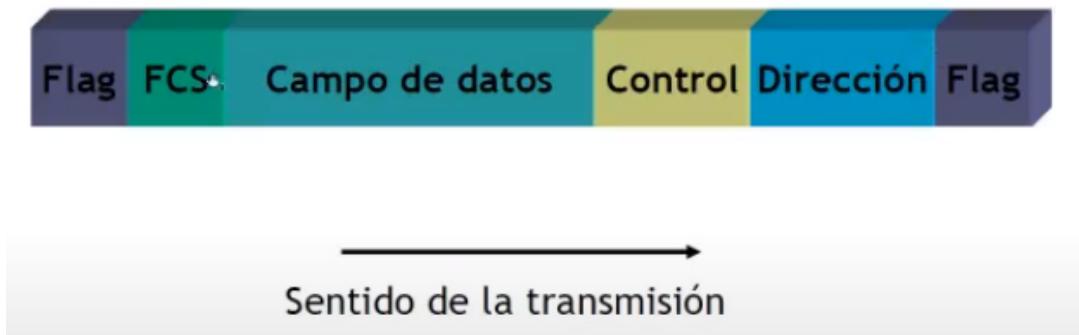
Cumplen el rol de primaria y secundaria a la vez.

Configuración Equilibrada:



Transmiten “órdenes” y “respuestas”.
Mantienen sesión con otras estaciones
Combinadas

Formato de trama:



1. 1 FLAGS:

1 byte, corresponde a la secuencia 01111110 (7E)

2. 2 FCS:

Es un CRC para la detección de errores. 16 o 32 bits

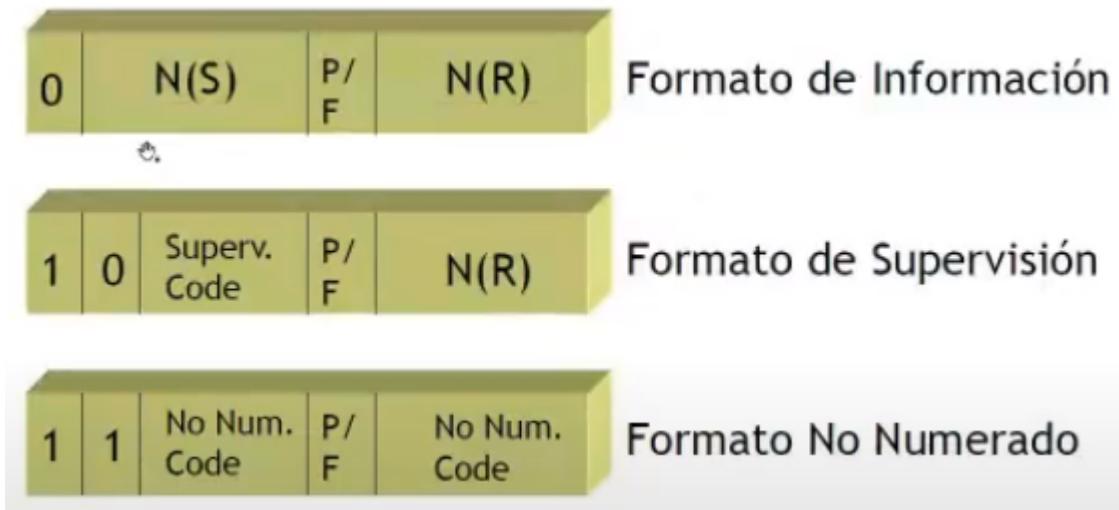
3. 3 Campo de datos:

Longitud variable, transparente e independiente del código.

4. 4 Control:

Tiene formato variable. 8 bits para simple o 16

Sobre él se implementan todos los mecanismos de control de flujo y control del enlace.



5. 5 Dirección:

8 bits

Es útil para la configuración no-equilibrada porque hay varios secundarios.

Tiene un valor predeterminado irrelevante.

Identifica la estación secundaria que ha transmitido o va a recibir la trama.
No se utiliza en enlaces punto a punto. No hace falta indicar la dirección.

3 tipos de tramas en HDLC:

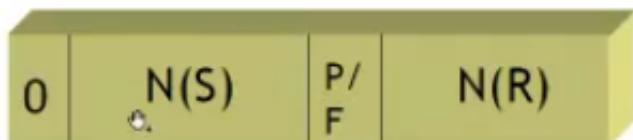
Se diferencian entre sí por el contenido del campo de control.

1. De información:

Si el primer bit es 0
Luego de utilizar la trama no numerada.
-N(S): Número de trama enviada
-N(R): Número de trama que espero recibir.
N(S) y N(R) se usan para la ventana deslizante (Campo sequence number y win en TCP) pero de una forma mucho más simple.

Sirven para:

- Envío de información
- Aceptación de tramas
- Información de trama enviada.



2. De supervisión:

Si el primer bit es 1 y el segundo 0
Eventualmente se intercambian algunas de estas tramas.
2 bits para código de supervisión, 4 bits de N(R) y 1 para P/F
Tienen N(R): Sirve para confirmar la recepción de tramas también.
Asisten en el control del enlace:

- Aceptación de tramas
- Solicitud de transmisión de tramas
- Suspensión temporal de la transmisión: Es control de flujo



Ejemplos:

- **RR** Receptor preparado
- **RNR** Receptor no preparado
- **REJ** Rechazo simple
- **SREJ** Rechazo selectivo

3. No numeradas:

Si primer y segundo bit son 1.

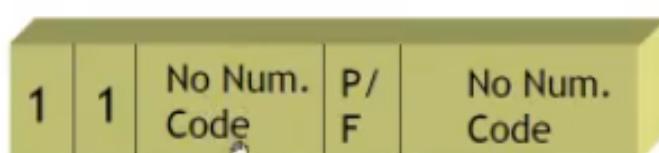
5 bits para codificar 32 comandos y 32 respuestas.

1 bit P/F: Pol final

Se llaman así porque no tienen N(S) ni N(R) no secuenciadas

Se utiliza para

- establecer la conexión y desconexión del enlace
- Control del enlace



Ejemplos:

- **SABM (C)** (Fijar modo asíncrono/balanceado)
- **UA (R)** Confirmación no-numerada
- **DISC (C)** Desconectar
- **DM (R)** Modo desconectado
- **FRMR (R)** Rechazo de trama

(C) = Comando

(R) = Respuesta

Disc: Desconectar la conexión.

DM: Para rechazar una petición de conexión.

Un host puede responder con un UA o un DM a una petición SABM.

Bit P/F (Poll/Final):

Presente en los 3 tipos de tramas HDLC.

-La estación primaria utiliza el bit P para solicitar una respuesta de estado a la estación secundaria. Está haciendo un sondeo, forzando una respuesta (hacer un poll), consultar.

-La estación secundaria responde al bit P con una trama de información o de supervisión en el bit F. La estación secundaria responde a la consulta de la primaria.

-El bit F indica también final de la transmisión de la estación secundaria, en NRM.

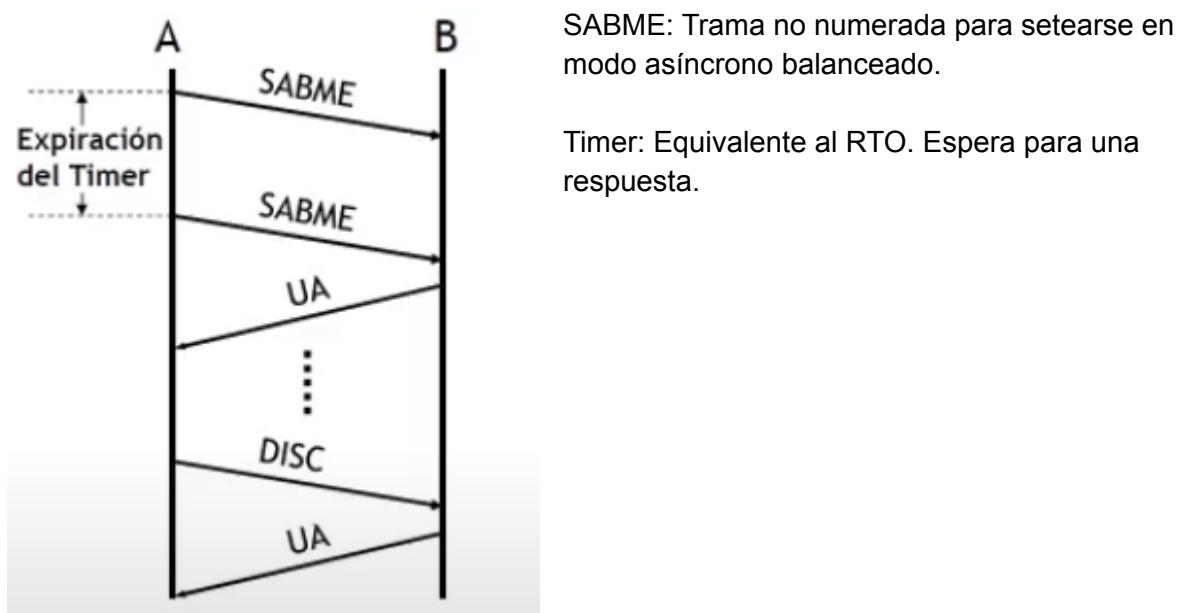
Iniciación del enlace:

- La solicita cualquiera de los dos extremos.
- Se avisa al otro extremo sobre la solicitud de inicialización.
- Se envía una trama no numerada.
- Se especifica cuál de los tres modos (NRM, ABM asíncrono balanceado el único que nos interesa en la configuración equilibrada, ARM) se está solicitando.
- Se especifica si se van a utilizar números de secuencia de 3 o 7 bits. (dependiendo si el campo de control es un byte o dos, y afecta a los campos N(S) y N(R)).
- Si el otro extremo acepta la solicitud envía un UA, caso contrario envía un DM

Desconexión del enlace:

- La puede iniciar cualquiera de los dos extremos.
- Se envía una trama de desconexión DISC.
- El extremo receptor acepta devolviendo un UA.

Ejemplo de conexión en configuración equilibrada:

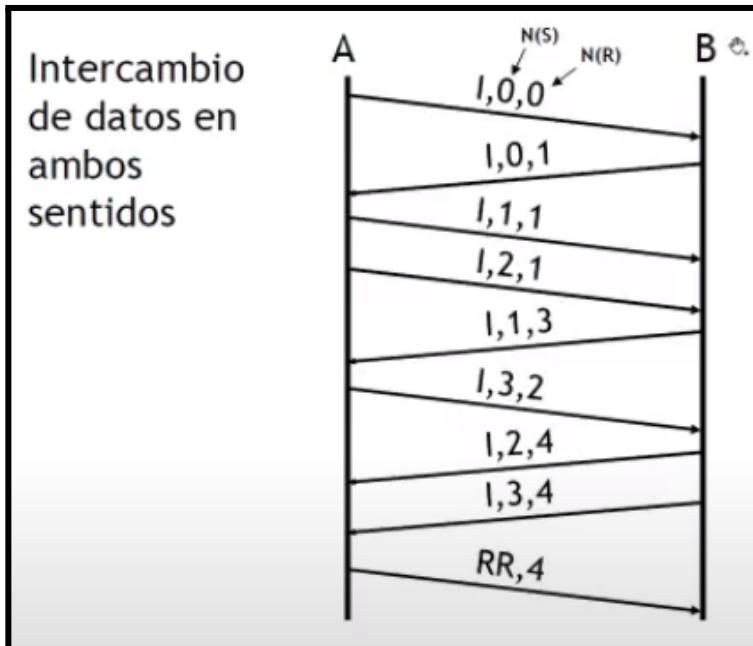


Transferencia de datos:

Aceptada la solicitud de inicialización, comienza la etapa de transferencia.

- Pueden transmitirse tramas de información, comenzando con el número de secuencia 0 (distinto a TCP)
- Con los campos N(S) y N(R) se llevará a cabo el control de flujo y de errores (va confirmando los paquetes que recibió): Se usa FCS como mecanismo de detección de errores, el N(R) sirve como mecanismo de confirmación porque afirma la correcta recepción de los datos y que te permite enviarle al mismo host otro mensaje (control de flujo) que es una ventana siempre de 8 a partir de la última trama.
- La secuencia de tramas se numerará secuencialmente módulo 8 o módulo 128 utilizando el campo N(S): Depende los bytes del campo de control.
- El campo N(R) se utiliza para la confirmación de las tramas recibidas.
- Las tramas de supervisión también se usan para el control de flujo y errores.
 - RR: Confirma la trama de información recibida, indicando la próxima trama de información que se espera recibir.
 - RNR: Confirma la trama de información recibida y solicita la suspensión momentánea de la transmisión.
 - REJ: Inicia el procedimiento go-back N (volver atrás y retransmitir todos a partir del último N(R)). Sigue la retransmisión de las tramas posteriores a N(R)
 - SREJ: Se utiliza para solicitar la retransmisión de una única trama.

Ejemplo 1: Con piggybacking



Primer mensaje, de tipo Información, N(S) indica que manda la trama 0 y N(R) indica que espera de B la trama 0.

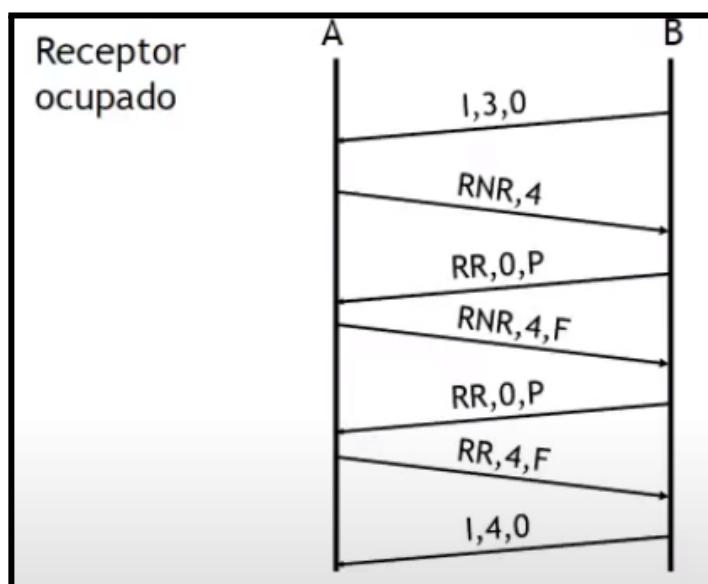
B envío la trama 3, para que B no la retransmita, A como no tiene una trama de datos/información que enviar para aprovechar y enviarle la confirmación (**Piggybacking**), usa una trama de supervisión RR con la correcta recepción de la trama que envió B. Es más sencillo que en TCP.

Ejemplo 2: Receptor ocupado (control de flujo)

Cuando A recibe el mensaje de B decide no aceptarlo porque está ocupado y sin recursos para procesar ese pedido. Detiene temporalmente la conexión enviando una trama de supervisión RNR (not ready) donde confirma el recibimiento del mensaje pero pide que se detenga.

Los mensajes que envía B vienen de PDUs de capas superiores, no lo puede “controlar”, entonces pasado un intervalo de tiempo B envía una trama de supervisión RR con P (poll) para sondear a A y pedir que le responda.

A le responde con otro RNR y le dice que sigue ocupado con F (final). Pasado un tiempo B vuelve a sondear, y ahí A le responde con RR ready y B le envía el mensaje que quería enviar retomando el enlace.



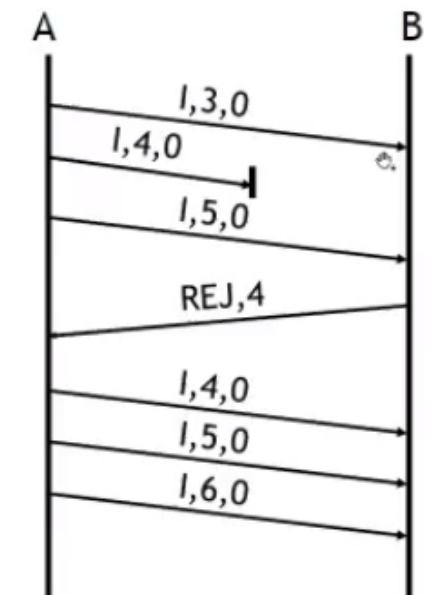
RNR, 4: Recibí correctamente la trama 4, no envíes más mensajes.

Ejemplo 3: Recuperación de un rechazo

A envía una trama a B que se pierde, error en la transmisión entonces B la ignora/descarta.

Se produce una condición de error por secuencia de números incorrecta (Recibe la 3 y la 5 pero falta la 4).

Recuperación de un rechazo



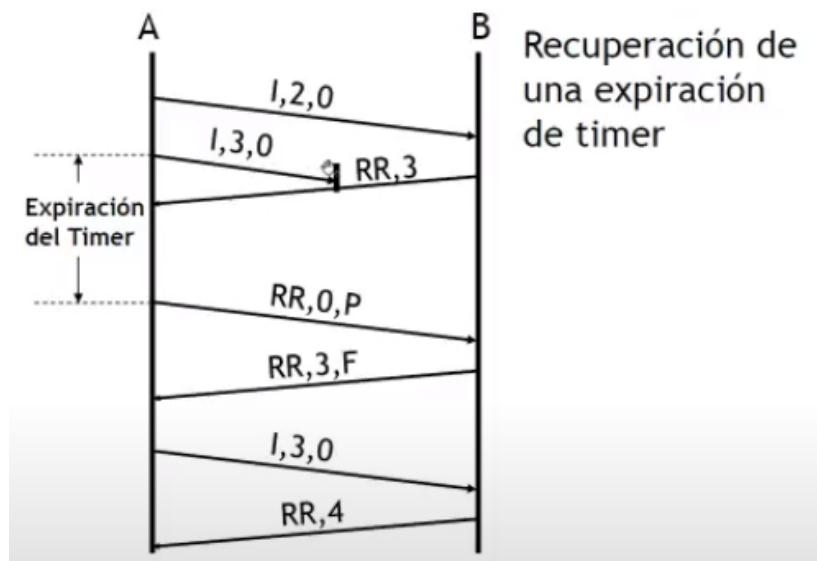
B envía una trama de supervisión REJ indicando que esperaba la 4. A reenvía la 4 y la 5 y luego lo nuevo.

TCP esto no existe porque si no llega ACK de confirmación se retransmite directamente todo, si llegó la 5 y no la 4 retransmite ambas igual. HDLC si tiene idea de rechazo.

Otra alternativa: Rechazo selectivo poco implementado porque requiere mayor procesamiento y sincronización entre ambos extremos. Posibilita que en el mensaje REJ de enviar una trama que diga "me falta la 4 pero tengo la 5" así no se retransmite todo.

Ejemplo 4: Recuperación de una expiración del timer

La trama 3 se pierde, B envía trama de supervisión RR porque no tiene trama de información para enviar. A no sabe que la 3 no llegó bien todavía, espera un timer y como no llega la confirmación de B envía un sondeo RR con bit P (poll) encendido. Y B le responde que está preparado con bit F entonces A retransmite el mensaje 3.



Derivados del HDLC:

Surgieron muchos protocolos a partir de HDLC pero ya no se utilizan.
El profe salteó la PPT de esto, no sé si es relevante.

