

## **REDES DE TRANSMISIÓN DE DATOS**

A veces no es práctico que dos dispositivos de comunicaciones se conecten directamente mediante un enlace punto a punto. Esto es debido a alguna (o a las dos) de las siguientes circunstancias:

- **Los dispositivos están muy alejados.**
- **Hay un conjunto de dispositivos que necesitan conectarse entre ellos en instantes de tiempo diferentes.** Un ejemplo de esta necesidad es la red de teléfonos mundial o el conjunto de computadores pertenecientes a una compañía. Salvo el caso de que el número de dispositivos sea pequeño, no es práctico utilizar un enlace entre cada dos.

La solución a este problema es conectar cada dispositivo a una red de comunicación. Para clasificar las redes tradicionalmente se consideran dos grandes categorías: las redes de área amplia (WAN, Wide Area Networks) y las redes de área local (LAN, Local Area Networks).

### **REDES DE ÁREA AMPLIA (WAN)**

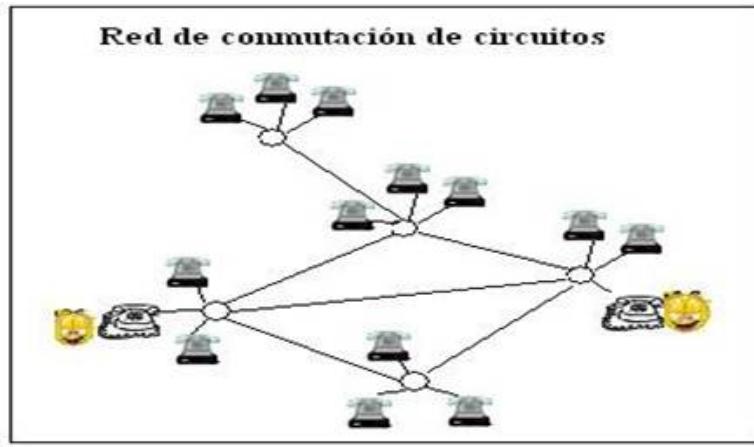
Generalmente, se considera como redes de área amplia a todas aquellas que cubren una **extensa área geográfica**, requieren **atravesar rutas de acceso público** y utilizan, al menos parcialmente, **circuitos proporcionados por una entidad proveedora de servicios de telecomunicación**.

Generalmente, una WAN **consiste en una serie de dispositivos de conmutación interconectados**. La transmisión generada por cualquier dispositivo **se encaminará a través de estos nodos internos hasta alcanzar el destino**. A estos nodos (incluyendo los situados en los contornos) no les concierne el contenido de los datos, al contrario, su función es proporcionar el servicio de conmutación, necesario para transmitir los datos de nodo en nodo hasta alcanzar su destino final.

Tradicionalmente, las WAN se han implementado usando una de las dos tecnologías siguientes: **comutación de circuitos y comutación de paquetes**. Últimamente, se está empleando como solución la técnica de **retransmisión de tramas (frame relay)**, así como las **redes ATM**.

#### Comutación de circuitos

En las redes de comutación de circuitos, para interconectar dos estaciones **se establece un camino dedicado a través de los nodos de la red**. El **camino** es una **secuencia conectada de enlaces físicos entre nodos**. En cada enlace, se dedica un **canal lógico** a cada conexión. Los **datos generados por la estación fuente se transmiten por el camino dedicado** tan rápido como se pueda. En cada nodo, los datos de entrada se encaminan o comutan por el canal apropiado de salida sin retardos. El ejemplo más ilustrativo de la comutación de circuitos es la red de telefonía.



### Comutación de paquetes

Un enfoque diferente al anterior es el adoptado en las redes de conmutación de paquetes. En este caso, **no es necesario hacer una asignación a priori de recursos** (capacidad de transmisión) en el camino (o **sucesión de nodos**). Por el contrario, los **datos se envían en secuencias de pequeñas unidades llamadas paquetes**. Cada paquete se pasa de nodo en nodo en la red siguiendo algún camino entre la estación origen y la destino. En cada nodo, el paquete se recibe completamente, se almacena durante un breve intervalo y posteriormente se retransmite al siguiente nodo. Las redes de conmutación de paquetes se usan fundamentalmente para las **comunicaciones terminal-computador y computador-computador**.

Demostración: [https://commons.wikimedia.org/wiki/File:Packet\\_Switching.gif](https://commons.wikimedia.org/wiki/File:Packet_Switching.gif)

### Retransmisión de tramas (frame relay)

La conmutación de paquetes se desarrolló en la época en la que los servicios de transmisión a larga distancia presentaban una tasa de error relativamente elevada, comparada con los servicios de los que se dispone actualmente. Por tanto, para compensar esos errores relativamente frecuentes, **en los esquemas de conmutación de paquetes se realiza un esfuerzo considerable, que se traduce en añadir información redundante** en cada paquete así como en la realización de un **procesamiento extra**, tanto en el destino final como en los nodos intermedios de conmutación, necesario para detectar los errores y, en su caso, corregirlos.

La tecnología de **retransmisión de tramas** se ha desarrollado teniendo presente que **las velocidades de transmisión disponibles en la actualidad son mayores, así como que las tasas de error actuales son menores**. Mientras que las redes originales de conmutación de paquetes se diseñaron para ofrecer una velocidad de transmisión al usuario final de 64 kbps, las redes con **retransmisión de tramas** están **diseñadas para operar eficazmente a velocidades de transmisión de usuario de hasta 2 Mbps**. La clave para conseguir estas velocidades **reside en eliminar la mayor parte de la información redundante** usada para el control de errores y, en consecuencia, el procesamiento asociado.

Las tramas son de **longitud variable**.

## ATM

El Modo de Transferencia Asíncrono (ATM, Asynchronous Transfer Mode), a veces denominado como modo de retransmisión de celdas (cell relay) se puede considerar como una **evolución de la retransmisión de tramas**.

La diferencia más obvia entre retransmisión de tramas y ATM es que la primera usa paquetes de longitud variable, llamados «tramas», y **ATM usa paquetes de longitud fija denominados «celdas»**. Al igual que en retransmisión de tramas, **ATM introduce poca información adicional para el control de errores**, confiando en la inherente robustez del medio de transmisión así como en la lógica adicional localizada en el sistema destino para detectar y corregir errores. Al utilizar paquetes de longitud fija, **el esfuerzo adicional de procesamiento se reduce incluso todavía más que en retransmisión de tramas**. El resultado es que ATM se ha diseñado para **trabajar a velocidades de transmisión del orden de 10 a 100 Mbps, e incluso del orden de Gbps**.

ATM se puede considerar, a su vez, como una **evolución de la conmutación de circuitos**. En la **conmutación de circuitos se dispone solamente de circuitos a velocidad fija** de transmisión entre los sistemas finales. **ATM permite la definición de múltiples canales virtuales con velocidades de transmisión que se definen dinámicamente** en el instante en el que se crea el canal virtual. Al utilizar celdas de tamaño fijo, ATM es tan eficaz que puede ofrecer un canal a **velocidad de transmisión constante aunque esté usando una técnica de conmutación de paquetes**. Por tanto, en este sentido, ATM es una generalización de la conmutación de circuitos en la que se **ofrecen varios canales, en los que la velocidad de transmisión se fija dinámicamente para cada canal según las necesidades**.