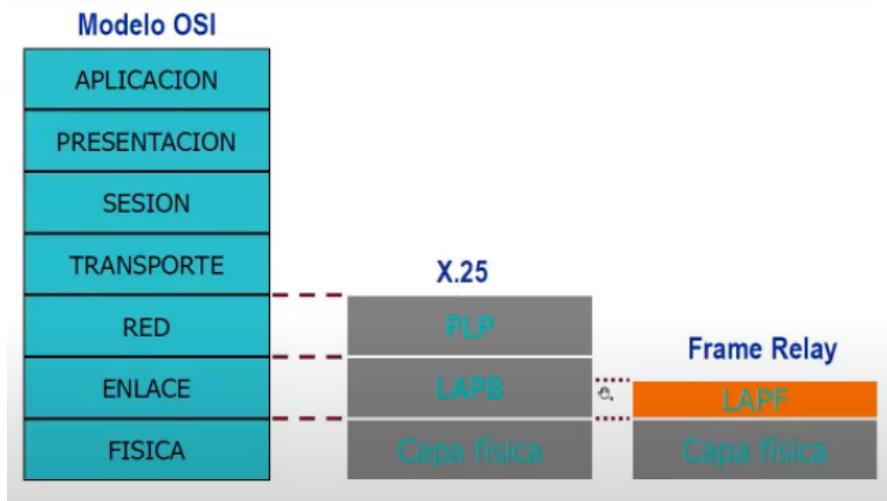
Frame Relay

- Protocolo de capa 2 orientado a la conexión
- Nacido para ser utilizado sobre el canal D en redes ISDN (LAP-D) algo histórico.
- Derivado del HDLC con características adicionales.
 - Más liviano, sin ventanas.
 - Ya no hay control de flujo y errores.
- No provee calidad de servicio ni recuperación de errores.

- Utiliza “circuitos virtuales” para interconectar sitios remotos. Generalmente permanentes (PVCs)
- Implementado sobre velocidades de n*64 (redes telefónicas de 64k) hasta 34 Mbps.
- Permite multiplexar en capa 2.
- Ventajas vs conectar físico que tiene límites más bajos y es más lento.
- Cayó en desuso pero otras tecnologías usan esto.
- Protocolo de WAN

Dónde se ubica:

LAPB = implementación de HDLC quedándose sólo con la configuración equilibrada.

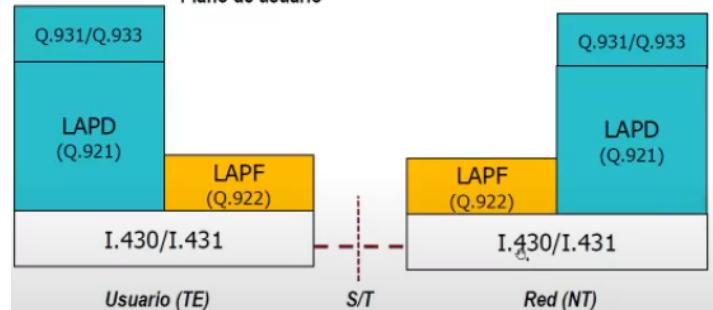


Arquitectura de Frame Relay:

No tan relevante hoy día.

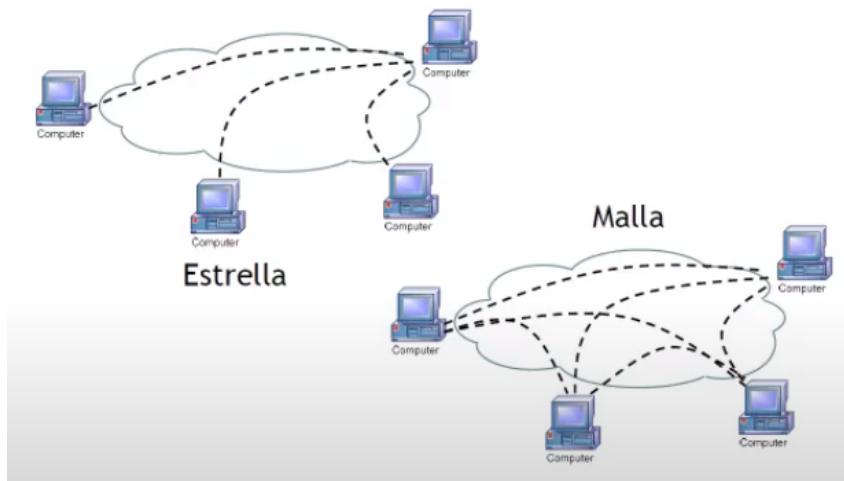
Plano de control

Plano de usuario



Topologías:

Lo relevante del frame relay.



Estrella: Funciona bien cuando el flujo va hacia el nodo central.

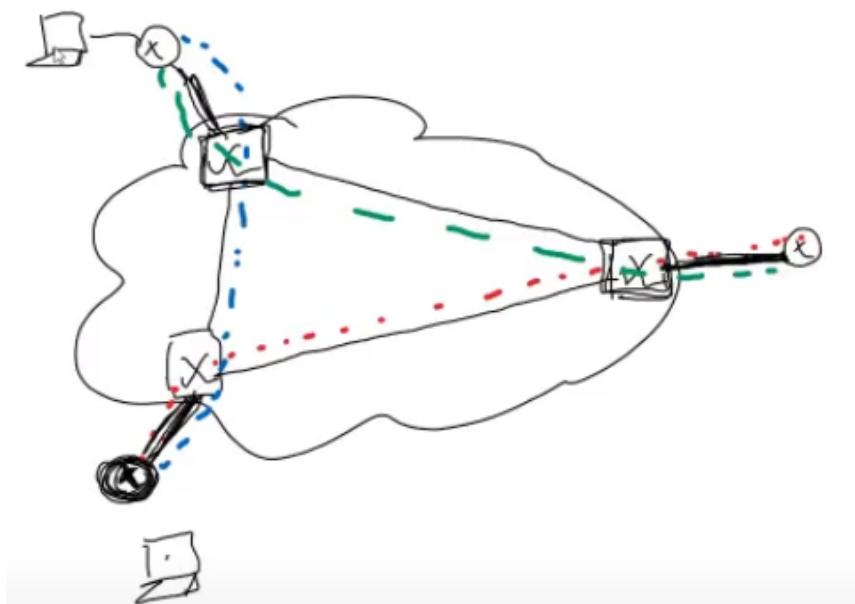
Malla:

- Surge cuando tengo que interconectar sucursales. Full merge, todos con todos.
- Con circuitos físicos no tiene mucho sentido, es mejor virtual.
- Circuito físico de cada nodo terminal al router o punto de salida y el resto es lógico.

Circuitos virtuales permanentes:

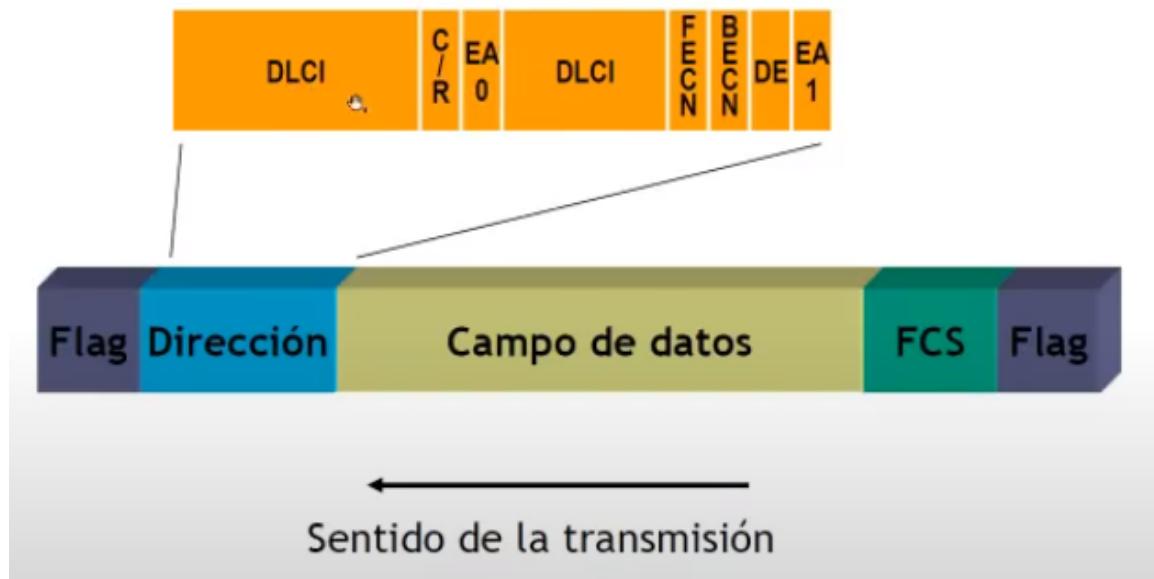
HDLC no sirve para esto porque es para conectar 2 dispositivos, uno con otro.

Frame relay te permite “multiplexar” comunicaciones, con un mismo acceso físico lo comparto multiplexando paquetes por división del tiempo para comunicar varios destinos.



Azul y rojo circuitos virtuales

Formato de trama:



1. Flag:

Lo mismo que en HDLC.

1 byte, corresponde a la secuencia 01111110 (7E)

2. Dirección:

Longitud mínima de 2 bytes

a. *DLCI:*

Data link channel indicator, campo de la cabecera de la trama para que el equipo terminal que debe determinar a qué circuito virtual quiere conectarse para comunicarse (canal rojo o azul) haga referencia a ese canal.

Cada canal tiene un identificador.

Permite $2^{10} = 1024$ DLCI distintos.

Tiene significado local y es configurado por el proveedor.

b. *EA:*

Extended address: Indica fin de cabecera. 1 para fin, 0 para indicar que continúa.

c. *C/R:*

Comando/respuesta. No importa.

d. *DE:*

Discard eligibility, elegible para descarte si se pasa del límite garantizado

e. *FECN:*

notificación explícita de congestión hacia adelante

f. *BECN:*

Notificación explícita de congestión hacia atrás

3. Campo de datos:

longitud variable.

Máximo 4096, generalmente se usan 1600 bytes.

4. FCS:

CRC-16

PVC - Circuito virtual permanente:

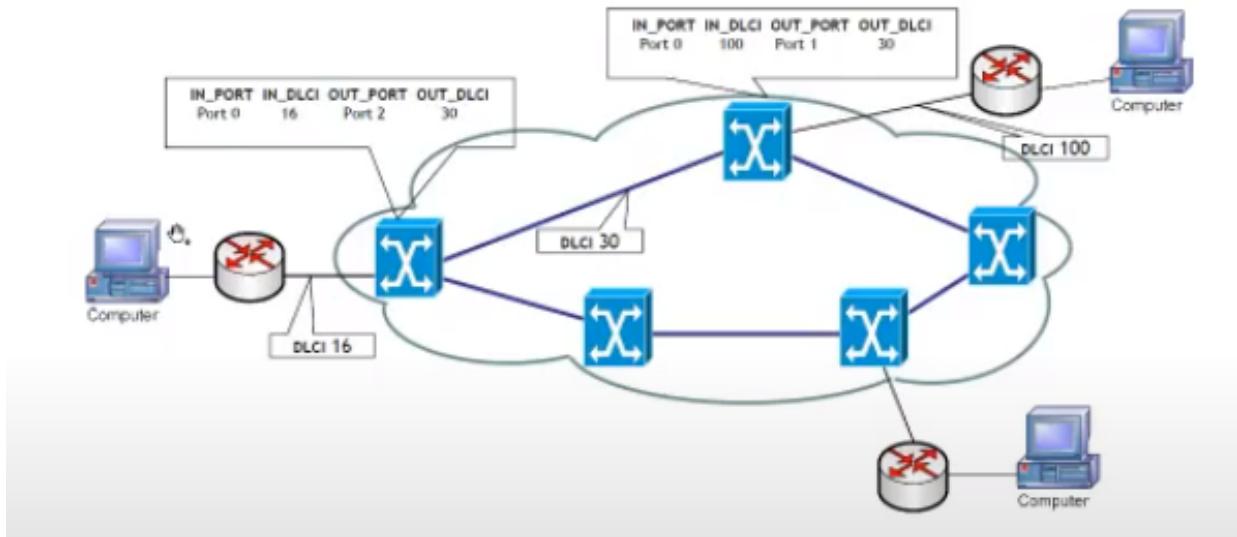
Es la concatenación de DLCIs para todas las sucursales de una red que configura el proveedor.

Rojo: Router

Azul: Nodo de provisión de red que configura lo que entra con el DLCI y sale por otro DLCI

El proveedor configura que DLCI tiene cada sucursal, tiene significado local (entre la estación y el punto de salida de la sucursal y el puerto o entre dos nodos de provisión de red y puertos).

Si se quisiera configurar otra sucursal como en el dibujo del profesor, debe configurarse otro DLCI para la sucursal de abajo, configurar distintos DLCIs para los nodos intermedios y configurar otro DLCI además del 16 (ej: DLCI 17) entre el mismo router y nodo del DLCI 16. Ej: El router de la izquierda va a tener 2 DLCIs, el 16 para ir hacia arriba y el 17 para ir hacia abajo.



Gestión del tráfico:



El tráfico en ráfagas no se ajusta bien a un ancho de banda fijo, tarda más tiempo. Lo que tenía picos cuando pase por el “caño” se va achatar y estirar. Esta ráfaga se ajustaría mejor a un servicio que deje transmitir picos, caños más grandes y para extremos distantes es muy caro, por eso conviene frame relay.

Algunas definiciones:

TC - Committed rate measurement interval:

Intervalo de tiempo durante el cual se mide la tasa de transmisión
Intervalo de medición.

BC - Committed burst size:

Cantidad máxima de bits que la red garantiza su entrega, durante TC bajo condiciones normales.
Tamaño de la ráfaga comprometida.

CIR - Committed information Rate:

Tasa de información comprometida.
Tasa de transmisión, en bits por segundo, que la red garantiza transmitir, bajo condiciones normales.
Cuánto quiero que me permitan transmitir sobre este enlace (ejemplo enlace rojo en dibujo), lo mínimo que yo tolero en momentos en que la red tiene mucho uso o está congestionada.
 $CIR = BC/TC$.

BE - Extended burst size:

Cantidad máxima de bits por encima del CIR, que la red intentará entregar, durante TC.
Es la ráfaga en exceso.

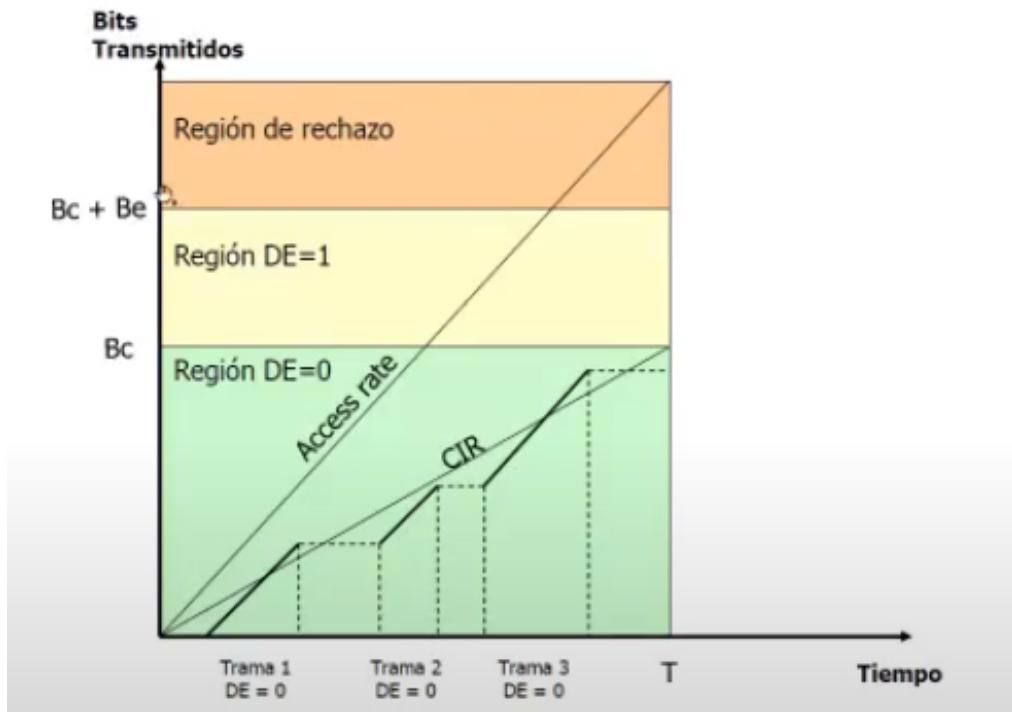
EIR - Extended information Rate:

Generalmente calculado como BE/TC

El access rate:

Es el ancho de banda de la última milla, la velocidad de transmisión del medio físico/cable o aire, la parte conectada físicamente entre la estación y el punto de acceso a la red.
Es el máximo que podés usar. Aunque físicamente haya más por el proveedor.

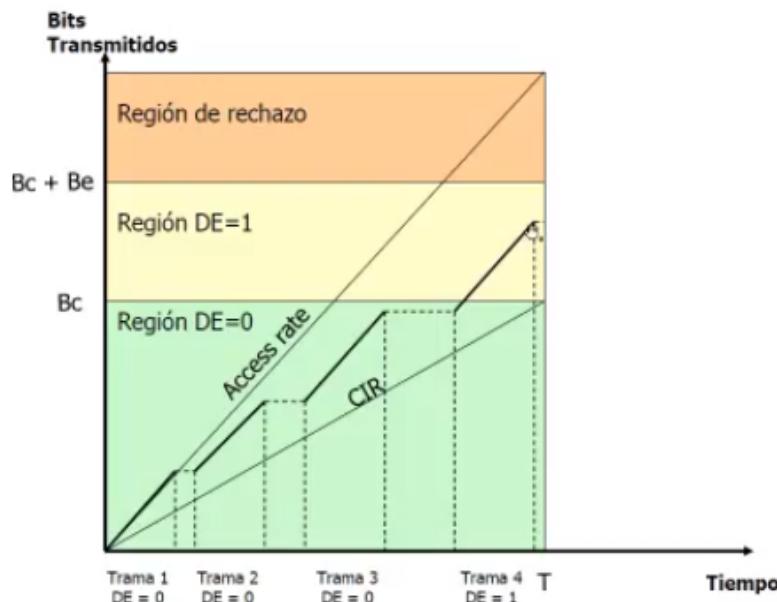
Ejemplo 1: tramas garantizadas:



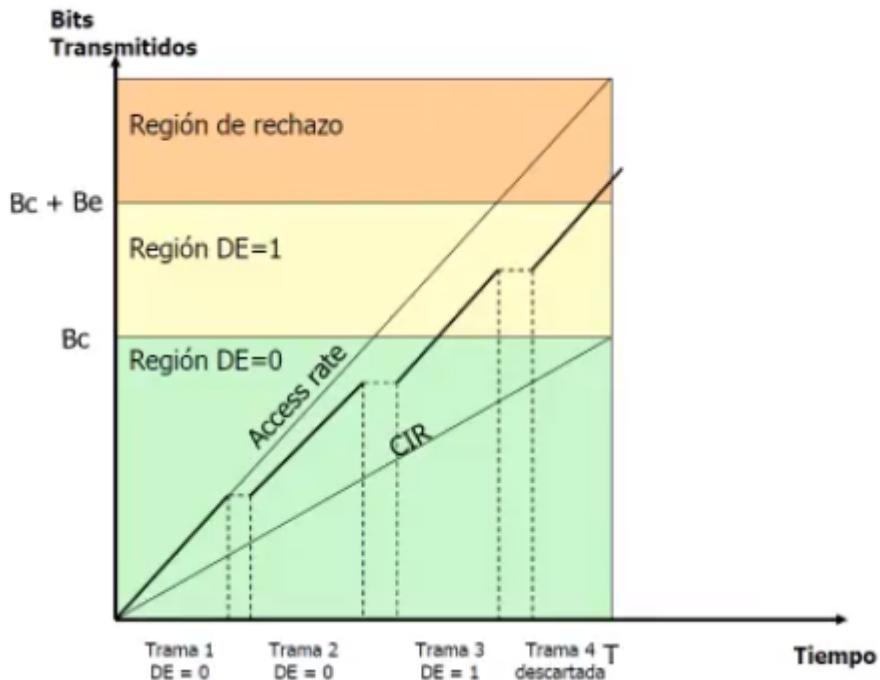
En la imagen los 3 mensajes están debajo de BC por lo tanto son tramas garantizadas.