## REDES DE ÁREA LOCAL

Es una red de comunicaciones que **interconecta varios dispositivos y proporciona un medio para el intercambio de información entre ellos**. Hay algunas **diferencias entre las redes WAN y LAN**:

- La **cobertura de una LAN es pequeña**, generalmente un edificio o, a lo sumo, un conjunto de edificios próximos.
- Es habitual que la LAN sea propiedad de la misma entidad propietaria de los **dispositivos conectados a la red**. En WAN, esto no es tan habitual o, al menos, una fracción significativa de recursos de la red son ajenos. Esto tiene dos implicaciones. La primera es que se debe cuidar mucho la elección de la LAN, ya que, evidentemente, **Ileva acarreada una inversión sustancial de capital** (comparada con los gastos de conexión o alquiler de líneas en redes de área amplia) tanto en la adquisición como en el mantenimiento. Segunda, la **responsabilidad de la gestión de la red local recae solamente en el usuario**.
- Por lo general, las **velocidades de transmisión internas** en una LAN son **mucho mayores** que en una WAN.

Para las LAN hay muy diversas **configuraciones**. De entre ellas, las más habituales son las **LAN conmutadas** y las **LAN inalámbricas**. Dentro de las conmutadas, las más populares son las **LAN Ethernet**, constituidas por un único conmutador, o, alternativamente, implementadas mediante un conjunto de conmutadores interconectados entre sí. Otro ejemplo muy relevante son las **LAN ATM**, caracterizadas por utilizar tecnología de red ATM en un entorno local. Por último, son también destacables las **LAN con canal de fibra** (Fiber Channel).

## **RED DE ÁREA METROPOLITANA**

El principal mercado para las MAN lo constituyen aquellos clientes que necesitan **alta capacidad en un área metropolitana (menor/mayor alcance que las WAN/LAN y mayor/menor velocidad que las WAN/LAN)**. Las MAN están concebidas para satisfacer estas **necesidades de capacidad a un coste reducido y con una eficacia mayor** que la que se obtendría mediante una compañía local de telefonía para un servicio equivalente.

## **TOPOLOGÍA**

En el contexto de una red de comunicaciones, el término topología se refiere a la **forma según la cual se interconectan entre sí los puntos finales, o estaciones, conectados a la red**.

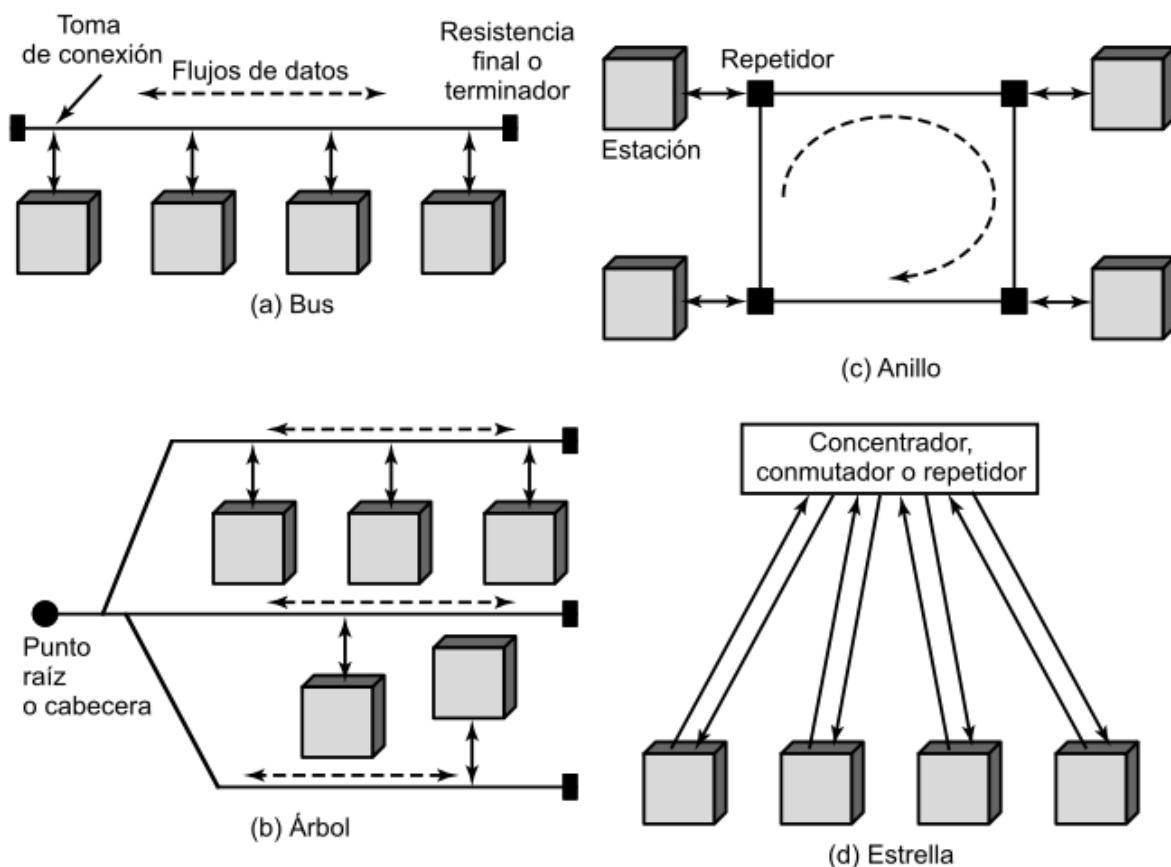


Figura 15.2. Topologías LAN.

### Bus y árbol

Ambas topologías **se caracterizan por el uso de un medio multipunto**, el **bus es un caso especial de la topología en árbol**, con un solo tronco y sin ramas.

En el caso de la topología en bus, **todas las estaciones se encuentran directamente conectadas, a través de interfaces físicas apropiadas conocidas como tomas de conexión (taps), a un medio de transmisión lineal o bus**. El funcionamiento **full-duplex** entre la estación y la toma de conexión permite la transmisión y la recepción de datos a través del bus. Una **transmisión desde cualquier estación se propaga a través del**

**medio** en ambos sentidos y es **recibida por el resto de estaciones**. En cada extremo del bus existe un terminador que absorbe las señales, eliminándolas del bus.

En la topología **árbol**, el medio de transmisión es un **cable ramificado sin bucles cerrados que comienza en un punto conocido como raíz o cabecera (headend)**. Uno o más cables comienzan en el punto raíz y cada uno de ellos puede presentar ramificaciones. **Las ramas pueden disponer de ramas adicionales**, dando lugar a esquemas más complejos. De nuevo, la **transmisión desde una estación se propaga a través del medio y puede alcanzar al resto de estaciones**.