Ejemplo 2: Tráfico en exceso

Excede el umbral de ráfaga comprometida, a la cuarta trama que va a la región de extensión se la marca con el bit DE (elegible para descarte) activo, produce que si algún nodo empieza a experimentar congestión descarta primero las tramas con DE activo, o sea, posibles descartes. Porque está en la parte de EIR



Ejemplo 3: Nos pasamos de CIR + EIR - exceso del access rate



A la tercera trama le marco DE activo, a la cuarta la descarta, no la deja ingresar a la red porque transmitiría por arriba del enlace de salida.

El proveedor por cosas físicas te da un enlace de última milla más grande que el que contratas para el futuro por eso existe el sector rojo del ejemplo aunque ahí ya no se tiene acceso por límite. No la deja ingresar a la red.

Se puede dar que una transmisión tenga algunas tramas enviadas dentro del cir, otras marcadas como descartables y otras descartadas para no saturar.

Frame relay permite:

Ajustarse a un patrón de ráfaga.

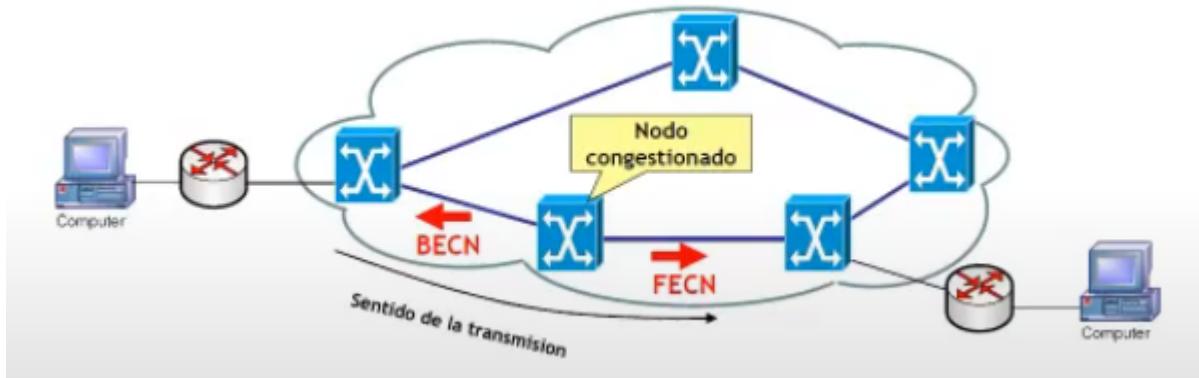
- Tener un valor comprometido: Lo mínimo garantizado. Es lo que pago
- Tener un valor en exceso: Me puedo exceder del valor contratado y todos esos picos que transmito en exceso ingresan a la red y se van a transmitir sin garantía, es decir, si la red tiene recursos el tráfico llega al otro extremo como si tuviera mucho ancho de banda(si hay lugar para enviar envía), si la red está congestionada sólo me garantizan el CIR.

Control de flujo y congestión:

No realiza control de flujo ni de congestión

Sólo realiza notificación de congestión.

Cada nodo tiene puertos y para cada puerto buffers para encolar tráfico, si los buffers se van llenando hasta el umbral de congestión se van activando los mecanismos de control de congestión que descartan tráfico. Antes de activar esos mecanismos de control el protocolo implementa una notificación de congestión.

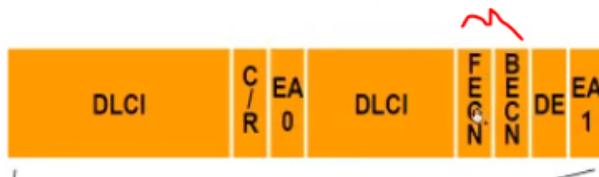


Notificación de congestión:

Avisa a los extremos que va a experimentar congestión y que deben bajar la tasa de transmisión e ingresar menos tráfico a la red.

Se hace con los 2 bits de la cabecera FECN (notificación explícita de congestión hacia adelante) y BECN (notificación explícita de congestión hacia atrás) dependiendo del sentido de la conexión.

Formato de trama



La notificación hacia el nodo que genera el tráfico es para avisar que cese o reduzca la transmisión, y la que va hacia los que no generan ese tráfico es para que le indique al origen que pare.

La acción que se tome en los nodos está fuera del protocolo frame relay

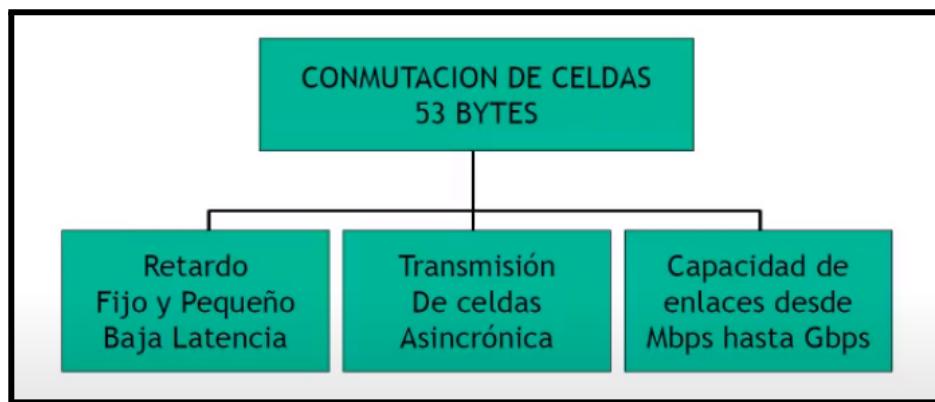
05/10 - grabada

ATM - Asynchronous transfer mode

1. Protocolo orientado al bit
2. Permite una transmisión “transparente”: Capaz de transportar cualquier secuencia arbitraria de bit sin ninguna dependencia de código.
3. Utiliza “Celdas” de longitud fija 53 bytes. Pequeñas.
4. Multiplexación de conexiones lógicas.

5. Soporta múltiples calidades de servicio. (nombramos esto en 802.1P incluído en la etiqueta de 802.1Q que incluía prioridades, y también en el campo tipo de servicio en la cabecera IP).
6. Protocolo de WAN
7. Red de paquetes
8. Es al estilo frame relay porque nos permite multiplexar.
9. Se desarrolló para correr en SDH (jerarquía digital sincrónica).

Características:



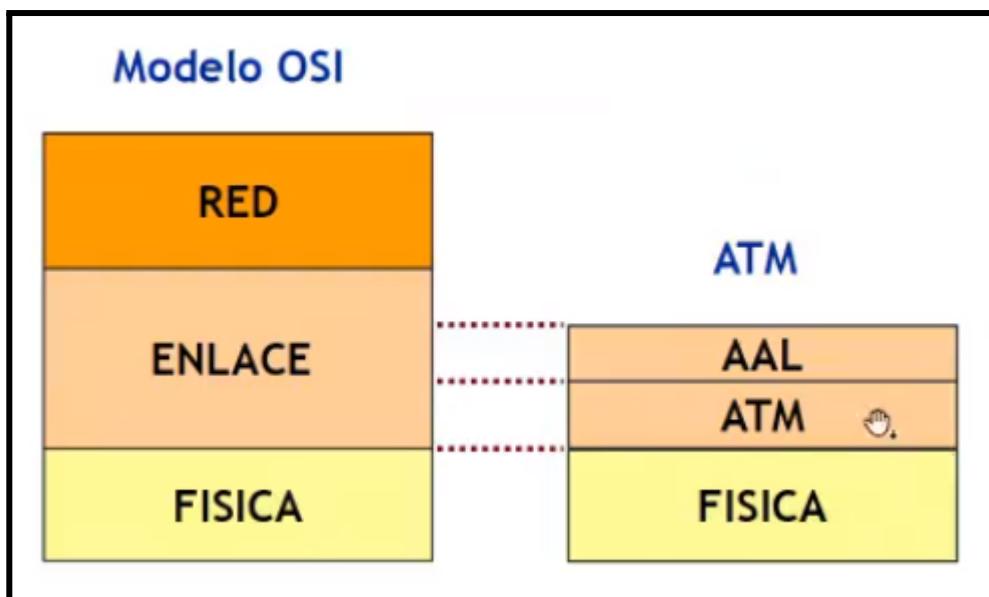
Con los protocolos de tramas variables, cuando van al buffer a encolarse, una trama larga va a tomar el control del procesador durante un tiempo determinado por la longitud de la trama. Con tramas fijas en el ATM y por la calidad del servicio podemos discriminar entre dos flujos cuál es más importante y darle prioridad.

La priorización requiere: diferentes colas en el buffer y priorización que con longitudes variables sería complicado, por eso usa unidades fijas y muy pequeñas.

Retardo fijo:

No es necesario para parsear la trama (donde empieza y dónde termina) sino que es automático, siempre se tarda lo mismo en procesar una celda ATM porque son todas iguales. **SI NO NECESITA ENVIAR NADA SIGUE ENVIANDO CELDAS VACÍAS.**

Dónde lo ubico - Arquitectura de ATM



No cumple todas las funcionalidades que OSI pretende para un protocolo de capa 2.

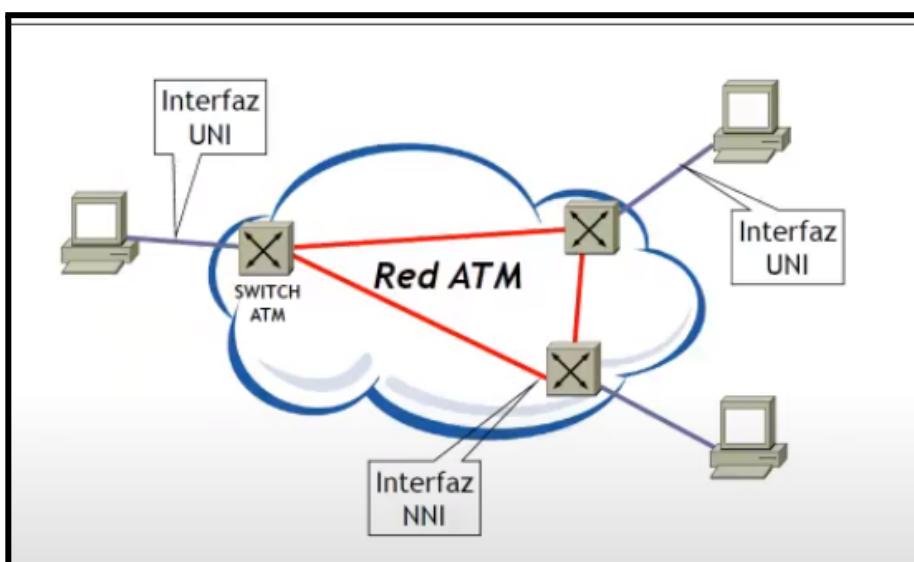
Tiene 2 subcapas dentro del modelo:

1. **AAL**: Capa de adaptación al ATM, necesaria porque los protocolos que corren por encima no están diseñados para tratar con unidades tan pequeñas como lo hace ATM. Para evitar que sea ineficiente la comunicación
2. **ATM**

La red conecta nodos ATM, entre los nodos corre una interfaz llamada NNI.

El usuario se conecta a la red utilizando la interfaz UNI.

Red ATM:



Interfaz NNI:

Network to network interface - interfaz red red. Entre nodos de red.

Interfaz UNI

User to network interface. Hay una ligera diferencia en el formato de la celda entre ambas interfaces.

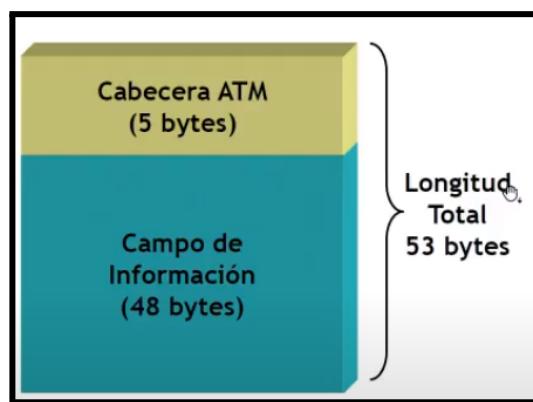
El control de flujo se realiza en el punto de ingreso a la red, y una vez que entra el tráfico solo es transportado, y transporta lo que se acordó con el usuario a menos que haya congestión.

Idea de ATM:

Crear una única red que sea capaz de transportar servicio tanto de voz (canal telefónico) como de datos y video, ser una red integrada.

Formato de Celda:

PDU pequeña de 53 bytes. 10% ocupan la cabecera ATM y el resto información. 48 bytes de carga útil.



Cabecera ATM - UNI:

Entre el usuario y el primer nodo ATM de comunicaciones.

1. GFC:

- Control genérico de flujo, presente sólo en comunicaciones con el usuario
- Para control de flujo de celdas en la interfaz usuario-red

2. VPI:

- 12 bits
- Virtual path identifier - Identificador de camino virtual
- Utilizado para el ruteo dentro de la red.

3. VCI:

- 16 bits
- Virtual channel identifier - Identificador de canal virtual
- Utilizado para el ruteo end-to-end

4. Payload Type:

- 1.Tipo de carga en el campo de datos (0=user, 1=OAM)

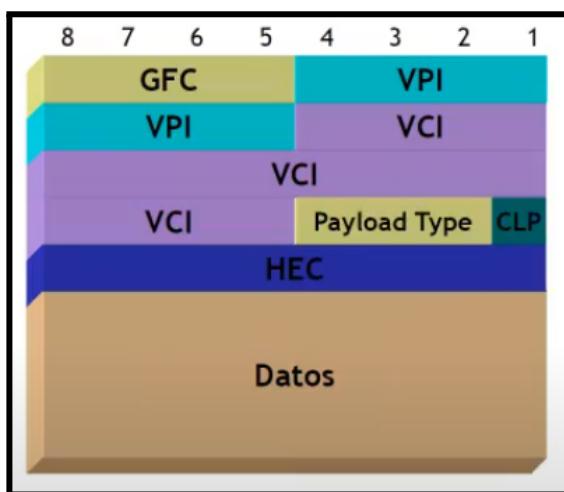
- 2. Indica si hay congestión en la red
- 3. SDU - Un 1 identifica extremo a extremo.
- 3 bits

5. CLP:

- 1 bit
- Cell loss priority - prioridad de la celda. 0=alta y 1=baja (Descartable)
- La red setea en 1 si el user excede alguno de los parámetros de tráfico acordados.
- Similar a DE en Frame relay

6. HEC

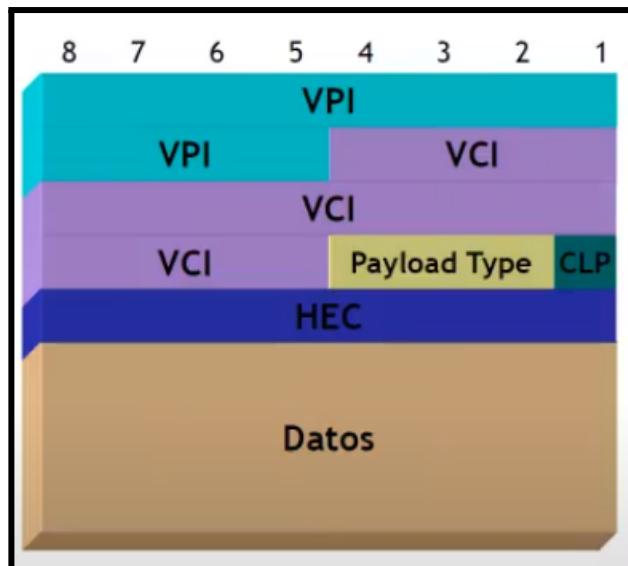
- Control de errores de la cabecera - Header Error Control. (sobre los otros 4)
- Permite corregir hasta 1 error.
- 8 bits. Son muchos bits de control, la redundancia es alta



$VPI + VCI =$ Identificador de la comunicación lógica (igual que el DLCI en frame relay) con niveles de jerarquía.

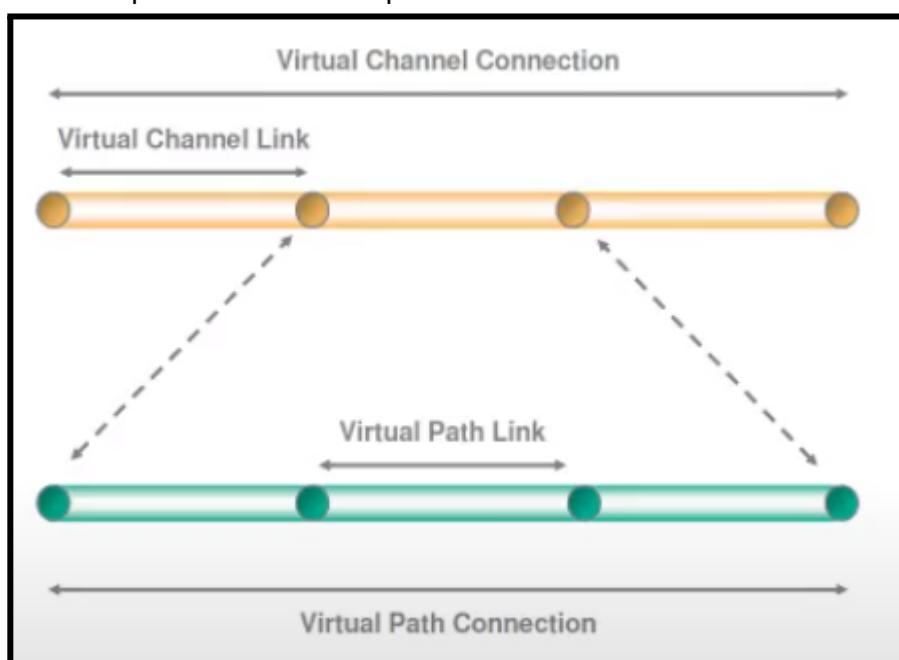
Cabecera ATM - NNI:

Entre nodos ATM de red

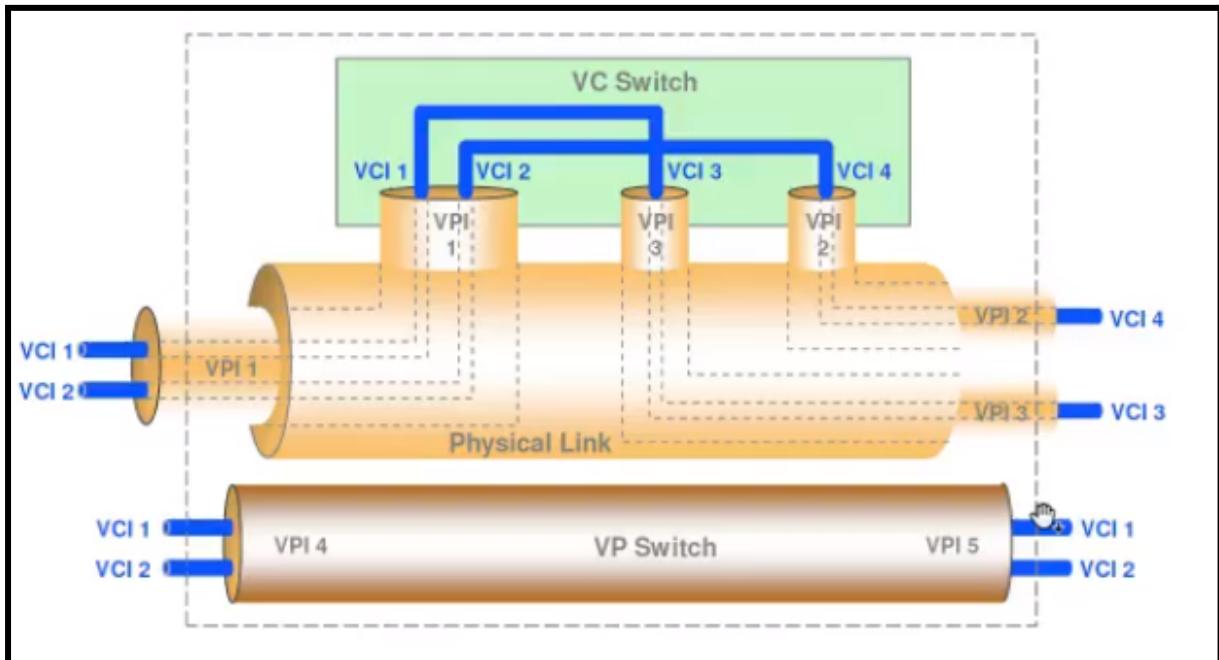


Par VPI - VCI:

Identificador de canal lógico (similar a DLCI en frame relay). Significado Local.
Se tuvo en cuenta el punto de vista del operador del servicio de telecomunicaciones



Los canales punto a punto van lógicamente dentro de caminos virtuales



Es un dispositivo donde ingresa interfaz ATM de entrada y otra de salida.