### Problemas que se deben resolver:

- Dado que la transmisión desde una estación se puede recibir en las demás estaciones, es necesario algún método para **indicar a quién va dirigida la transmisión**.
- Se precisa un **mecanismo para regular la transmisión**. Para ver la razón de este hecho hemos de comprender que si **dos estaciones intentan transmitir simultáneamente, sus señales se superpondrán y serán erróneas**; también se puede considerar la situación en que una **estación decide transmitir continuamente durante un largo periodo de tiempo**.

Para solucionar estos problemas, **las estaciones transmiten datos en bloques pequeños llamados tramas**. Cada trama consta de una **porción de los datos que una estación desea transmitir además de una cabecera** de trama que **contiene información de control**. A cada estación en el bus se le asigna una dirección, o identificador, única, incluyéndose en la cabecera la dirección destino de la trama. Ademas, **las tramas se transmiten por turnos** de acuerdo con alguna forma cooperativa.

En la topología en bus o en árbol **no son necesarias acciones especiales para eliminar tramas del medio**: cuando una señal alcanza el final de éste, es **absorbida por el terminador**.

### Anillo

En la topología en anillo, **la red consta de un conjunto de repetidores unidos por enlaces punto a punto formando un bucle cerrado**. El repetidor es un dispositivo relativamente simple, capaz de recibir datos a través del enlace y de transmitirlos, bit a bit, a través del otro enlace tan rápido como son recibidos. **Los enlaces son unidireccionales**; es decir, los datos se transmiten sólo en un sentido, de modo que éstos **circulan alrededor del anillo en el sentido de las agujas del reloj o en el contrario**.

**Cada estación se conecta a la red mediante un repetidor**, transmitiendo los datos hacia la red a través de él. Como en el caso de las topologías en bus y en árbol, **los datos se transmiten en tramas**. Una trama que circula por el anillo pasa por las demás estaciones, de modo que la estación de destino reconoce su dirección y copia la trama, mientras ésta la atraviesa, en una memoria temporal local. **La trama continúa circulando hasta que alcanza de nuevo la estación origen, donde es eliminada del medio**. Dado que el anillo es compartido por varias estaciones **se necesita una técnica de control de acceso al medio** para determinar cuándo puede insertar tramas cada estación.

## Estrella

En redes LAN con topología en estrella cada **estación está directamente conectada a un nodo central común**, generalmente a través de dos enlaces punto a punto, uno para transmisión y otro para recepción.

En general, existen dos alternativas para el **funcionamiento del nodo central**.

- Una es el funcionamiento en **modo de difusión**, en el que la transmisión de **una trama por parte de una estación se retransmite sobre todos los enlaces de salida del nodo central**. En este caso, aunque la disposición física es una estrella, lógicamente **funciona como un bus**: una transmisión desde cualquier estación es recibida por el resto de estaciones, y **sólo puede transmitir una estación en un instante de tiempo dado**. En tal caso, al dispositivo central se le conoce como **concentrador (hub)**.
- Funcionamiento del **nodo central como dispositivo de conmutación de tramas**. Una trama entrante se almacena temporalmente en el nodo y se **retransmite sobre un enlace de salida hacia la estación de destino**.

## Híbrida

Combinación de las anteriores.

# **ARQUITECTURA DE PROTOCOLOS**

Una arquitectura de protocolos es una **estructura en capas de elementos hardware y software que facilitan el intercambio de datos entre sistemas** y posibilita aplicaciones distribuidas, como el comercio electrónico y la transferencia de archivos.

En una arquitectura de protocolos, **los distintos módulos se disponen formando una pila vertical**. Cada capa de la pila realiza el **subconjunto de tareas** relacionadas entre sí que son necesarias para comunicar con el otro sistema.

Evidentemente, para que haya comunicación se necesitan dos entidades, por lo que **debe existir el mismo conjunto de funciones en capas en los dos sistemas**. La comunicación se consigue haciendo que **las capas correspondientes, o pares, intercambien información**. Las capas pares se comunican intercambiando bloques de datos que **verifican una serie de reglas o convenciones** denominadas **protocolo**.

Los aspectos clave que definen o **caracterizan a un protocolo** son:

- **La sintaxis**: establece cuestiones relacionadas con el formato de los bloques de datos.
- **La semántica**: incluye información de control para la coordinación y la gestión de errores.
- **La temporización**: considera aspectos relativos a la sintonización de velocidades y secuenciación.

## **MODELO OSI**

Es una abstracción que constituye una norma de la ISO (Organismo de Estandarización Internacional). Es un **modelo de referencia basado en capas** (modularidad), donde las capas agrupan funciones para permitir la comunicación entre sistemas abiertos y heterogéneos. **Es un estándar de modelo de red que ayuda a los diseñadores de red a implementar redes que pudieran comunicarse y trabajar en conjunto.**

**OSI es un modelo lógico, no físico.** Es esencialmente un conjunto de directrices que los desarrolladores pueden utilizar para crear e implementar aplicaciones para ejecutar en una red. También proporciona un marco para crear e implementar estándares de redes, dispositivos y esquemas de interconexión.

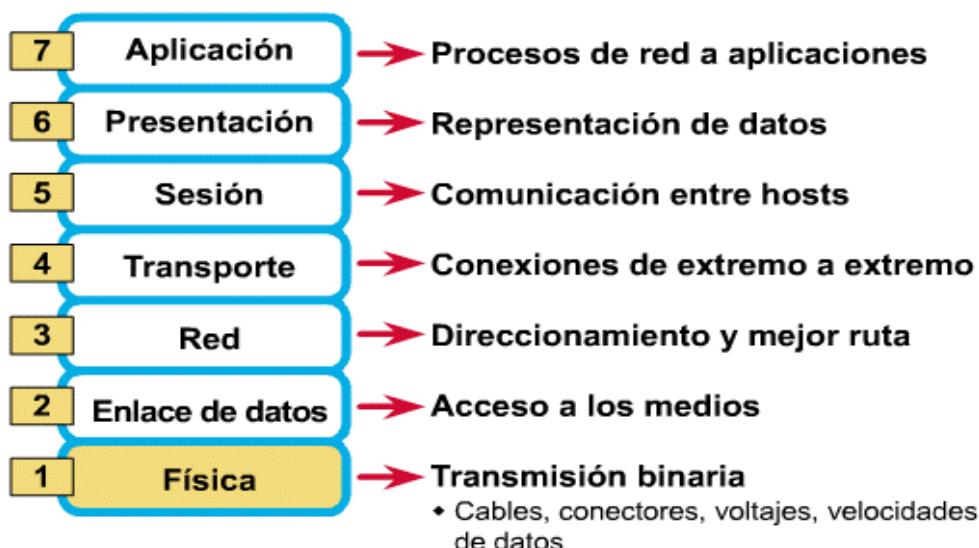
El OSI tiene siete capas diferentes, divididas en dos grupos. **Las tres capas superiores definen cómo las aplicaciones dentro de las estaciones finales se comunicarán entre sí, así como con los usuarios. Las cuatro capas inferiores definen cómo se transmiten los datos de extremo a extremo.**

Comprender que los usuarios interactúan con la computadora en la capa de aplicación y también que las capas superiores son responsables de las aplicaciones que se comunican entre los hosts. **Ninguna de las capas superiores sabe nada sobre redes o direcciones de red, porque esa es la responsabilidad de las cuatro capas inferiores.**

**Exceptuando la capa física, no hay comunicación directa entre las capas pares** (es decir, misma capa del dispositivo origen y destino). Esto es, por encima de la capa física, cada entidad de protocolo pasa los datos hacia la capa inferior contigua, para que ésta los envíe a su entidad par. Es más, el modelo OSI **no requiere que los dos sistemas estén conectados directamente, ni siquiera en la capa física.** Por ejemplo, para proporcionar el enlace de comunicación se puede utilizar una red de conmutación de paquetes o de conmutación de circuitos.

El modelo permite la implementación parcial (no todas las capas).

## **Las 7 capas del modelo OSI**



## Funciones de cada capa

Cada capa individual del modelo OSI tiene un conjunto de funciones que debe realizar para que los paquetes de datos puedan viajar en la red desde el origen hasta el destino. A continuación, presentamos una breve descripción de cada capa del modelo de referencia OSI tal como aparece en la figura.

**La comunicación entre capas iguales se da a través de protocolos, mientras que entre capas adyacentes se utilizan interfaces.**

**Los servicios son provistos por la capa inferior a la superior.**

### Capa 7 - La capa de aplicación

La capa de aplicación es la capa del modelo OSI **más cercana al usuario; suministra servicios de red a las aplicaciones del usuario**. Difiere de las demás capas debido a que **no proporciona servicios a ninguna otra capa OSI, sino solamente a aplicaciones que se encuentran fuera del modelo OSI**.

Algunos ejemplos de aplicaciones son los programas de hojas de cálculo, de procesamiento de texto y los de las terminales bancarias.

La capa de aplicación establece la disponibilidad de los potenciales socios de comunicación, sincroniza y establece acuerdos sobre los procedimientos de recuperación de errores y control de la integridad de los datos.

Funciones:

- Definición de un terminal virtual para permitir el dialogo entre terminales incompatibles
- Proporcionar **interfaz de usuario**, dándole **acceso al entorno OSI**.
- Proporciona servicios de información distribuida.
- Establece **autorizaciones**.
- **Autenticidad** de datos.
- Determinación de la disponibilidad actual.
- A esta capa pertenecen las aplicaciones de uso general: transferencia de archivos, correo electrónico, acceso a terminales remotos, etc.
- Da la **interfaz al software de la aplicación**.

### Capa 6 - La capa de presentación

La capa de presentación **garantiza que la información que envía la capa de aplicación de un sistema pueda ser leída por la capa de aplicación de otro. De ser necesario, la capa de presentación traduce entre varios formatos de datos utilizando un formato común**.

Servicios que provee:

- **Codificación** de datos.
- Manejo de **abstracciones y conversiones**: adaptación de diferentes códigos utilizados por los extremos.

- **Compresión y criptografía.**

### **Funciones:**

- Permite la **comunicación entre equipos con distintas representaciones.**
- **Adecua sintaxis.**
- No necesariamente entiende de semántica.

### Capa 5 - La capa de sesión

Como su nombre lo implica, la capa de sesión **establece, administra y finaliza las sesiones entre dos hosts (aplicaciones) que se están comunicando.** La capa de sesión proporciona sus servicios a la capa de presentación. También **sincroniza el diálogo entre las capas de presentación de los dos hosts y administra su intercambio de datos.** Además de regular la sesión, la capa de sesión ofrece disposiciones para una **eficiente transferencia de datos, clase de servicio y un registro de excepciones** acerca de los problemas de la capa de sesión, presentación y aplicación.

### **Servicios:**

- Gestionar el **control de dialogo**: solicitudes de canales simultáneos (full-duplex) o alternados (half-duplex).
- Recuperación: procedimientos de puntos de comprobación para **recuperación de fallos e interrupción de operaciones.**
- **Sincronización y administración del testigo.**

### **Funciones:**

- **Establecimiento y liberación de conexión.**
- Usuarios de distintas maquinas establezcan conexión.
- **Mejorar servicios.**

### Capa 4 - La capa de transporte

La capa de transporte **segmenta los datos originados en el host emisor y los reensambla en una corriente de datos dentro del sistema del host receptor.** Mientras que las capas de **aplicación, presentación y sesión** están relacionadas con **asuntos de aplicaciones**, las **cuatro capas inferiores se encargan del transporte de datos.**

Está encargado de la **transferencia libre de errores** de los datos entre el emisor y el receptor, aunque no estén directamente conectados, así como de **mantener el flujo de la red.**

### **Servicios**

- **Conexión extremo a extremo sin errores.**
- **Calidad de funcionamiento** (calidad de servicio) solicitada por la capa de sesión.

### **Funciones**

- **Multiplexión.**
- **Regular flujo de datos.**
- El servicio de transporte **orientado a conexión asegura que los datos se entregan libres de errores, en orden y sin pérdidas ni duplicaciones.**
- **TCP:**
  - Permite **colocar los segmentos nuevamente en orden** cuando vienen del protocolo IP.
  - Permite el **monitoreo del flujo de los datos** y así **evita la saturación de la red.**
  - Permite que los **datos se formen en segmentos de longitud variada** para "entregarlos" al protocolo IP.
  - Permite multiplexar los datos, es decir, que la **información que viene de diferentes fuentes (por ejemplo, aplicaciones)** en la misma línea pueda circular simultáneamente.

Ej: Protocolos TCP, SPX

### Capa 3 - La capa de red

La capa de red es una capa compleja que **proporciona conectividad y selección de ruta entre dos sistemas de hosts que pueden estar ubicados en redes geográficamente distintas**. Si desea recordar la Capa 3 en la menor cantidad de palabras posible, piense en **selección de ruta, direccionamiento y enrutamiento.**

#### **Servicios:**

- Servicio **orientado a la conexión o sin conexión a la capa 4**

#### **Funciones:**

- **Encaminamiento.**
- Funciones de **conmutación.**
- Oculta a las capas superiores los detalles de la red subyacente (paquetes / circuitos).
- **Gestión de prioridades.**
- **Interconexión de redes.**
- **Tratamiento de congestión y facturación.**
- **Reenvío por sistemas intermedios.**
- **Interconexión de redes heterogéneas.**
- **IP: fragmentación y reensamblado.**

Ej: Protocolos IP, IPX, X.25

### Capa 2 - La capa de enlace de datos

La capa de enlace de datos **proporciona tránsito de datos confiable a través de un enlace físico**. Al hacerlo, la capa de enlace de datos **se ocupa del direccionamiento**

**físico (comparado con el lógico), la topología de red, el acceso a la red, la notificación de errores, entrega ordenada de tramas y control de flujo.** Si desea recordar la Capa 2 en la menor cantidad de palabras posible, piense en tramas y control de acceso al medio.