VP switch:

- Commuta solo el VPI, o sea commuta sólo caminos, commuta todos los canales que van dentro de ese camino (Ej: el VCI 1 y 2).
- Permite manejar de forma más sencilla caminos para hacer ingeniería del tráfico moviendo grupos de conexiones de un nodo a otro
- Se encuentran en el core de la red.

Los VCI de entrada y de salida se mantienen

VC switch:

- Es un poco más complejo
- Commuta los CV, canales, que están en un mismo camino en distintos caminos porque las comunicaciones requieren destinos diferentes que requieren caminos virtuales diferentes.
- Los VCI de entrada y salida se modifican

El par VPI - CPI tiene significación local, lo que ve un usuario de un lado como su VPI CPI no es lo mismo que ve otro usuario.

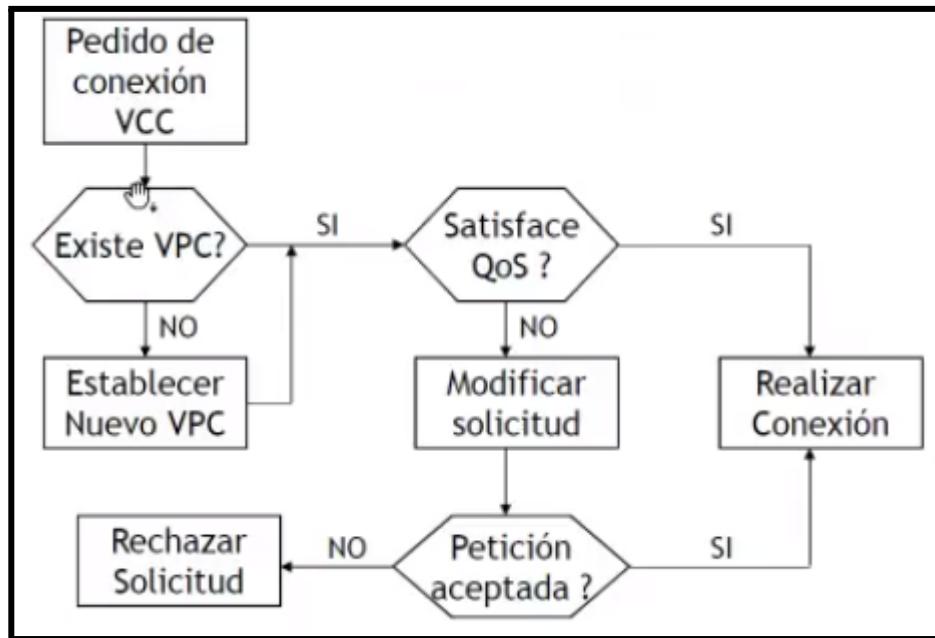
Se pueden realizar entre:

Tienen diferentes tipos

1. Usuario - usuario: Transportan datos de usuario de extremo a extremo
2. Usuario - red: Pueden utilizarse para “agregar” tráfico de un usuario hacia un network server. Un usuario se conecta a un nodo para señalización (indicar con quién se quiere comunicar)
3. Red - red: Transporta información de ruteo.

Conexión ATM:

Establecimiento de una conexión usando Virtual Path VP



Un usuario en un extremo de la red realiza un pedido de conexión, utilizando un protocolo de señalización (me quiero conectar con X extremo e indica ancho de banda necesario y cuales son las características del tráfico que quiere cursar).

La red verifica si existe un camino que lo lleve al extremo. Si no existe establece uno nuevo utilizando señalización interna de la red.

Si existe se verifica si satisface la calidad del servicio chequeando si tiene suficientes recursos para las características de la petición que dio el usuario.

Si no cumple la calidad del servicio se notifica al usuario y el usuario puede modificar la solicitud que debe ser aceptada por la red y sino no hace nada.

HEC - Header error check:

- Procedimiento de detección de errores similar al usado por HDLC
- Utiliza un HEC de 8 bits para controlar errores en los 32 bits restantes.
- Utiliza el polinomio estandarizado: $X^{16} + X^2 + X + 1 \rightarrow$ CRC
- Permite la corrección de 1 bit erróneo. A más de uno lo descarta.
- Tiene la funcionalidad de detectar y corregir un bit errado en la cabecera y además del sincronismo

Diagrama de estados:

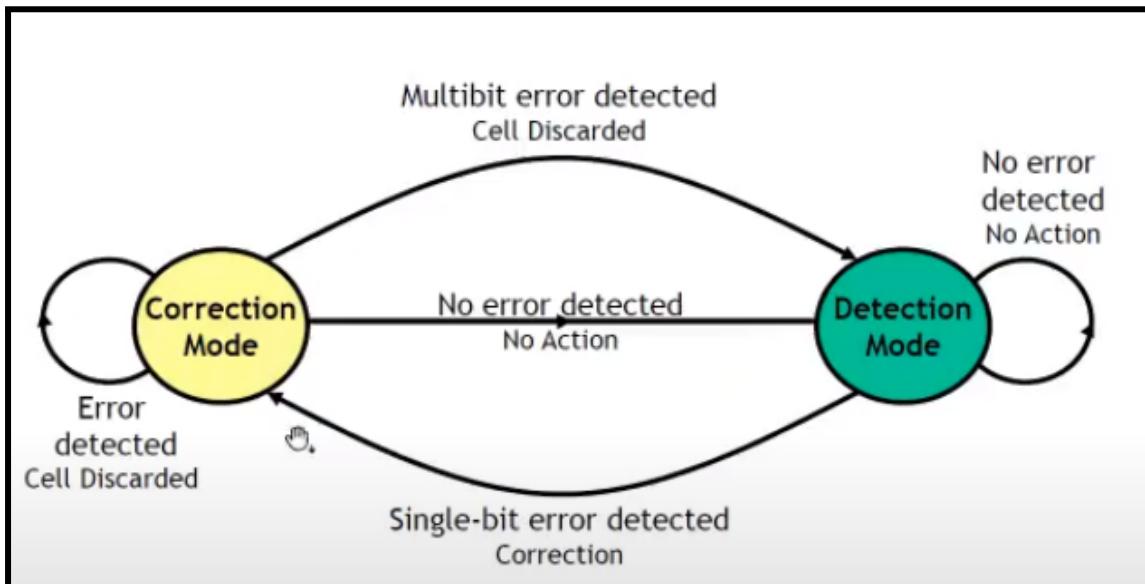
En una conexión establecida donde fluyen celdas en ambos sentidos. El nodo está en:

- **Modo detección:**

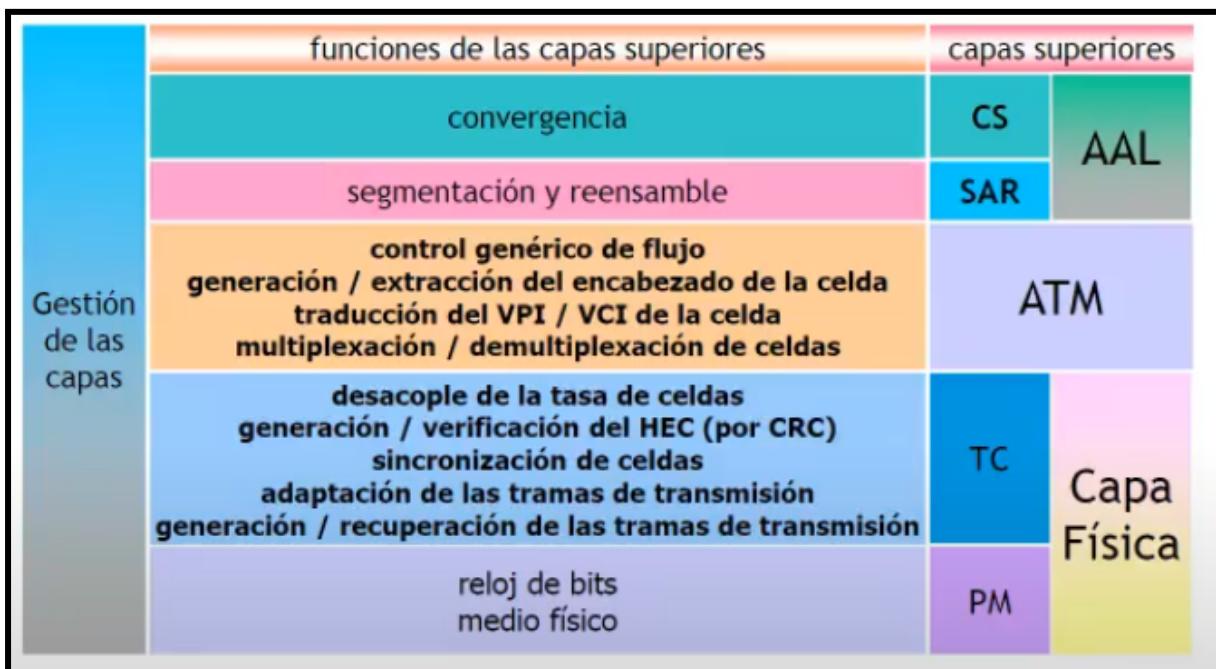
No suele detectarse ningún error y queda en esta parte hasta que un HEC detecta 1 error (más de 1 bit erróneo descarta de una), pasa a modo corrección.

- Modo corrección:

Corrige bit errado y si la siguiente celda da correcta vuelve al modo detección. Si la siguiente celda tiene errores también, la descarta y pasa a otro tipo de diagrama de estados.



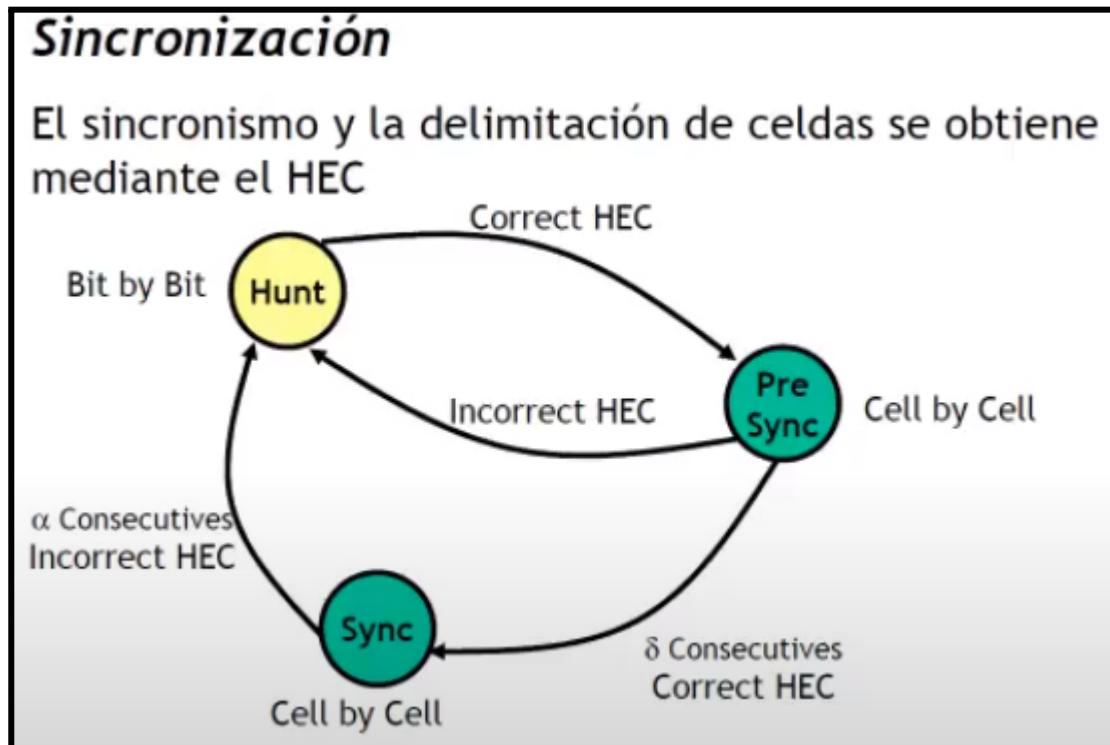
Modelo ATM en detalle:



Capa física - Funciones:

- Delimitación de celdas
- Monitoreo de errores
- Inserción de celdas vacías (idle): Porque va una celda a continuación de otra. Si no hay nada para transmitir inserta celdas vacías para mantener sincronismo.

- Mantenimiento del sincronismo



Como todas las celdas son de 53 bytes sé dónde empiezan y terminan todas.

1 Estado de captura - Hunt:

Toma de a un bit por vez.

Un nodo empieza a transmitir un flujo de bits. El extremo remoto recibe el flujo y trata de definir dónde comienza y termina cada secuencia para establecer sincronismo, le calcula el HEC, si da incorrecto se corre un bit y vuelve a calcular HEC y así sucesivamente hasta que obtiene lo que cree que es una celda porque le da el HEC correcto y pasa a estado de pre sincronismo.

2 Estado pre sincronismo - Pre Sync:

Toma de a una celda por vez (53 octetos de bit).

Toma los siguientes 53 bytes y vuelve a calcular el HEC, si da incorrecto el HEC correcto anterior fue una casualidad pero no era el comienzo de la celda y vuelve al estado de captura.

Si los nuevos 53 bytes dan correcto, se corre otra celda más y así sucesivamente. Luego de delta HECS correctos seguidos pasa a un estado de sincronismo.

3 Estado de sincronismo - sync:

Se corresponde con el modo de detección del diagrama de estados anterior.

Si sucede que se recibe una celda errónea se pasa a modo de corrección, etc.

Si pasa alfa HECS incorrectos en vez de tratar de corregir esos errores se da cuenta que perdió el sincronismo y vuelve al estado de captura.

Capa de adaptación al ATM:

- Manejo de los errores en la transmisión
- Segmentación y reensamblado
- Manejo de las celdas perdidas o mal insertadas
- Control de flujo

Recibe el PDU de la capa usuaria, le agrega su PDU, lo pasa a la subcapa de segmentación y reensamblado que lo parte en 48 octetos donde la subcapa ATM le agregue el VPI-CPI-HEC. etc.

Clases ATM:

ATM maneja los distintos servicios (voz, datos, video) en distintas clases que manejan el tráfico de distinta forma.

Marco conceptual de cómo se van separando las clases de tráfico y como ATM las maneja.

| | Class A | Class B | Class C | Class D |
|--|-------------------|---------------------|--------------------------|----------------------|
| Characteristics | Constant bit rate | Variable bit rate | Connection oriented data | Connection less data |
| Synchronization between Source and Destination | Required | | Not Required | |
| Bit rate | Constant | | Variable | |
| Connection Type | | Connection Oriented | | Conn. less |
| Adaption Layer | AAL 1 | AAL 2 | AAL 5 | AAL 3/4 |

AAL1: Diseñado para tratar flujos de bits constantes como la voz y vídeo sin comprimir.

AAL2: Voz y vídeo comprimido porque la compresión depende de la entrada, a veces se comprime más a veces menos (más o menos tráfico).

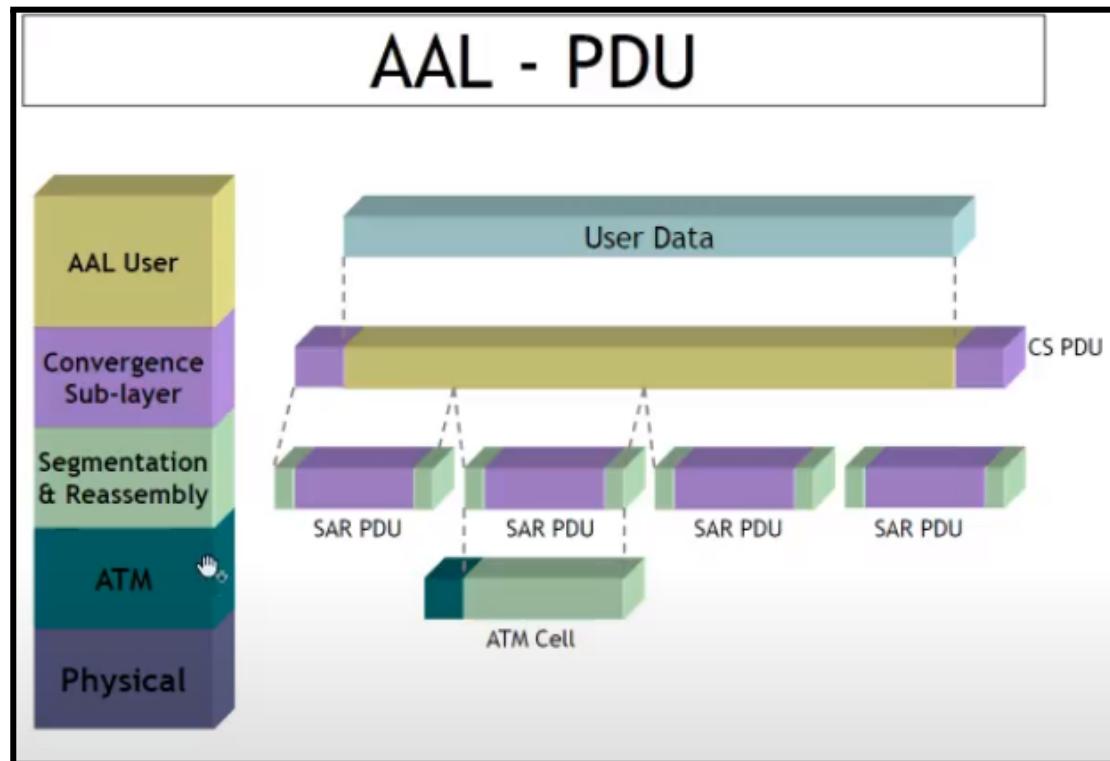
AAL5: Datos, orientada a la conexión.