#### Servicios:

- **Establecer, mantener y liberar conexiones de la capa 3.**

#### Funciones:

- **Detección y corrección de errores.**
- **Control de flujo de datos.**
- Delimitar secuencia de bits, asegurando **transparencia**.
- **Recuperación de datos perdidos, duplicados o erróneos.**

Ej: Protocolo HDLC, LAP-B, PPP.

### Capa 1 - La capa física

La capa física se encarga de la **interfaz física entre los dispositivos**. Además, define las **reglas que rigen en la transmisión de los bits**. La capa física tiene cuatro características importantes:

- **Mecánicas:** relacionadas con las **propiedades físicas de la interfaz** con el medio de transmisión. Normalmente, dentro de estas características se incluye la especificación del **conector** que transmite las señales a través de conductores. A estos últimos se les denominan **circuitos**.
- **Eléctricas:** especifican **cómo se representan los bits** (por ejemplo, en términos de niveles de tensión), así como su **velocidad de transmisión**.
- **Funcionales:** especifican las **funciones que realiza cada uno de los circuitos** de la interfaz física entre el sistema y el medio de transmisión.
- **De procedimiento:** especifican la **secuencia de eventos que se llevan a cabo en el intercambio** del flujo de bits a través del medio físico.

#### Servicios:

- **Conexión física al medio transmisor**

#### Funciones:

- **Definición de características mecánicas, eléctricas, funcionales y de procedimientos**

Ej: Interfaz RS 232

La **cantidad mínima de capas** para comunicar dos extremos es dos (capa física y capa de enlace, considerando que los dispositivos están conectados físicamente). Si se quieren comunicar tres o mas extremos, se necesita tres capas (las anteriores mas la capa de red).

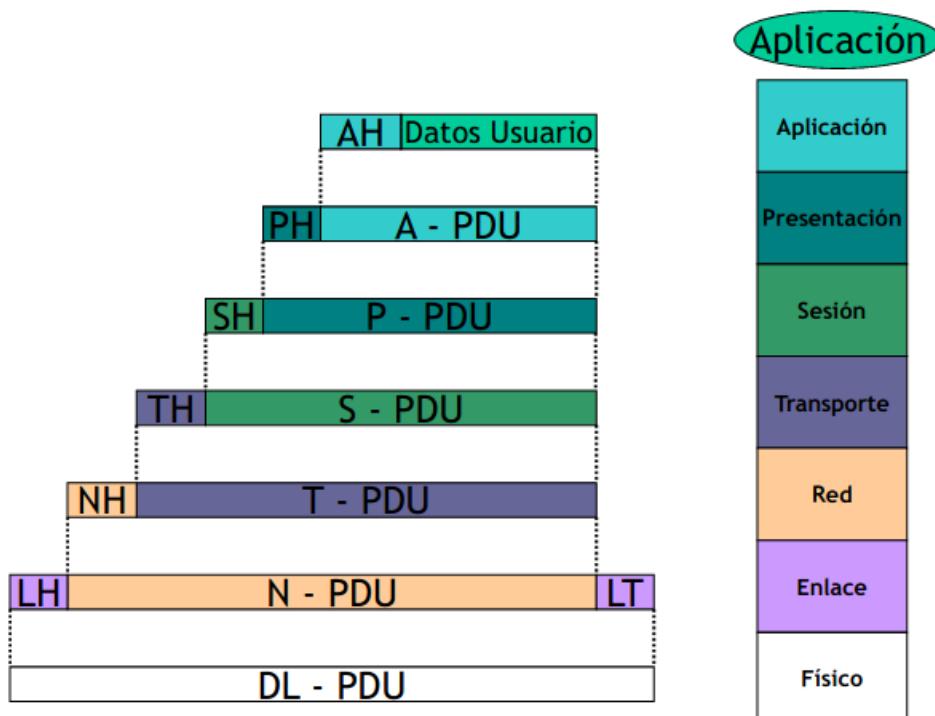
## Unidad de datos de protocolo (PDU)

La figura muestra cómo se usan las unidades de datos de protocolo (PDU) en la arquitectura OSI. **El PDU de cada capa es la combinación del PDU de la capa anterior (o los datos del usuario si es la capa aplicación) y el header de la capa (y una cola si es la de enlace).**

**Encapsulamiento** es el proceso de abstracción que realiza una capa inferior al recibir un paquete de la capa superior. Ese paquete (datos + header capa superior) lo pone en su campo datos (como si fueran todo datos, sin reconocer el header de la capa superior) y le agrega el header de su capa.

En primer lugar, considérese la forma más habitual de implementar un protocolo. Cuando la aplicación X tiene un mensaje para enviar a la aplicación Y, transfiere estos datos a una entidad de la capa de aplicación. A los datos se les añade una cabecera que contiene información necesaria para el protocolo de la capa 7 (**encapsulado**). Seguidamente, los datos originales más la cabecera se pasan como una unidad a la capa 6. La entidad de presentación trata la unidad completa como si de datos se tratara y le añade su propia cabecera (**un segundo encapsulado**). Este proceso continúa hacia abajo hasta llegar a la capa 2, que normalmente añade una cabecera y una cola. La unidad de datos de la capa 2, llamada trama, se pasa al medio de transmisión mediante la capa física. En el destino, al recibir la trama, ocurre el proceso inverso. Conforme los datos ascienden, cada capa elimina la cabecera más externa, actúa sobre la información de protocolo contenida en ella y pasa el resto de la información hacia la capa inmediatamente superior.

El protocolo se implementa en la cabecera del protocolo. Son bits auxiliares que se agregan al mensaje que le permite dialogar con la misma capa en el otro extremo (extremo receptor). La cabecera contiene por ejemplo con qué algoritmo está encriptado.