AAL 3/4: Datos, no orientada a la conexión

Con jerarquía como si fuera el modelo OSI:

- Capa física
- Capa 2: ATM y adaptación (convergence sublayer y segmentation and reassembly)

- Capa usuaria, superior: AAL User. Invoca los servicios de la capa 2.
 - AAL 1,2,3-4 O 5.



Categoría de servicios:

Esto hace a la implementación de como un usuario hace una petición mediante el protocolo de señalización a la red para establecer una conexión extremo extremo con determinadas características. Debe indicar qué categoría quiere.

Cada categoría requiere una determinada cantidad de recursos de la red.

Real-Time Services:

- Constant bit rate CBR: Se adecúa a una clase A.
- Real-time variable bit rate VBR: Se adecúa a una clase B

Non-Real-Time Services:

Se adecúa más a la transmisión de datos.

- Non-Real-Time variable bit rate nrt-VBR: Desde hasta una tasa de bit determinada
- Available bit rate ABR: El usuario puede transmitir sobre la capacidad ociosa de la red.
- Unspecified bit rate UBR: No se especifica ningún tipo de compromiso. Modalidad best effort.

Atributos de tráfico:

Lo debe indicar el usuario. Descriptor de tráfico fuente acordado.

Estos tráficos van a monitorear la red para que el usuario no se pase, si lo hace se marca la celda como descartable.

1. Peak cell rate (PCR):

Pico de celdas - Velocidad máxima de transmisión (que tan larga es la ráfaga).
Límite máximo en la tasa de celdas entregadas a la red.
Se define en función de T intervalo mínimo entre celdas.
Obligatorio para servicios CBR y VBR. (Clase a,b)

2. Sustainable Cell Rate (SCR):

Límite al promedio de la tasa de transmisión.
Tasa sostenida para bits variables.
Definido en función de X mayor a T.
Requerido para VBR. (Clase b porque para a SRC = CBR)

3. Maximum Burst Size (MBS):

Cantidad máxima de celdas enviadas continuamente a velocidad PCR.
Tamaño máximo de la ráfaga
Requerido para VBR

4. Minimum cell rate (MCR):

Utilizado en ABR
Define la tasa mínima requerida a la red.

Parámetros de QoS:

Parámetros de calidad del servicio con los que el usuario indica qué calidad espera de la red.

Tienen diferente incidencia dependiendo el servicio que esté utilizando.

1. Peak-to-peak cell delay variation:

Variación máxima soportada en el retardo. Variación del retardo entre celdas.
Se mide entre el umbral de descarte y el maxCTD.
En servicios isócronos (voz y video).
denominado Jitter.
La variación de retardo me genera más retardo (por el buffer donde debo meterlos para sacarlos con el retardo inicial)

2. Maximum Cell Transfer Delay:

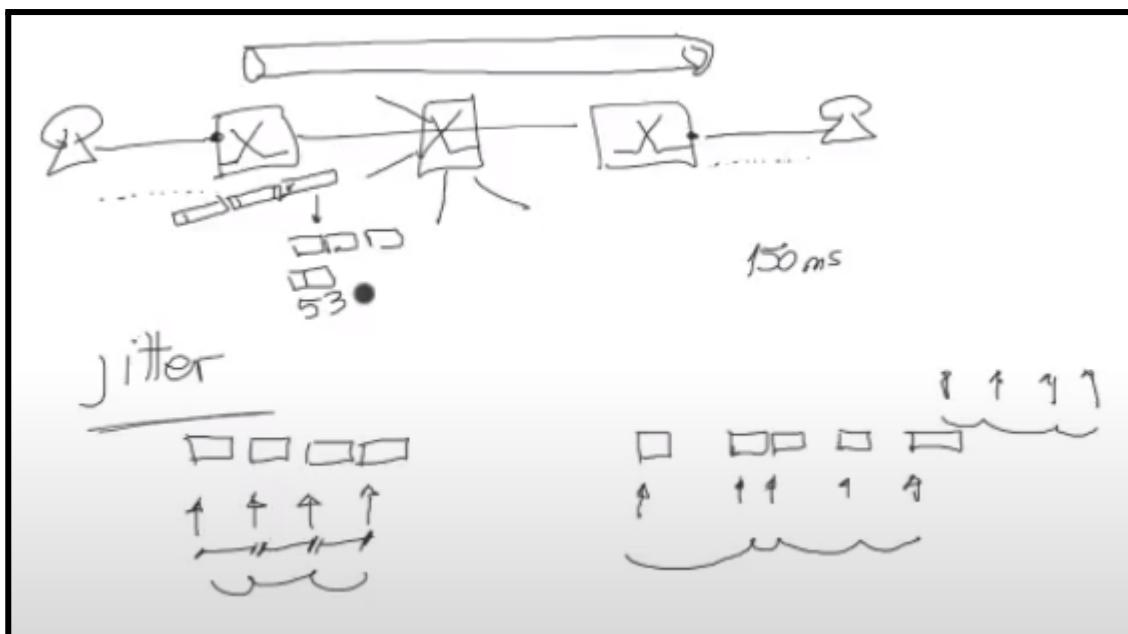
Tiempo entre la transmisión del último bit en la UNI, y la recepción del primer bit en la UNI destino.
Retardo máximo que tolera la conexión extremo a extremo
Es el tiempo que tarda un bit que ingresa en una interfaz en salir por otra (Atravesar la red)

3. Cell Loss Ratio:

Tasa de pérdida (De celdas) máxima soportada en la conexión.

Depende la cantidad a veces es imperceptible en una llamada por ejemplo.

La clase A es casi insensible a la pérdida, la clase B es sensible.



Call Admission Control - CAC:

En resumen, cuando el usuario solicita una nueva VCC debe especificar los servicios requeridos en ambas direcciones para esa conexión:

1. Categoría del servicio
CBR, rt-VBR, nrt-VBR, ABR o UBR.
2. Descriptor del tráfico
PCR, SCR, MBS, MCR.
3. Indicar la calidad de servicio que espera de la red
Peak-to-peak CDV, maxCTD, CLR.

Con todo eso la red puede determinar si puede cursar el llamado o no.

UPC - User parameter control:

Control realizado por la red en el primer punto de acceso al VCC (una vez que acepte lo requerido por el usuario va a controlar que se ejecute lo acordado).

Controla que el usuario cumpla con los valores contratados de:

- PCR: Tasa pico
- SCR: Tasa sostenida

Si se excede marca las celdas como excedidas (prioridad de la celda)

Aplicaciones de ATM:

Ya no se usan:

| Aplicaciones ATM |
|-------------------------|
| • LAN Emulation |
| • Classical IP over ATM |
| • Trunking de VLANs |

Se usa en el backbone de proveedores de servicio pero cada vez menos.

MPLS - Multiprotocol Label Switching

Comutación de etiquetas multiprotocolo

- Mecanismo de conmutación basado en etiquetas (labels)
- Diseñado para transportar múltiples protocolos de capa 3. Ej: IP (IPv4 e IPv6) y cualquier otro.
- Las etiquetas MPLS identifican las redes destino: Busco a partir de esta etiqueta que identifica el destino al que llega el paquete cómo realizar ese paquete de conmutación.
- Es una capa 2.5, ni capa 2 ni capa 3.
- orientado a la conexión
- Aumenta velocidad de procesamiento de paquetes y gana en latencia

Brinda una forma de conmutar más eficiente y más rápido que las tablas de ruteo:

El router en lugar de ir a la cabecera IP con el proceso de abajo, va a ir a la cabecera MPLS y busca el campo “etiqueta” string de longitud fija y busca una coincidencia exacta.

Ruteo IP (sin MPLS)

- Protocolos de ruteo distribuyen información de ruteo. Cada router se conecta con otro router y comparte la información que tiene con las redes conectadas. Todos los routers conocen la existencia y ubicación de cada una de las redes y puede armarse con información parcial una imagen de dónde conmutar los paquetes para llegar a destino.
- La conmutación se basa en
 - Cabecera del paquete: La dirección IP destino
 - Tabla de ruteo local: Chequea/ contrata la IP destino.
- Búsqueda independiente realizada en cada salto: Cada paquete que llegaba a un router el router hacía los mismos pasos hasta conmutar.
- Desventaja: El uso de tablas de ruteo se complejiza bastante cuando la red se hace muy grande. Las tablas crecen en entradas y la ip destino que se contrasta se debe

contrastar con muchísimas entradas “parciales” y hace todo el seleccionamiento con la entrada que mejor encaje (longest match).

Beneficios de MPLS:

- Ruteo de IP unicast y multicast
- VPN: Aplicación central desde el punto de vista del proveedor.
- QoS: Calidad de servicio que no es muy utilizada en IP.
- MPLS reduce la tarea de commutación en el “core”: El procesamiento al mensaje es más rápido a partir de buscar un match exacto de un string de longitud fija.
- Puede transportar otros protocolos, no solo IP

Arquitectura de un router MPLS:

Conmuta paquetes IP y etiquetas.

● Plano de control:

Todos aquellos procesos que corren en el router que tienen una función administrativa o de control.

El protocolo de ruteo intercambia información con los otros equipos (Ej con RIP pasa en paquetes su tabla de ruteo) tanto información de ruteo (prefijo y longitud de prefijo) como también información de etiquetas.

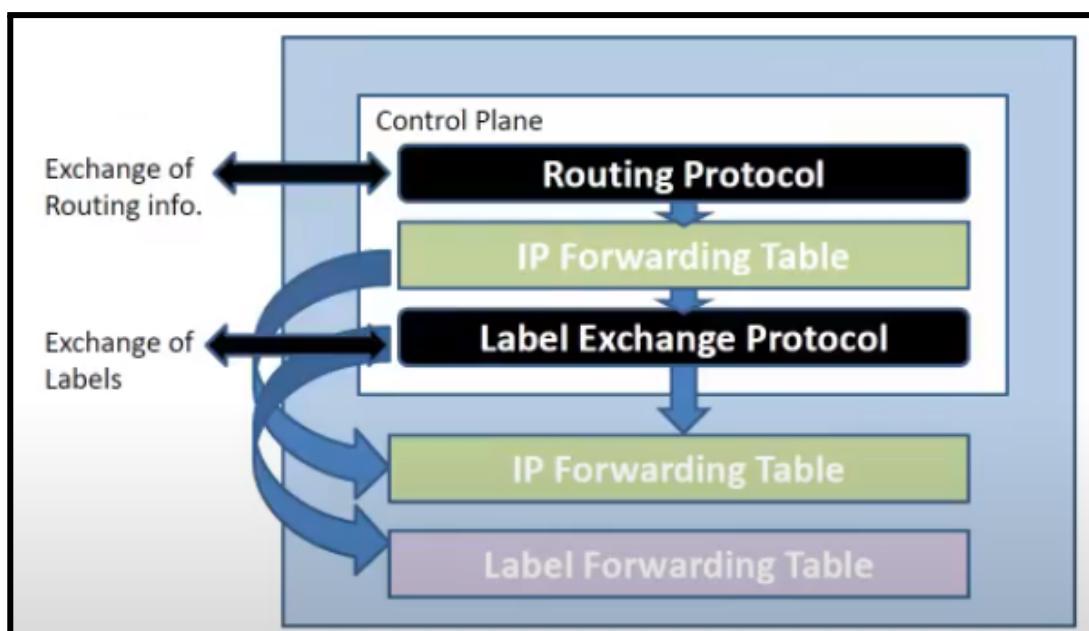
● Plano de datos:

Parte del router que se encarga de la commutación del tráfico.

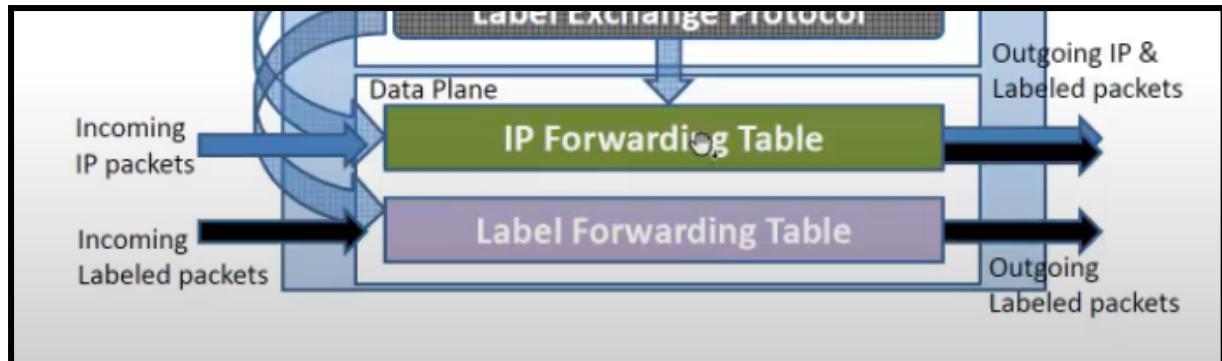
Implementado por la tabla de ruteo

Se encargan de las dos tareas centrales del router, identificar el mejor camino dentro de la tabla de ruteo y conmutar el datagrama IP.

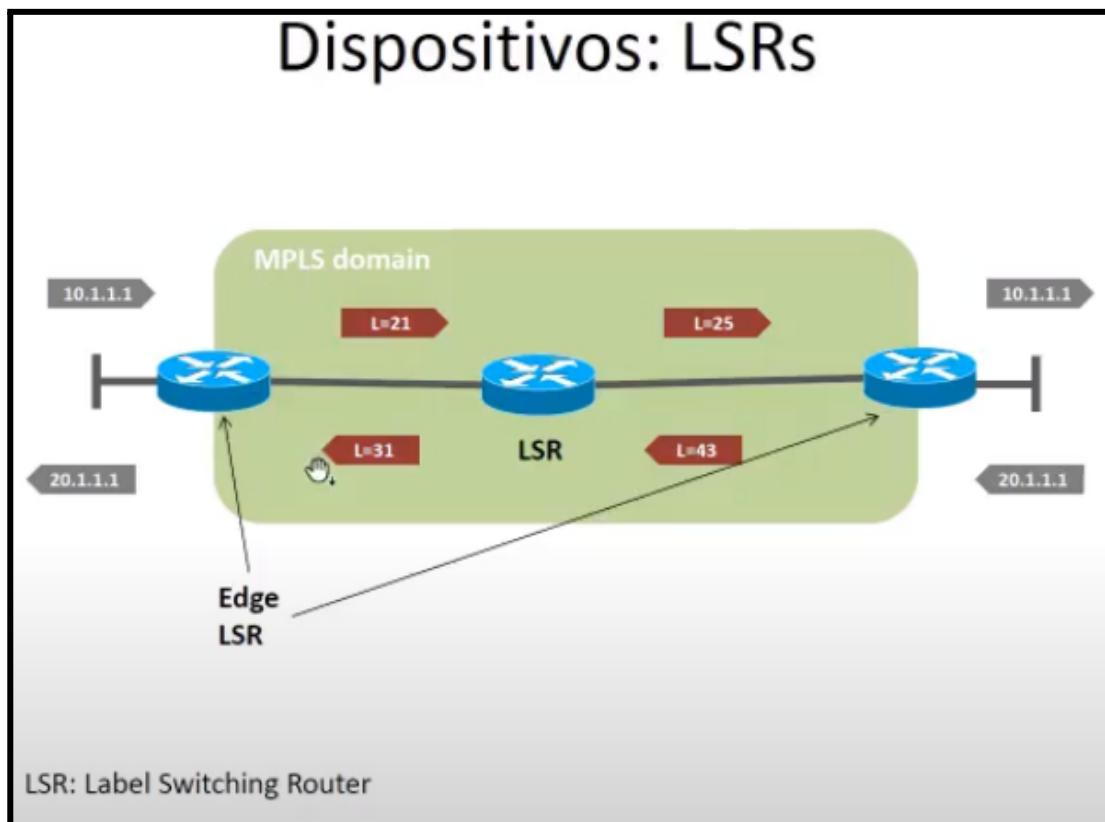
Cuando un router usa MPLS además de conmutar paquetes conmuta etiquetas.



El paquete IP que ingresa busca en la tabla de ruteo como sacarlo, al conmutar puede salir un paquete IP tal cual o puede ser un paquete etiquetado.
 Si ingresa un paquete etiquetado sale un paquete etiquetado.



El dominio MPLS ocurre dentro de la red del proveedor.



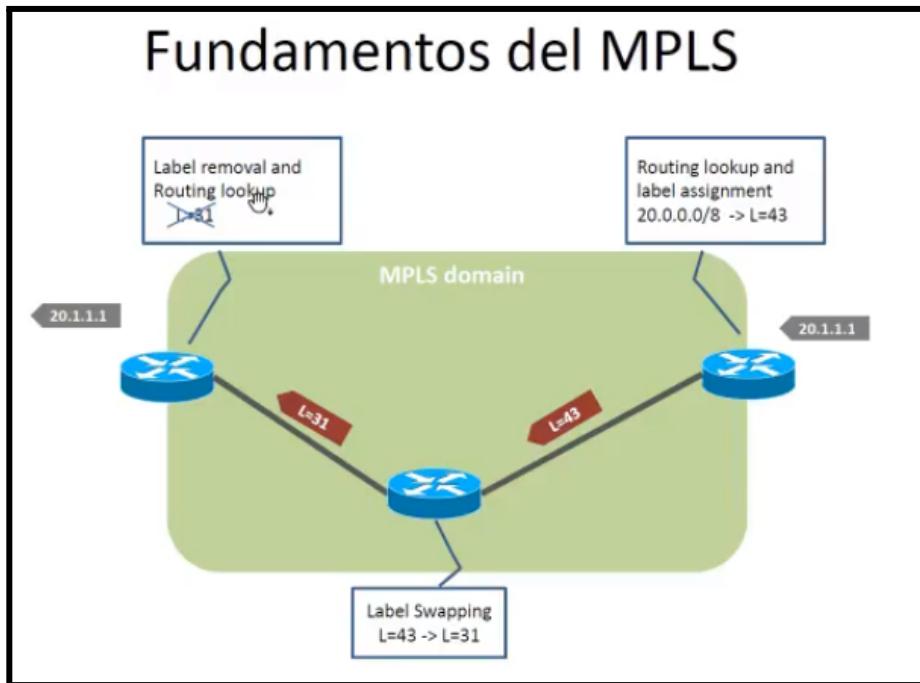
Label Switching router:

- Routers que pueden hablar IP y MPLS
- De 2 tipos:
 - **Edge LSR:** Toman un paquete IP, lo analizan y cuando lo conmutan hacia adentro de la red el paquete va a estar etiquetado. Cuando reciben un paquete etiquetado le quitan la etiqueta y lo entregan como paquete IP.

- **Core LSR:** Donde confluyen los flujos de los clientes soporta más volumen de paquetes.
Solo habla MPLS y es mucho más eficiente que un router normal

Etiquetas MPLS:

- Identificador de 4 bytes
- Define el destino y el servicio de un paquete
- Identifica un FEC (forwarding equivalent class)
- Tiene significado local
 - Cada LSR (router) mapea una etiqueta a una FEC
 - Esta asociación es intercambiada entre los LSRs



Cómo sabe el router de la derecha qué etiqueta tiene: El proceso que corre en el plano de control del router core le indica a los routers edge que etiqueta corresponde a cada IP

Label distribution protocol: protocolo en el que los enrutadores capaces de intercambiar etiquetas multiprotocolo intercambian información de mapeo de etiquetas. Un router MPLS le indica a otro con este protocolo qué etiqueta le pone a la red.

Forwarding Equivalent Class:

Una FEC es un grupo de paquetes tratados:

- De la misma manera
- Sobre un mismo camino

Identifica a una red destino pero dentro de una misma red puedo tener dos flujos distintos (Ej flujo urgente y flujo diferido).

La conmutación de paquetes MPLS consiste en:

- Asignar un paquete a una FEC determinada
- Determinar el próximo salto para cada FEC.

MPLS es orientado a la conexión

Formato de la etiqueta:

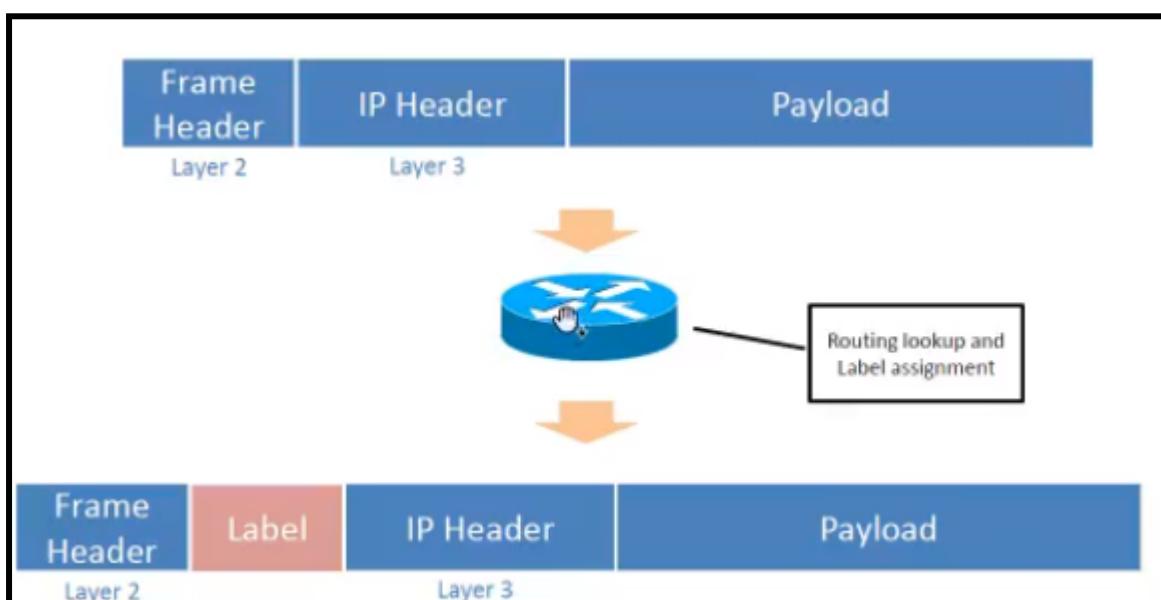


1. Label: La etiqueta string de 20 bits que identifica a FEC.
2. Experimental
3. Indicador de última etiqueta: Indica si hay o no más etiquetas, se pueden anidar etiquetas pero no se ve en el curso.
4. Time-to-Live: 8 bits que cumplen la misma función que cumple el time to live en IPV4. El valor que se decrementa al pasar por cada salto/router. Como entre routers MPLS no tocan la cabecera IP sino la MPLS se decrementa desde ahí

MPLS Labels:

Donde se inserta esta etiqueta.

El router de la imagen es un edge (extremo) donde Ingresa un datagrama IP, le quita el header de capa 2 y analiza el header de capa 3 busca en la tabla de ruteo la ip destino y ve que tiene que salir por x interfaz donde se tiene que etiquetar. Entonces el router pone la etiqueta

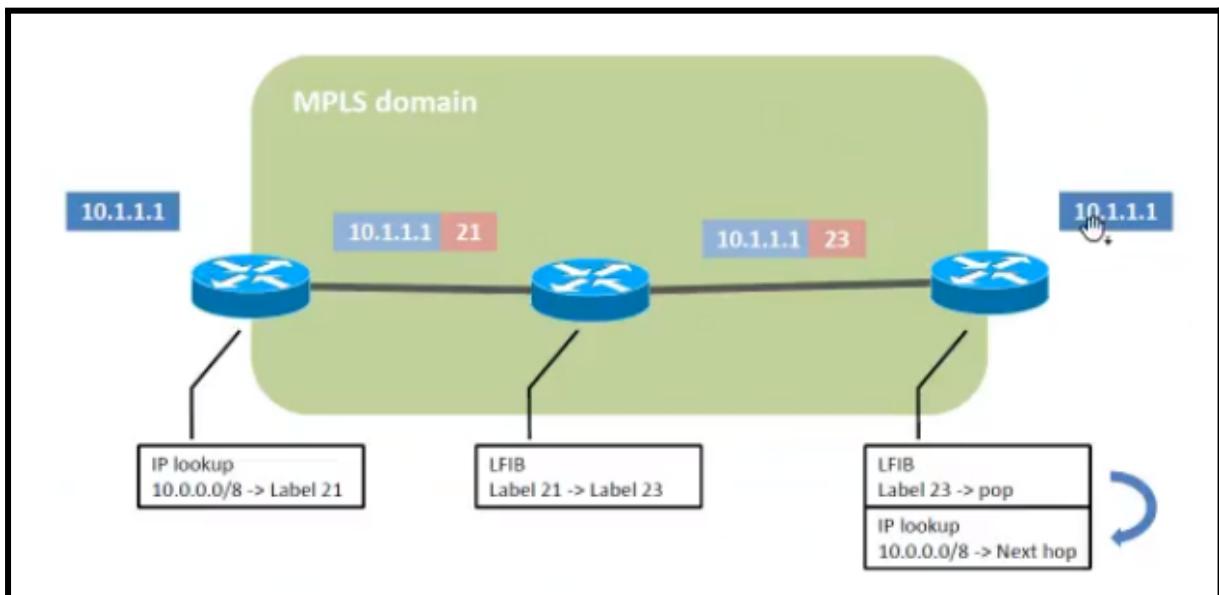


Lo que entra como trama ethernet sale como trama ethernet pero con una etiqueta.

MPLS en OSI:

- No cumple la función de un protocolo de capa 2 (no encapsula el datagrama), en el ejemplo sigue siendo ethernet.
- No cumple todas las funciones de un protocolo de capa 3 según OSI como el enruteamiento o la resolución de direccionamiento como hace IP.
- Es una capa 2.5, ni capa 2 ni capa 3.

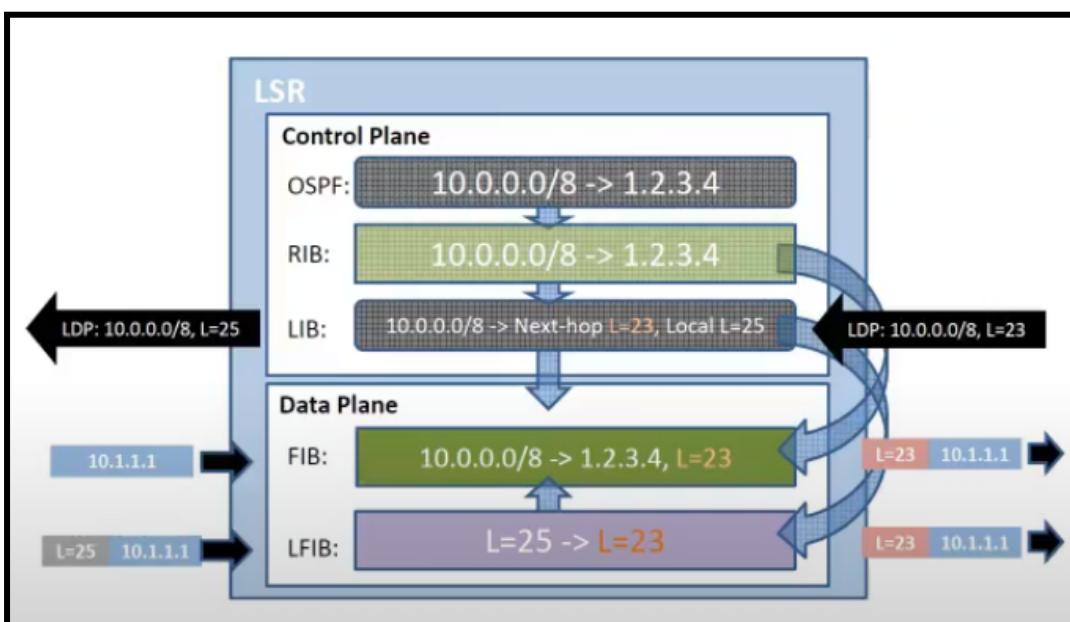
Otro ejemplo del funcionamiento:



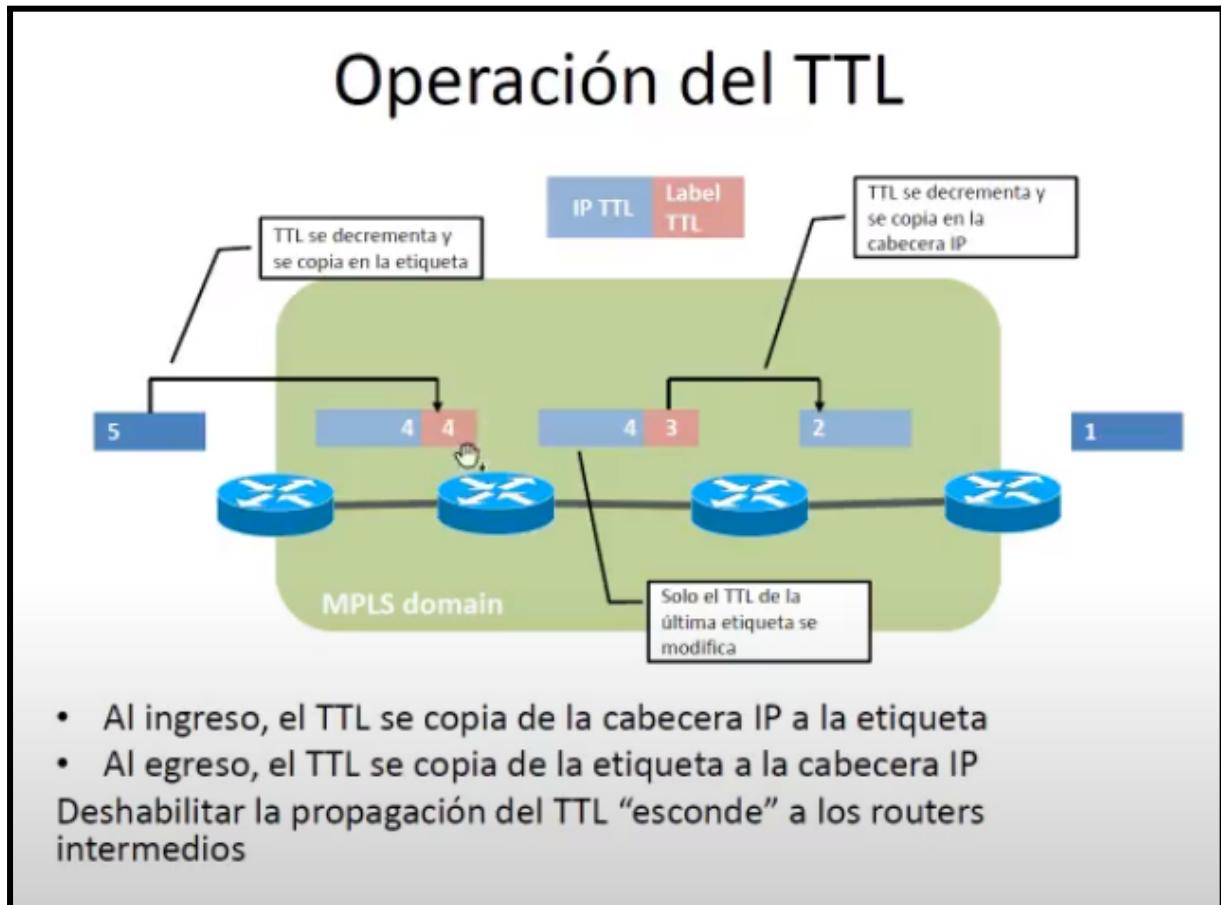
El lookup (búsqueda en tabla de ruteo) lo hacen los extremos, dentro de la red se usan solo etiquetas.

El extremo izquierdo sabe que una IP corresponde a la etiqueta 21 porque qué router de etiquetas (central en este ejemplo) le aviso antes. El router del medio tiene tabla de etiquetas.

Cómo funciona todo dentro del router:



Otro ejemplo:



Si se deshabilita la actualización del TTL de IP por cada salto de router MPLS que pasa, desde la vista del usuario todo el entorno MPLS se ve como un único gran salto. Modifica el tiempo de vida de un paquete en IP porque hay más saltos que no se cuentan.

Arquitectura MPLS VPN:

Beneficio en la práctica de MPLS. (Lo vemos por arriba en el curso)

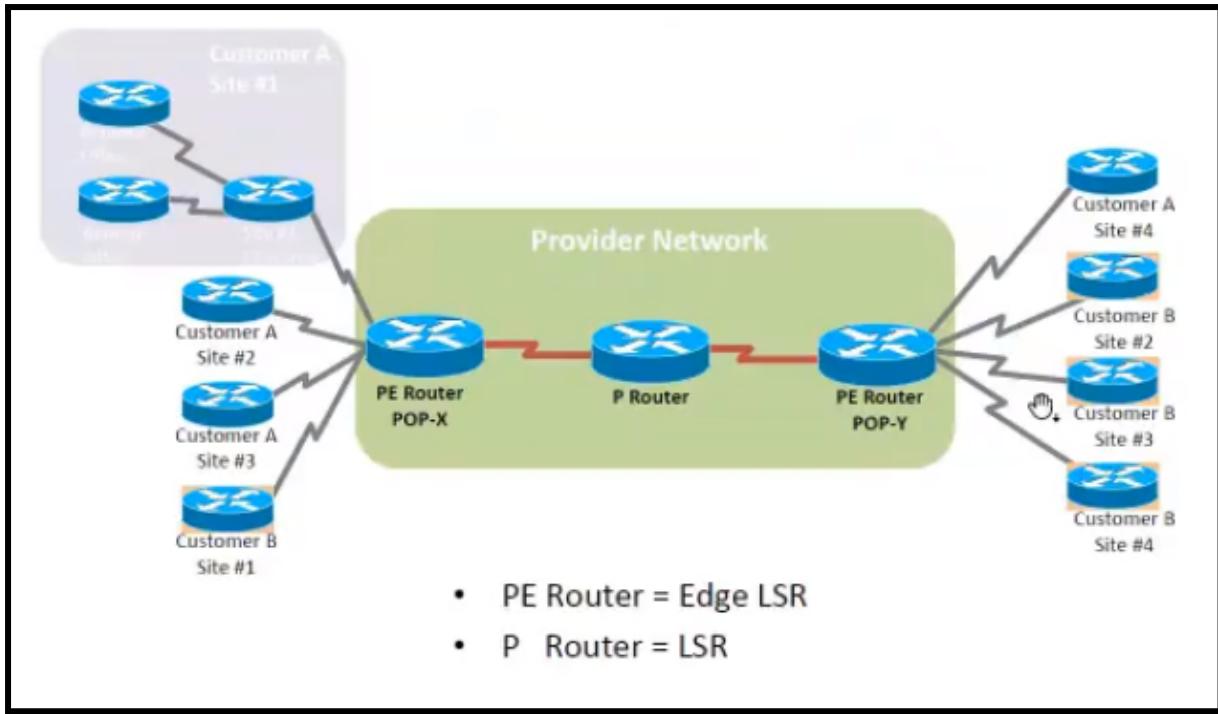
No es lo mismo que una VPN IPSec

Haciendo una red pública (además de internet, redes de proveedores por ejemplo) donde hay varios clientes dentro de una red MPLS de un determinado proveedor.

Permite que el cliente B no se cruce para nada con el tráfico del cliente A. Están completamente aislados. Forme una red privada virtual entre todos los sitios de B

Me permite configurar los routers para que solo vea a los routers de mi cliente y ve a la red MPLS con varios clientes como su red individual.

No se arma un circuito virtual permanente entre el router site de un customer con otro del mismo customer que deba seguir un determinado camino (túnel) sino que permite que todos hablen con todos. No tenés que ir a un router central como otros tipos de VPNS.



12/10

Seguridad

Criptografía

Ciencia de leer y escribir mensajes codificados. Es el componente fundamental en los mecanismos de:

- Autenticación
- Integridad
- Confidencialidad

Autenticación:

Establece la identidad de ambos, el transmisor y el receptor de la información

Integridad:

Asegura que los datos no han sido alterados

Confidencialidad:

Asegura que nadie, excepto el transmisor y receptor son capaces de interpretar los datos transmitidos.

Vinculación o no repudio:

Permite vincular un documento o transacción a una persona o un sistema de gestión criptográfico automatizado.

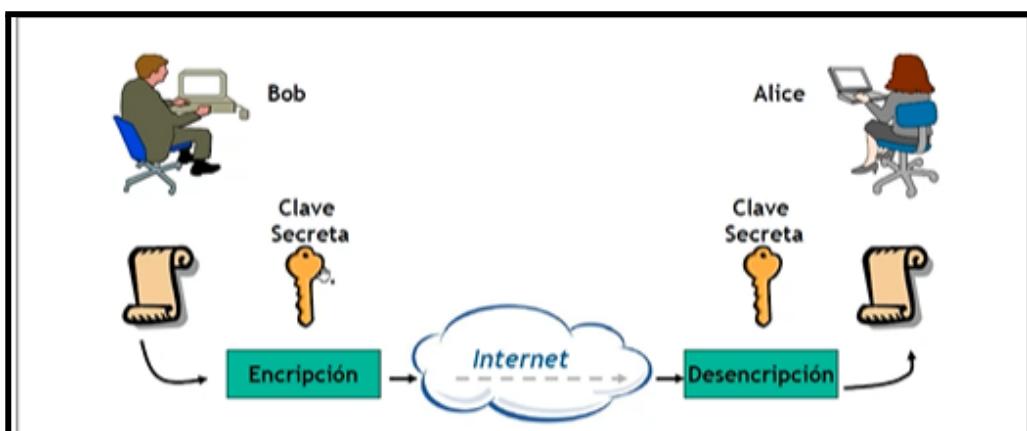
Encriptar:

Tenemos un mecanismo criptográfico que es un algoritmo (función matemática) que usa un valor secreto llamado “clave” para encriptar el mensaje.

La longitud de la clave es importante para la fortaleza del mecanismo de encriptación. Más difícil es la prueba de fuerza bruta (aplicar todas las combinaciones posibles hasta encontrar la clave).

Encripción Simétrica:

De “clave secreta” que utiliza una única clave en común y el mismo algoritmo criptográfico para encriptar/desencriptar el mensaje.



El eslabón clave: La clave secreta, a muchos participantes es compleja la distribución de claves.

Ventajas

Más simple y eficiente que la asimétrica en cuanto a requerimientos de cómputo. Ej: Se usa para el inicio de la sesión y después se cambia

Algunos algoritmos usados:

- Data encryption Standard DES: Roto por fuerza bruta ya no se usa.
- 3DES: Lo mismo.
- Rivest Cipher 4 RC-4: Primitivo, se rompe por fuerza bruta. Se usaba porque el poder de cómputo de la época gastaban las baterías de los dispositivos rápido.
- International Data encryption Algorithm IDEA
- AES (Advanced Encryption Standard) 128 - 192 - 256

El problema es cuando esa clave se empieza a divulgar, ya no tiene sentido el mecanismo.