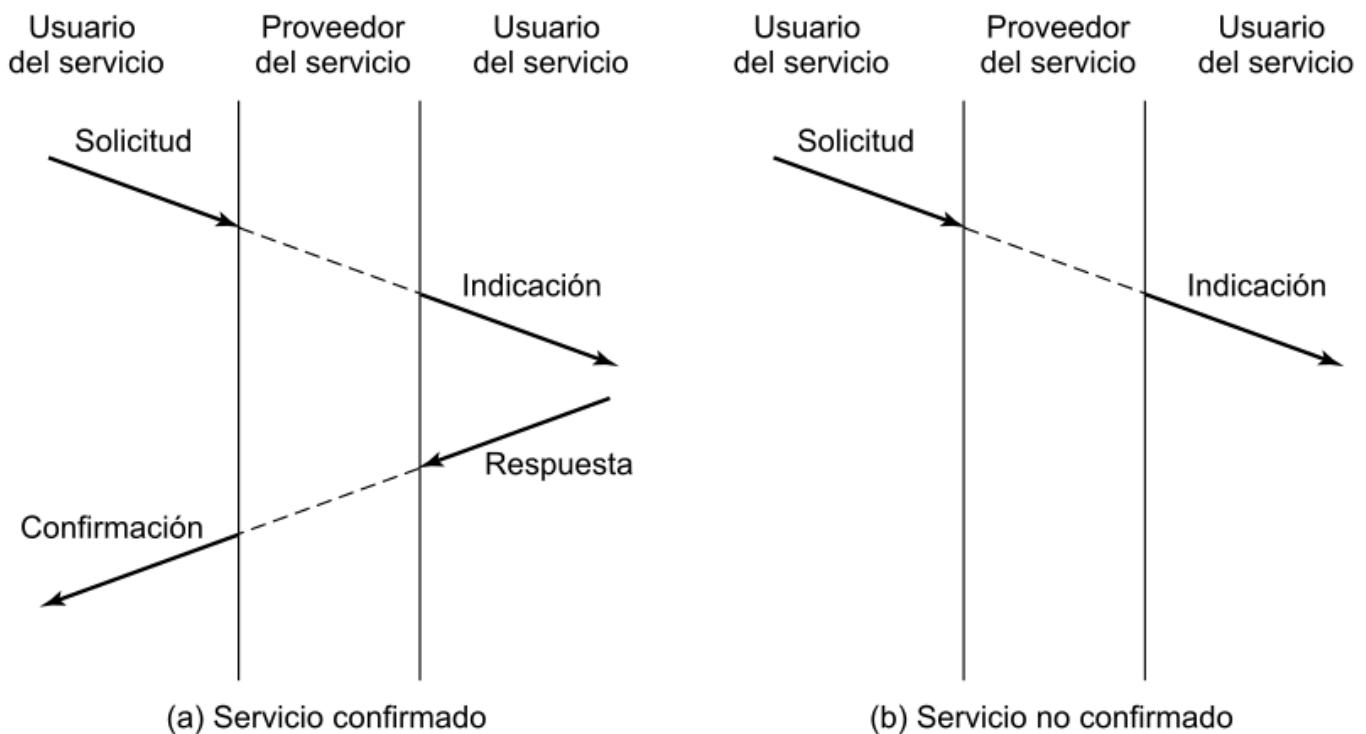
## Parámetros y primitivas de servicio

En la arquitectura OSI los servicios entre capas adyacentes se describen en términos de primitivas y mediante los parámetros involucrados. Una primitiva especifica la función que se va a llevar a cabo y los parámetros se utilizan para pasar datos e

**información de control.** La forma concreta que adopte la primitiva dependerá de la implementación. Un ejemplo es una llamada a un procedimiento.

Para definir las interacciones entre las capas adyacentes de la arquitectura se utilizan **cuatro tipos de primitivas**:

- **Solicitud:** Primitiva emitida por el usuario del servicio para invocar algún servicio y pasar los parámetros necesarios para especificar completamente el servicio solicitado.
- **Indicación:** Primitiva emitida por el proveedor del servicio para:
  - Indicar que ha sido invocado un procedimiento por el usuario de servicio para la conexión y para suministrar los parámetros asociados.
  - Notificar al usuario del servicio una acción iniciada por el suministrador.
- **Respuesta:** Primitiva emitida por el usuario del servicio para confirmar o completar algún procedimiento invocado previamente mediante una indicación a ese usuario.
- **Confirmación:** Primitiva emitida por el proveedor del servicio para confirmar o completar algún procedimiento invocado previamente mediante una solicitud por parte del usuario del servicio.



**Figura 2.10.** Diagramas temporales de las primitivas de servicio.

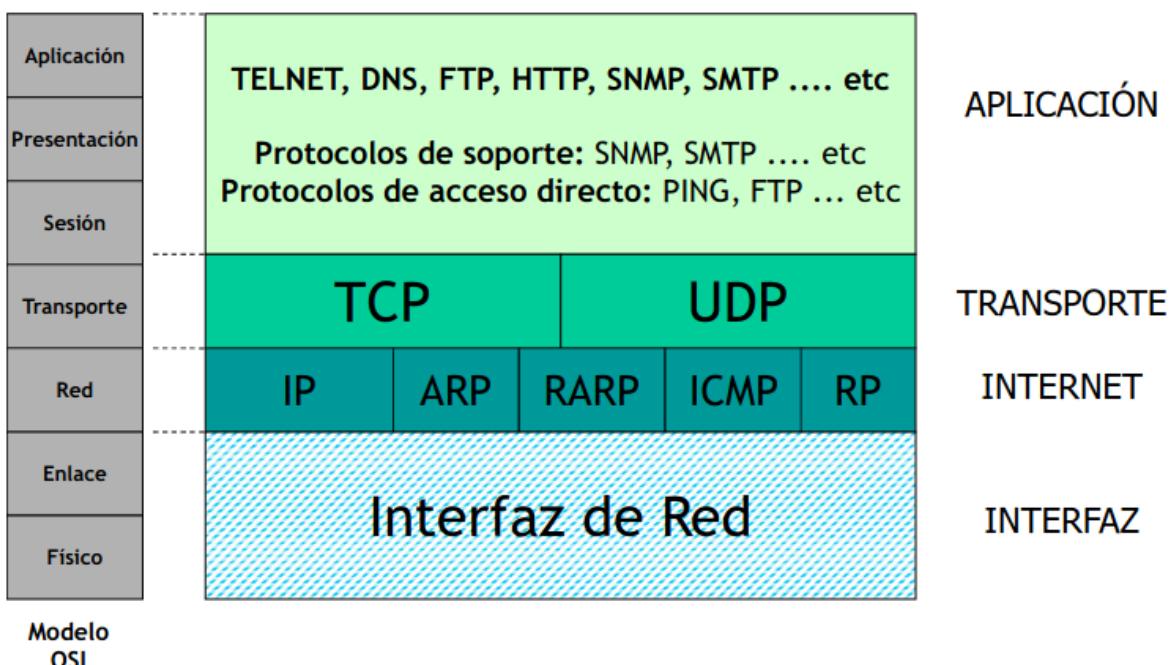
1. La entidad origen (N) invoca a su entidad (N -1) con una primitiva de solicitud. Asociados a esta primitiva están los parámetros necesarios, como por ejemplo, los datos que se van a transmitir y la dirección destino.
2. La entidad origen (N-1) prepara una PDU (N-1) para enviársela a su entidad par (N-1).

3. La entidad destino (N - 1) entrega los datos al destino apropiado (N) a través de la primitiva de indicación, que incluye como parámetros los datos y la dirección origen.
4. Si se requiere una confirmación, la entidad destino (N) emite una primitiva de respuesta a su entidad (N-1).
5. La entidad (N- 1) convierte la confirmación en una PDU (N – 1).
6. La confirmación se entrega a la entidad (N) a través de una primitiva de confirmación.

## Arquitectura TCP/IP

Significa **transmission control protocol / internet protocol**.

### Comparación con el modelo OSI



Con IP se enruta el mensaje para que llegue a destino y TCP abre puertas y establece conexión de extremo a extremo (mirar modelo OSI).

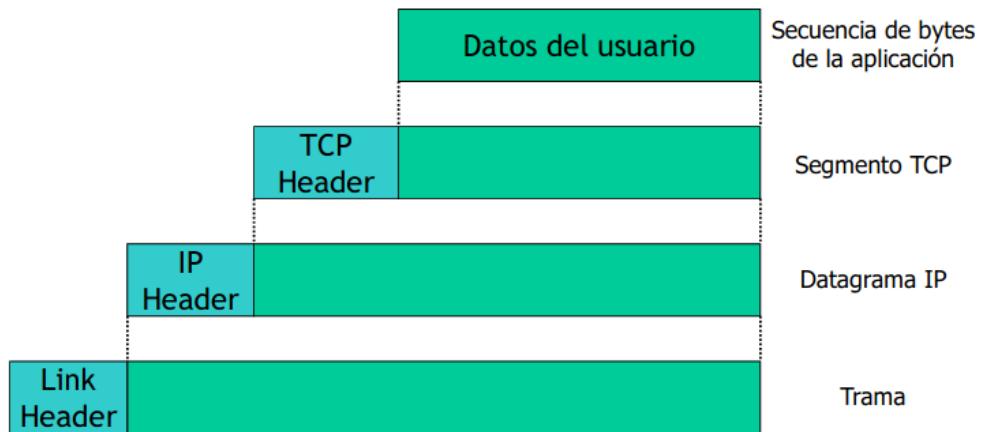
Observación: la barra vertical separa alternativas, es decir el protocolo TCP/IP permite el uso del protocolo TCP o UDP para la capa de transporte.

Todo esta montado sobre el internet protocol, que es un protocolo de red. **No define como debe operar la capa física y la capa de enlace.**

La forma en que se transportan los datos (marcado como interfaz de red) no es definida o especificada en el protocolo, por lo que se tiene la libertad de decidir como hacerlo mientras que permita el intercambio de mensaje (con forma de datagrama IP) entre dos nodos.

Las **capas del protocolo TCP/IP** son **interfaz, internet, transporte y aplicación**.

## Entorno TCP/IP



El encapsulado es igual que en el modelo OSI. Los datos del usuario son pasados a la capa de transporte que agrega una cabecera de transporte. Todo esto pasa a la capa de red que agrega una cabecera de red, y después pasa a la interfaz de red o link que agrega una cabecera de interfaz.

## Preguntas

### 1. Justifique si es necesaria o no una capa de red (capa 3 del modelo OSI) en una red de difusión (Broadcast).

Una red de difusión o Broadcast son por ejemplo la radio o la televisión. La antena difunde la señal y cada extremo sintoniza la señal. Como no necesito identificar a la fuente y al receptor (la comunicación es simplex, de una fuente a múltiples destinos) no necesito a alguien que traspase el mensaje.

Lo que podría utilizar es un repetidor pero eso es un dispositivo de capa 1.

Por lo tanto no lo necesito para radio o televisión. En el caso de streaming si la necesito porque estoy usando internet.

### 2. Analizando el entorno de OSI o TCP/IP, la unidad de datos del protocolo (PDU) de la capa N se encapsula en una PDU de la capa N-1. Igualmente, se puede partir la PDU del nivel N en varias PDU del nivel N-1 (**segmentación**), o agrupar varias PDU del nivel N en una única PDU del nivel N-1 (**agrupamiento**).

1. En la segmentación, ¿es necesario que cada segmento del nivel N-1 contenga una copia de la cabecera del nivel N?
2. En el agrupamiento, ¿es necesario que cada una de las PDU conserve su cabecera o se pueden agrupar los datos en una única PDU de nivel N-1 con una única cabecera del nivel N?



a) No es necesario, porque el bloque PDU de capa N con cabecera de capa N baja a N-1, no reconoce cual es la cabecera y cual es el mensaje, lo recibe como un bloque que es el mensaje a transmitir. Necesita agregar una cabecera de capa N-1 a cada segmento para indicar en que orden van.

b) Se tiene que conservar la cabecera de cada PDU del nivel N por lo explicado anteriormente. El aislamiento entre capas impide que dos capas distintas interpreten la cabecera del otro (la capa N-1 no entiende de datos ni cabecera de la capa N).

3. Tomando como ejemplo el modelo TCP/IP, supongo una primitiva que solicite el envío de un segmento. La llamada se realiza desde el nivel de transporte (TCP) hacia el nivel de red (IP). ¿Que parámetros debe pasar el TCP a IP como mínimo?

Primero el mensaje (la PDU del mensaje a transmitir), luego a quien se lo tiene que transmitir (la dirección a quien debe transmitir).

El protocolo IP toma estos dos datos y los encapsula agregándole una cabecera IP y se lo va a pasar a la interfaz de red.

5. Considerando el Modelo de capas OSI, ubique a los siguientes dispositivos en la capa que mejor describe las funciones que realiza:

- Repetidor: capa 1.
- HUB: capa 1.
- Bridge: entiende de bits y tramas → capa 2.
- Modem: solo entiende bits → capa 1.
- LAN Switch → capa 2.
- Router: también entiende de direcciones IP → capa 3.
- Firewall: puede ser de capa 4 hasta capa 7 (entiende de aplicaciones).

# REDES LAN

## Tecnología Ethernet

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

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

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

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

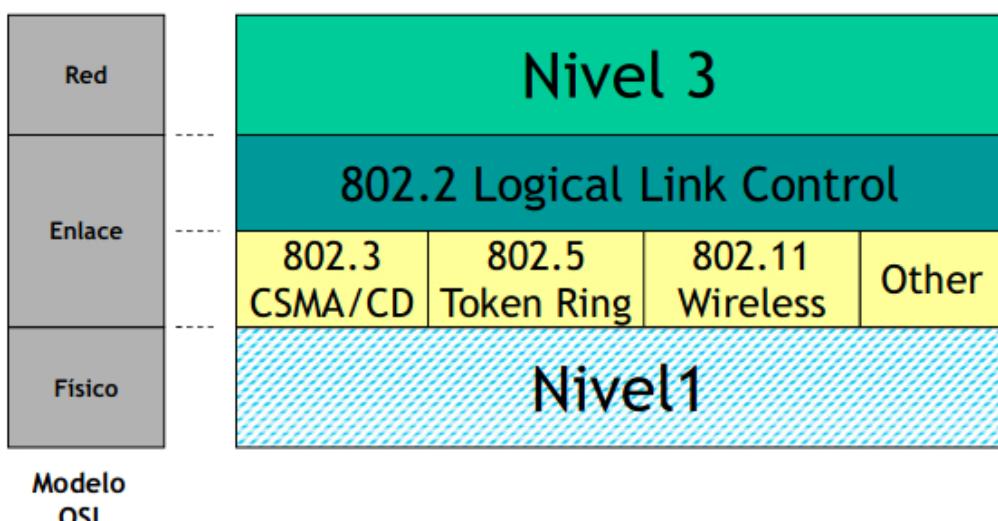
La IEEE 802.3 estandarizo el CSMA/CD que es lo que creó Ethernet (luego incorporó a Ethernet). Por lo que es lo mismo decir **protocolo 802.3 y Ethernet**.