Encripción Asimétrica:

Conocida como encriptación de clave pública. Los extremos pueden o no utilizar el mismo algoritmo pero complementario para encriptar y desencriptar la información. Dos valores diferentes, clave pública y privada pero complementarias.

Se obtiene:

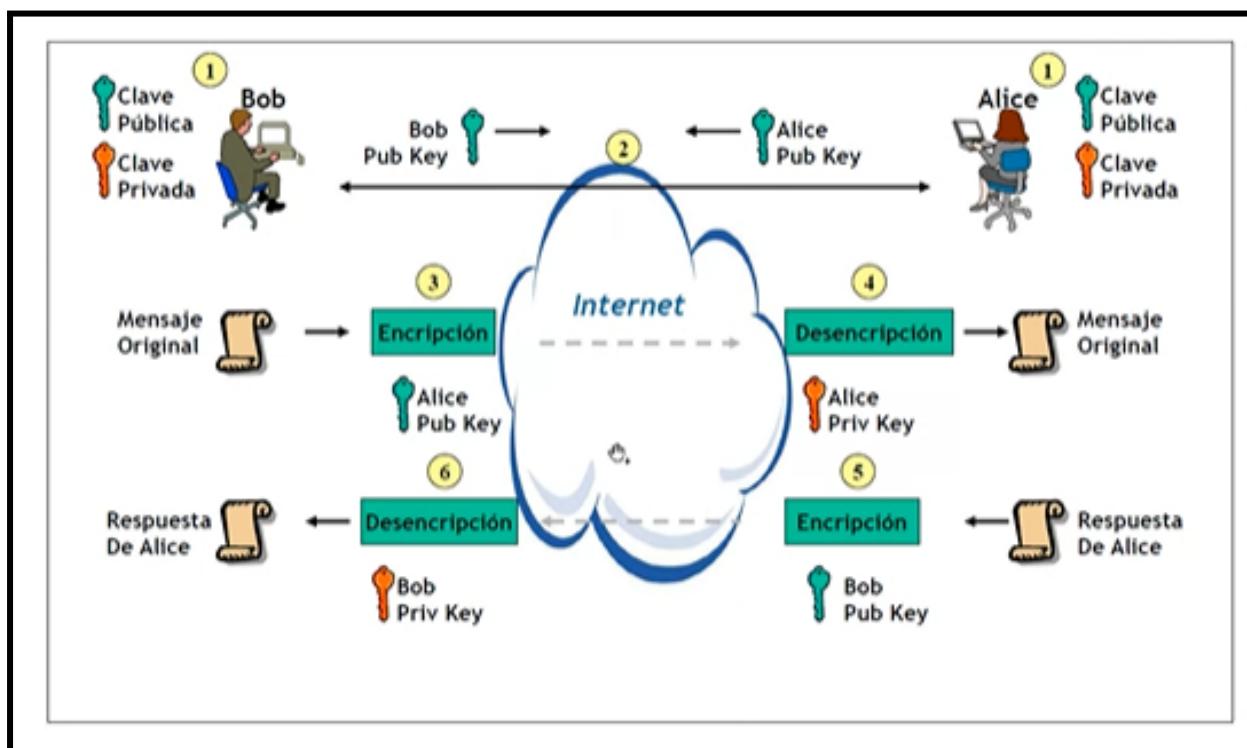
- Integridad de los datos
- Confidencialidad
- No repudio
- Autentificación

Pasos:

- 1- Bob y Alice crean su par de clave privada/pública
- 2- Bob y Alice intercambian las públicas.
- 3- Alice escribe mensajes a Bob y utiliza **la clave pública de Bob** para encriptar mensajes. Y lo envía por internet. Sólo Bob lo puede desencriptar.
- 4- Bob utiliza clave privada para desencriptar mensajes.

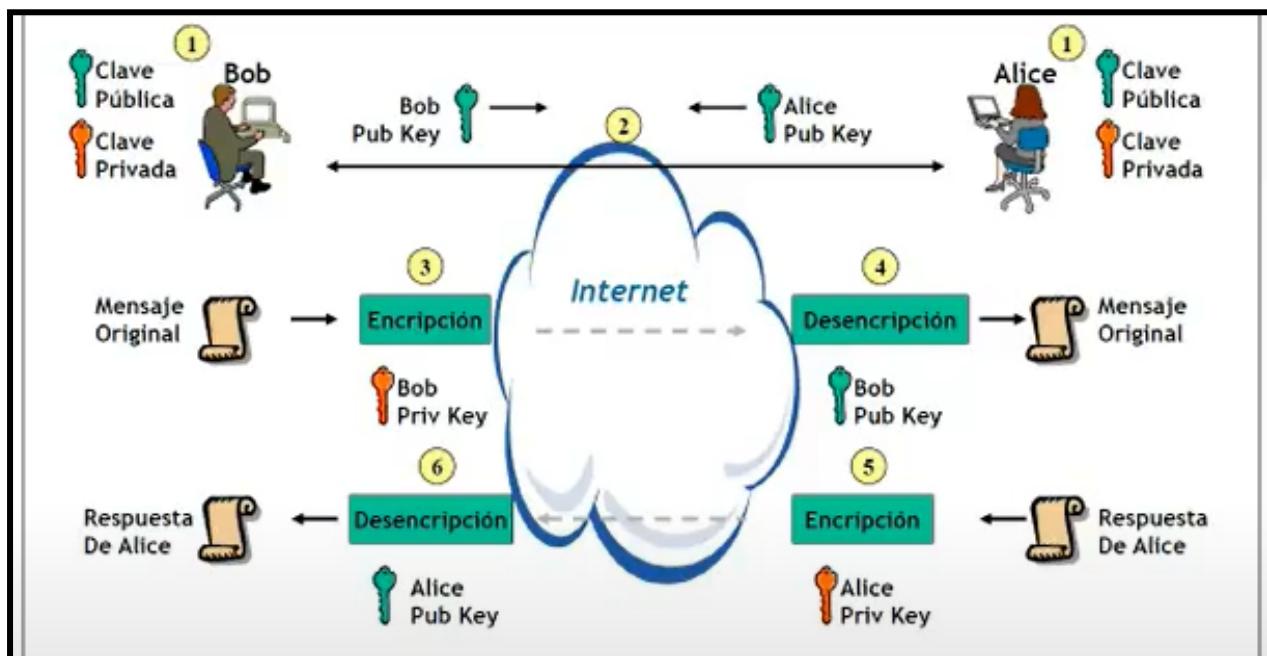
Encriptar el mensaje sólo con la llave pública del receptor:

El problema es que solo necesitas la clave pública de la otra persona, cualquiera podría hacerse pasar por vos. Se garantiza confidencialidad del mensaje (solo receptor lo puede abrir) e integridad (no se modifican los datos si no es con la llave privada del receptor) pero no garantizo autentificación y no repudio.



Encriptar el mensaje con la llave privada del emisor (yo):

Resuelvo autenticación y no repudio pero hay un problema cualquiera lo puede leer. Bob encripta el mensaje con su clave privada, Alice desencripta con clave pública de Bob.



Doble encriptación para solventar estos inconvenientes:

Entonces encriptamos con clave privada propia y clave pública del receptor del mensaje. Alice desencripta con clave pública de Bob y clave privada suya.

Funciones de Hash:

Una función de hash toma una entrada de longitud arbitraria y genera una salida de longitud fija. La salida, de longitud fija, se llama “digesto” o fingerprint.

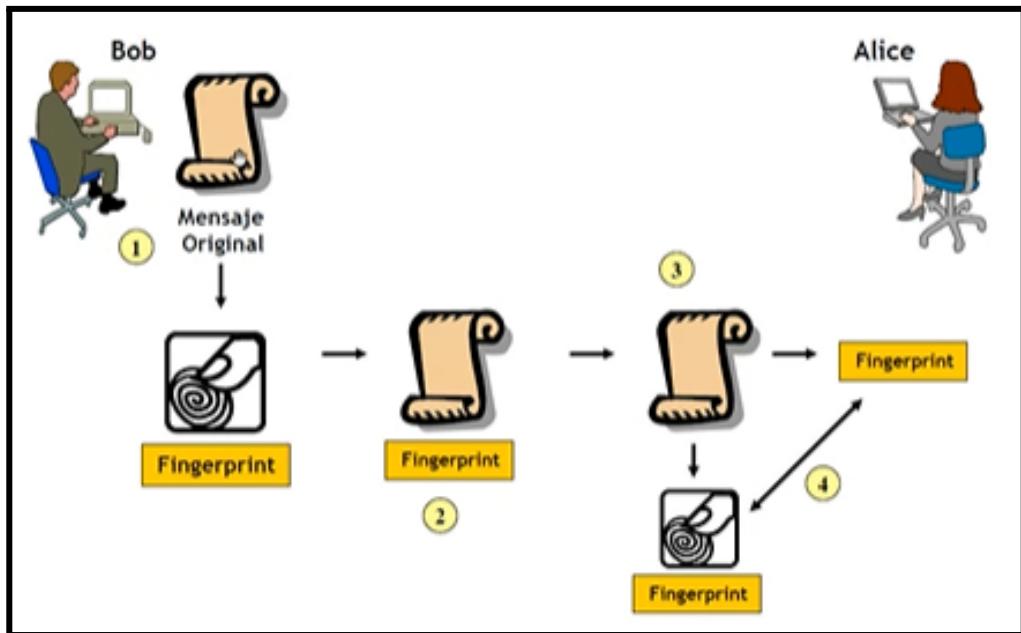
Un algoritmo de hash debe cumplir estos requisitos:

- **Consistencia:** La misma entrada debe generar la misma salida
- **Aleatoriedad:** Que impida adivinar el mensaje original
- **Unicidad:** Debe ser casi imposible encontrar 2 mensajes con mismo digest
- **One way:** Para un digesto yo no pueda deshacer la operación y obtener el mensaje original.

La idea es aplicar el hash al mensaje, si yo modiflico el mensaje y le aplico el hash nuevamente el digesto es diferente, si no se puede corroborar el digesto hay alteración.

Las funciones de Hash más comunes son:

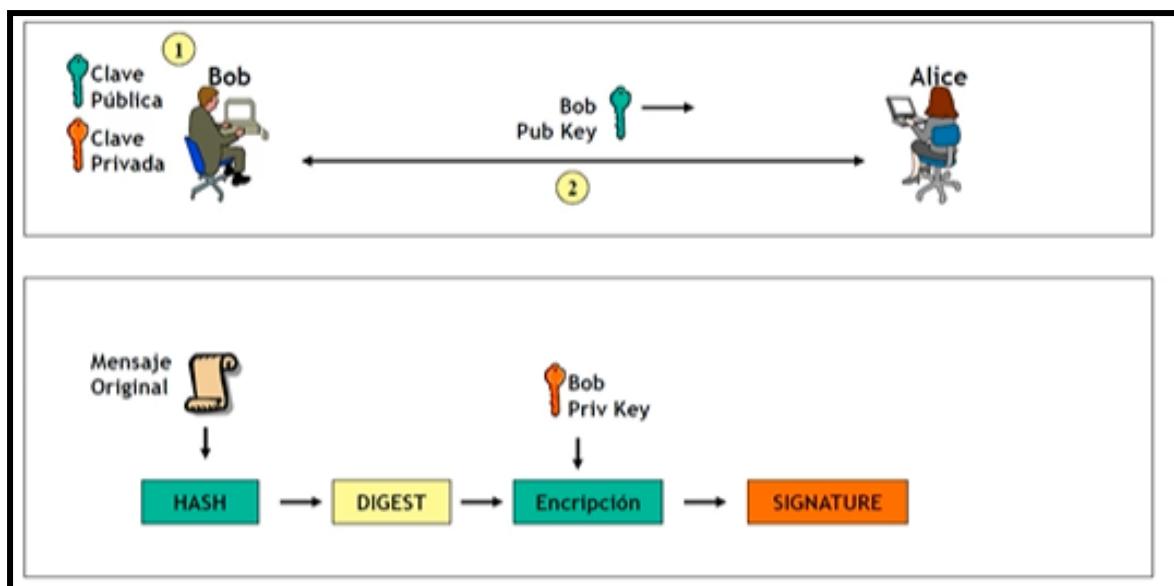
- Message Digest 4 (MD4)
- Message Digest 5 (MD5)
- Secure Hash Algorithm (SHA)



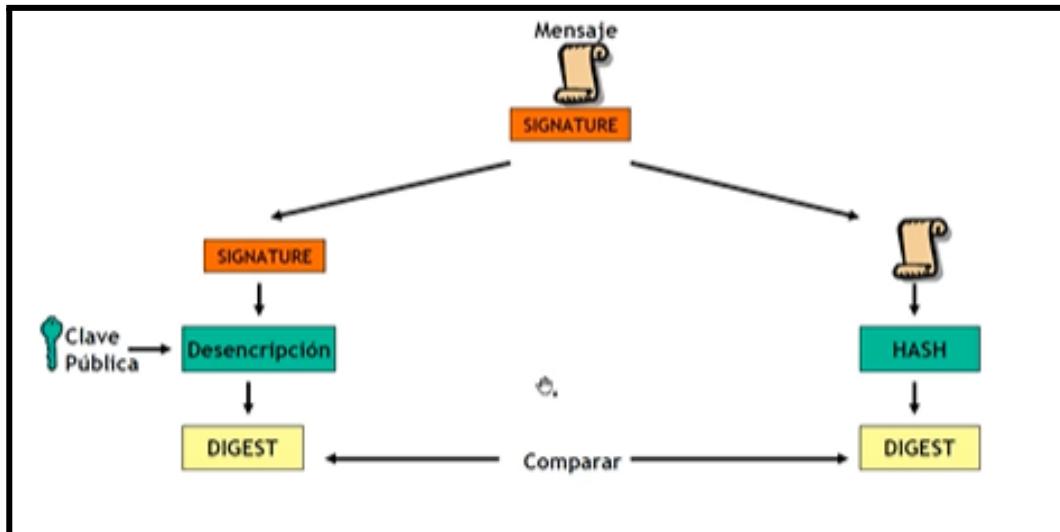
Firma digital:

Es un digesto encriptado que se adiciona a un documento. Da vinculación o no repudio e integridad.

- Confirmar la identidad del emisor
- Garantizar la integridad del documento



Verificación:

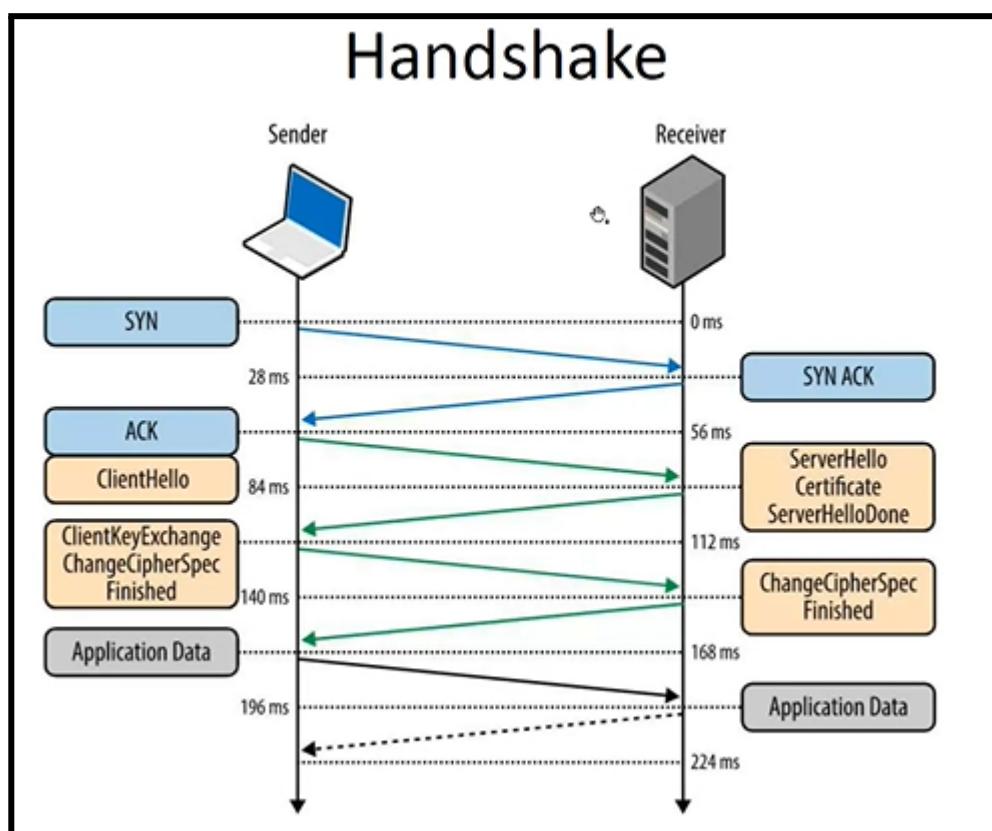


TLS - Transport Layer Security

Se llama SSL (Secure Socket Layer) → evoluciona en TLS.

- Es un protocolo
- Evolución del secure socket layer que está obsoleto hoy.
- Ampliamente utilizado para proteger tráfico web entre un servidor HTTP y un browser
- Utiliza encriptación simétrica (resto de comunicación) y asimétrica (establecimiento de la conexión).
- Puerto 443 HTTPS usa este protocolo.

Cómo funciona:



El sender hace una propuesta de encripción luego que el servidor le envíe su certificado, el servidor aceptará o pedirá cambiar y se establece la conexión. Ej:

Cipher Suite es una lista de algoritmos criptográficos ordenados por orden de preferencia. El servidor elegirá el mayor que pueda soportar. Contiene:

key exchange algorithm – cómo se intercambiarán las claves simétricas

authentication algorithm – cómo se autenticará

bulk encryption algorithm – algoritmo de clave simétrica a utilizar

Message Authentication Code (MAC) – método para chequear integridad.

TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384

TLS - protocol; **ECDHE** key exchange algorithm; **ECDSA** authentication algorithm; **AES_256_CBC** bulk encryption algorithm; and **SHA384** MAC algorithm.

Certificados:

Los presenta el servidor a los clientes.

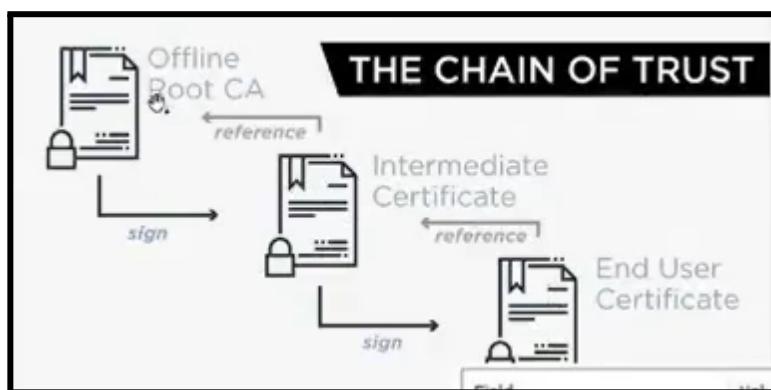
Dice qué servidor es (url) y quién certifica que soy quien digo ser. (entidad certificante).

Entidad certificante:

Genera certificados con un tiempo de duración y luego se vence y debe renovarse. Son entidades confiables para los clientes.

Hay una cadena de seguridad

El certificado que ve el usuario → el certificado intermedio → el certificado root CA (instalado en mi repositorio como confiable, convenios de microsoft o quien sea para ciertas entidades certificantes de una.)



Instalar un certificado:

- Generar un CSR (certificate signing request)
- Cargo los valores que quiera que tengan
- Se genera la clave privada que me guardo
- Ese pedido lo envío a una entidad certificante para comprarlo.

IPV6 - Ipng

RFC 2460

- Surge por escasez de direcciones IP
- Tiene 128 bits de largo.

Algunos proveedores realizan **doble traducción**:

Antes de IPV6 el proveedor asignaba a sus clientes IPs no públicas (como hace nuestro router) y luego las traducía en un gran router (carrier grade).

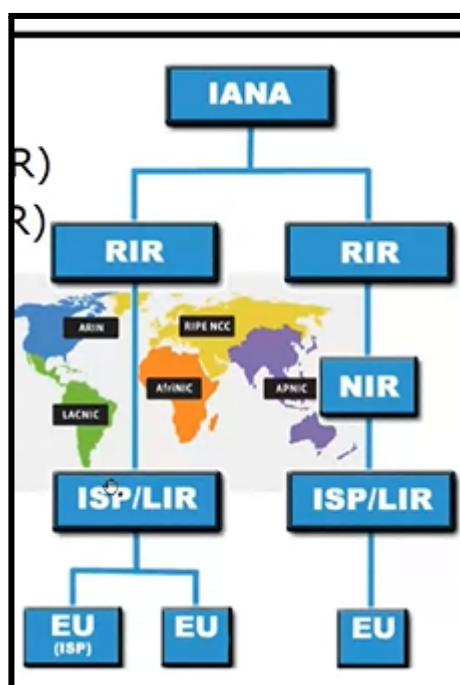
De la IP privada que usamos a una pseudo pública (100. ...) y luego a una IPV6

Ethertype: 0x86dd para IPV6

Administración del espacio:

Surge de IANA

1. Internet registry IR
2. Regional Internet Registry
3. National Internet Registry
4. Local internet



IANA asigna rangos /12 a los RIRs.

LIRs (proveedores de internet) reciben un rango /32 y delegan:

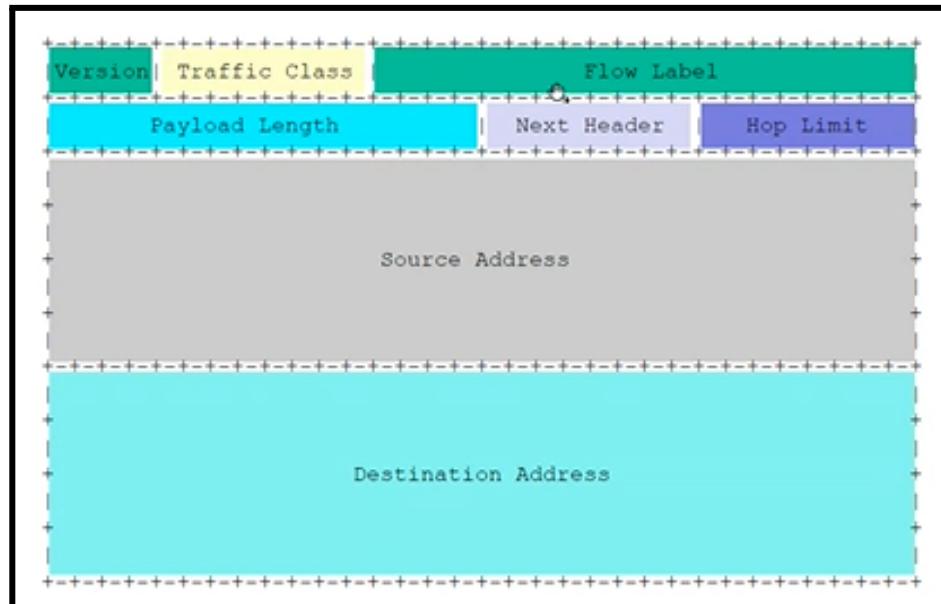
- /48 a usuarios finales
- /64 solo una red
- /128 solo un dispositivo

Cabecera IPV6:

Tiene menos campos que IPV4.

32 bits de ancho, ocupa 40 bytes, el doble que IPV4.

Es la cabecera mínima y tiene cabecera extensible también.



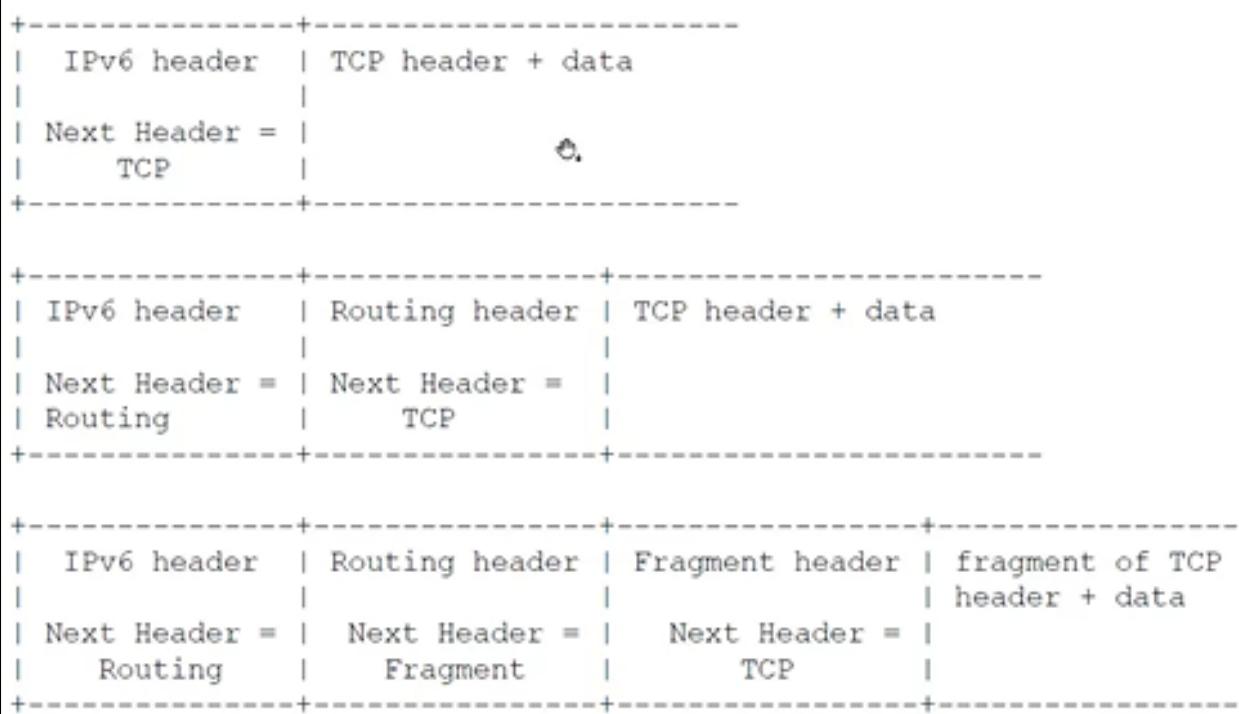
Una dirección ocupa 4 palabras, mucho espacio.

1. Versión: Compatible con el campo versión de IPV4. Valor = 6.
2. Traffic Class: Similar type of service de IPV4.
3. Flow Label:
 - 20 bytes
 - Novedoso para IP pero similar a MPLS.
 - Etiqueta de flujo
 - No tengo necesidad de leer las 2 direcciones para commutar. Tomo decisiones de conmutación mucho más rápidas con una etiqueta.
4. Payload Length: 16 bits, un datagrama IPV6 tiene longitud máxima de 65.535 bytes
5. Next header
(en teoría refiere a que tipo es el prox. header, qué protocolo encapsula)
Indica un puntero al próximo header.
Si encapsula TCP el valor será 6 (el protocolo de IPv4)
Puede ser un protocolo o cabecera extensible.
6. Hop Limit: TTL(time to live) con otro nombre, límites de saltos.
7. Source Address y destination address: 128 bits cada una, 4 palabras cada una.

El prefijo de red es fijo de 64 bits, los otros campos están en la cabecera extensible.

Cabecera extensible:

Introduce una cabecera opcional y extensible adicional a la mínima (arriba)



Routing header:

Cabecera extensible opcional por si necesito agregar información de ruteo no hay options como IPV4.

Fragment header:

Es una cabecera extensible opcional donde encuentro todos los bits que implementaban el mecanismo de fragmentación en IPV4 (más fragmentos, no fragmentar, offset, etc).

Unicast Address Format:

En IPV6 la IP no identifica un host sino que identifica a una interfaz en un host.

| | | |
|-----------------------|-----------|----------------|
| n bits | m bits | $128-n-m$ bits |
| global routing prefix | subnet ID | interface ID |

RFC 3513, Abril 2003

| | | | |
|-----|-----------------------|-----------|--------------|
| 3 | 45 bits | 16 bits | 64 bits |
| 001 | global routing prefix | subnet ID | interface ID |

De forma genérica:

1. Global routing prefix:
2. Subnet ID
3. Interface ID

En la práctica:

Partimos los 128 bits a la mitad, la primera es el **prefijo (64 bits)** y la segunda es el **identificador de la interfaz**.

En la primera mitad:

- Los primeros 48 bits son el global routing prefix. (los 3 primeros inicialmente tenían sentido pero ya no).
- 16 bits para subnet
- los 64 restantes para interface id.

Cada red en IPV6 puede tener muchísimos más hosts que toda la internet en IPV4

La idea de IPV6: Un host se auto configura, elige el identificador de interfaz y no se lo asignan como en IPV4

Sintaxis de la dirección:

Representación Hexadecimal, agrupado de a 16 bits, separados por ":".

2001:0DB8:0000:2F3B:02AA:00FF:FE28:9C5A

Notación con supresión de ceros:

2001:DB8:0:2F3B:2AA:FF:FE28:9C5A

Notación con compresión de ceros:

Por ejemplo, la dirección local **FE80:0:0:0:2AA:FF:FE9A:4CA2**

Se representa como: **FE80::2AA:FF:FE9A:4CA2**

Existen notaciones para hacer más amigable la interfaz, aunque la dirección viaja como 128 bits. Pero al humano es más fácil de leer.

Notación con supresión de ceros

- Hace más amigable la notación.
- En la imagen se usan las dos juntas.

Notación con compresión de ceros

-Donde hay solo ceros se suplanta por ":" y significa "todos ceros", para saber cuántos, es la cantidad necesaria hasta completar los 128 bits

-Solo se puede comprimir una vez

Direcciones notables:

Ya no existen clases sino:

- Loopback = ::/1 127 ceros y un 1 es local host. ::1/128
- Default route = ::/0 el peor match en la tabla de ruteo, va al default gateway

```
Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . : home
IPv6 Address . . . . . : 2800:810:597:831:1cd6:b1d2:f94d:101b
IPv6 Address . . . . . : fdaa:bbcc:ddee:0:1cd6:b1d2:f94d:101b
Temporary IPv6 Address . . . . . : 2800:810:597:831:25a4:3291:cf66:2df5
Temporary IPv6 Address . . . . . : 2800:810:597:831:f507:bdd9:7d1f:46c4
Temporary IPv6 Address . . . . . : fdaa:bbcc:ddee:0:25a4:3291:cf66:2df5
Temporary IPv6 Address . . . . . : fdaa:bbcc:ddee:0:f507:bdd9:7d1f:46c4
Link-local IPv6 Address . . . . . : fe80::1cd6:b1d2:f94d:101b%23
IPv4 Address . . . . . : 192.168.0.14
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.0.1
```

Prefijos IPV6:

Se usa la representación CIDR: Dirección / longitud del prefijo

Dirección/Longitud del prefijo

2001:DB8:2A0:2F3B::/64 es un prefijo de red

2001:DB8:3F::/48 es un prefijo de ruta summarizada

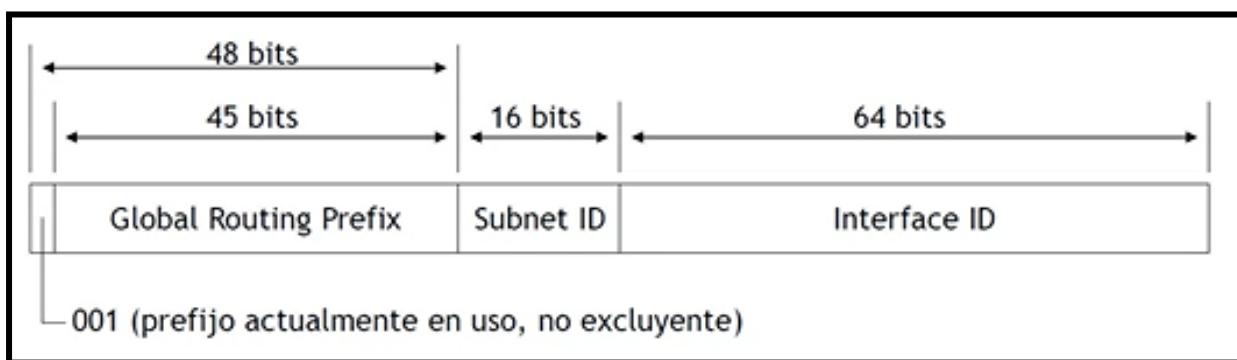
Si una dirección es /64 = Representa una red porque los otros 64 son el identificador de interfaz. No es necesario expresar el prefijo.

Si una dirección es /48 = Es una ruta summarizada o un rango de direcciones que summariza una porción del espacio de direcciones V6.

Tipos de direcciones:

Global Unicast Addresses:

- Son direcciones globales y pueden ser alcanzadas por toda la internet.
- Equivalentes a una dirección IPV4 registrada. A una dirección pública.
- Diseñadas para ser summarizadas: IANA asignó bloques a los RIRS continuos de manera que se puedan agregar/agrupar y tratar a todos de igual manera. Una dirección IPV6 viene nativamente, pertenece a una región global.



En linux, unix y sistemas abiertos el *Interface ID* o identificador de interfaz de 64 bits se utiliza con la MAC address del dispositivo para aprovechar ese número que es único.

Para windows no se hace así, para no exponer esa MAC address ni datos sensibles, se genera un identificador de interfaz dinámico que cambia con el tiempo y es aleatorio.

Local-use Unicast Addresses:

- De uso local para conectarse con vecinos de la red LAN
- Empiezan con prefijo determinado FE80::/64
- Link-Local: Utilizadas entre vecinos on-link y en el proceso de Neighbor discovery
- Site-local: Obsoleto



```
Ethernet adapter VMware Network Adapter VMnet1:
Connection-specific DNS Suffix . . . .
Link-local IPv6 Address . . . . . fe80::483b:644a:8471:8ea7%13
IPv4 Address . . . . . 192.168.203.1
Subnet Mask . . . . . 255.255.255.0
Default Gateway . . . . .
```

```
wireless LAN adapter Wi-Fi:
Connection-specific DNS suffix . . . home
IPv6 Address . . . . . 2800:810:597:831:1cd6:b1d2:f94d:101b
IPv6 Address . . . . . fdaa:bbcc:ddee:0:1cd6:b1d2:f94d:101b
Temporary IPv6 Address . . . . . 2800:810:597:831:25a4:3291:cf66:2df5
Temporary IPv6 Address . . . . . 2800:810:597:831:f507:bdd9:7d1f:46c4
Temporary IPv6 Address . . . . . fdaa:bbcc:ddee:0:25a4:3291:cf66:2df5
Temporary IPv6 Address . . . . . fdaa:bbcc:ddee:0:f507:bdd9:7d1f:46c4
Link-local IPv6 Address . . . . . fe80::1cd6:b1d2:f94d:101b%23
IPv4 Address . . . . . 192.168.0.14
Subnet Mask . . . . . 255.255.255.0
Default Gateway . . . . . 192.168.0.1
```

Mi dispositivo usará la IP de link local entre vecinos de la red LAN y no va a salir a un destino público nunca.

Requiere identificador de zona (zone-identifier - scope id) para evitar ambigüedad. La ambigüedad se genera porque todas mis interfaces tienen una link local, tengo n conexiones a la red fe80 o n redes fe80. Debo indicar qué interfaz quiero usar (Ej: la %23 es la interfaz wireless en la imagen).

Ya no necesito direcciones privadas, tengo mi dirección pública con la que me conecto con cualquier host IPV6. Y tengo mi link local para vecinos.

```

Ethernet adapter VMware Network Adapter VMnet2:

Connection-specific DNS Suffix . . .
Link-local IPv6 Address . . . . . fe80::58f1:904b:18ac:f4%12
Autoconfiguration IPv4 Address. . . . . 169.254.0.244
Subnet Mask . . . . . 255.255.0.0
Default Gateway . . . . .

Ethernet adapter VMware Network Adapter VMnet3:

Connection-specific DNS Suffix . . .
Link-local IPv6 Address . . . . . fe80::895e:3a1b:50:136f%15
Autoconfiguration IPv4 Address. . . . . 169.254.19.111
Subnet Mask . . . . . 255.255.0.0
Default Gateway . . . . .

Ethernet adapter VMware Network Adapter VMnet4:

Connection-specific DNS Suffix . . .
Link-local IPv6 Address . . . . . fe80::f5a8:df38:14de:3cf8%11
Autoconfiguration IPv4 Address. . . . . 169.254.60.248
Subnet Mask . . . . . 255.255.0.0
Default Gateway . . . . .

```

Se tiene una tabla de ruteo para IPV4 y una para IPV6.

Broadcast en IPV6:

En IPV4 mapeo un broadcast todos unos de capa 3 a uno de capa 2 para ethernet con dirección mac destino todos unos.

En IPV6 no existe dirección broadcast todos unos sino una dirección de multicast que indica todos los nodos de la red Dirección ff02.

Al mapear ff02 a ethernet, no se mapea a una MAC destino todos unos porque eso es broadcast y en IPV6 es multicast todos los nodos de la red, entonces la trama encapsulada sólo resultará interesante para dispositivos que corren IPV6.

| No. | Time | Source | Destination | Protocol | Info |
|------|-----------|--------------------------|-------------|----------|---|
| 6643 | 21.516822 | fe80::6e99:61ff:fe7:ccef | ff02::1 | ICMPv6 | Router Advertisement from 6c:99:61:f7:c |

Ej: El router envía a la red una advertencia para todos los nodos que quieran usar IPV6 debe autoconfigurar con el prefijo indicado.

```
Reachable time (ms): 30000
Retrans timer (ms): 0
↳ ICMPv6 Option (Prefix information : 2800:810:597:831::/64)
    Type: Prefix information (3)
    Length: 4 (32 bytes)
    Prefix Length: 64
    Flag: 0xc0, On-link flag(L), Autonomous address-configuration flag(A)
    Valid Lifetime: 2186163
    Preferred Lifetime: 2186163
    Reserved
    Prefix: 2800:810:597:831::
    ↳ ICMPv6 Option (Route Information : Medium 2800:810:597:831::/64)
```

Un ejemplo con IPV6 con ID de interfaz con la MAC address:

Completó la mac address con ff para pasar de 48 bits de la MAC a los 64 requeridos

```
Payload Length: 232
Next Header: ICMPv6 (58)
Hop Limit: 255
Source Address: fe80::6e99:61ff:fef7:ccef
Destination Address: ff02::1
[Source SA MAC: Sagemcom_f7:cc:ef (6c:99:61:f7:cc:ef)]
Internet Control Message Protocol v6
```