## Evolución

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

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

## Donde se situá en el modelo OSI



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

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

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

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

### Subcapa LLC

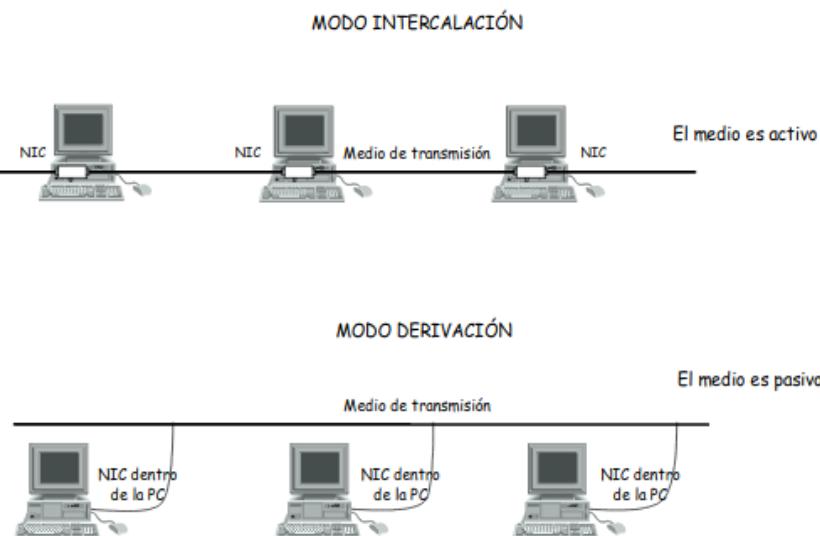
La subcapa LLC de Ethernet se ocupa de la **comunicación entre las capas superiores y las capas inferiores**. Se implementa mediante software (no depende del hardware).

### Subcapa MAC

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

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

### Topología original red LAN Ethernet



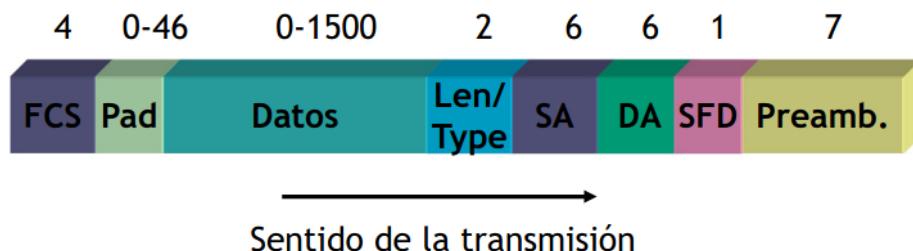
Se dice topología original porque el **BUS** hecho con cable coaxil que era inflexible, se suplantó por cable UPC. Se usa esta topología para explicar porque explica bien el funcionamiento y es fácil de interpretar.

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

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

## Formato de la trama Ethernet / 802.3

### Ejemplo de trama 802.3



La trama posee:

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

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

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

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

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

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

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

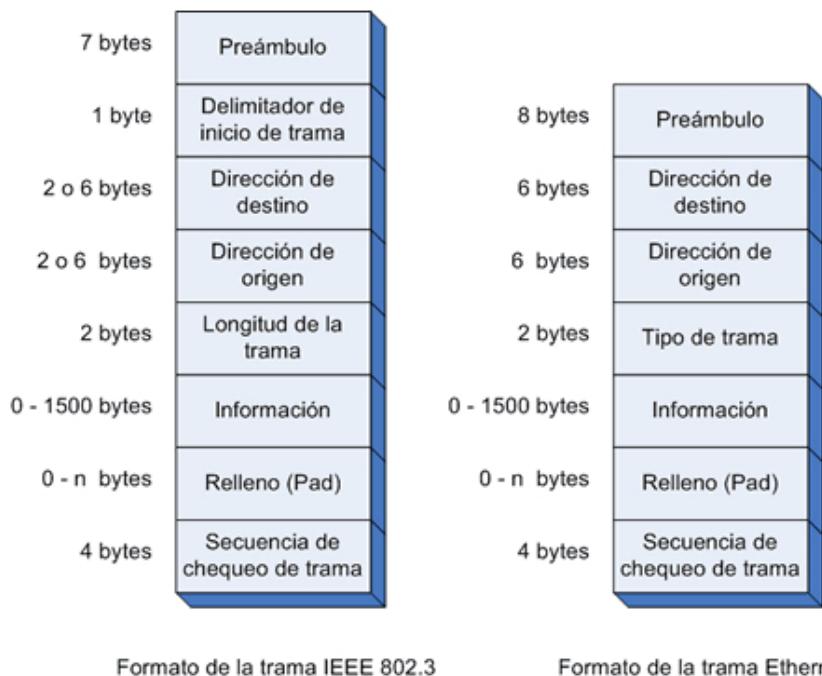
- Para **802.3** (Frame length) es el campo longitud que indica cual es la **longitud del campo de datos** de longitud variable que sigue a continuación.
- Para **Ethernet** (Ethertype) es el campo que indica cual fue la **capa usuaria (la capa superior)** del modelo OSI que **origino el mensaje**, así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

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

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

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

### Diferencia entre trama IEEE 802.3 y Ethernet



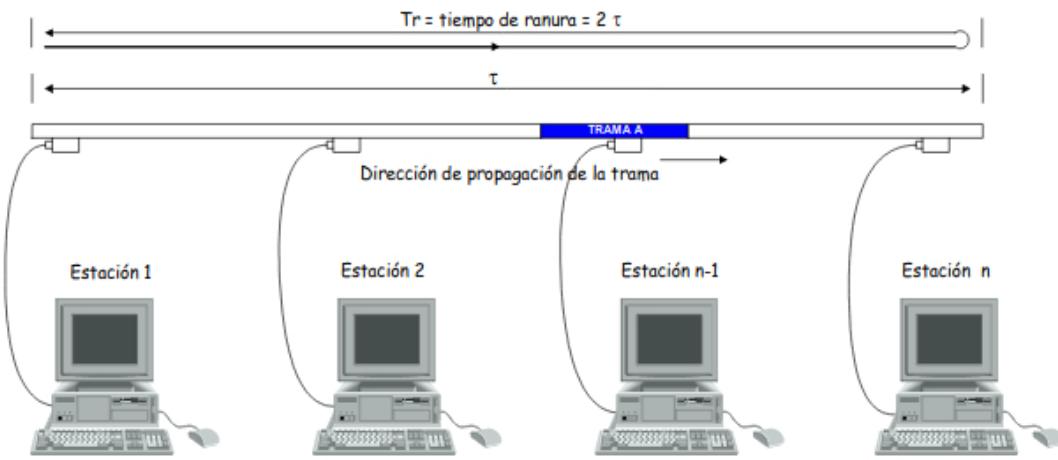
Una de las diferencias entre el formato de las dos tramas está en el **preámbulo**. El propósito del preámbulo es anunciar la trama y permitir a todos los receptores en la red sincronizarse a si mismos a la trama entrante. **El preámbulo en Ethernet tiene una longitud de 8 bytes pero en IEEE 802.3 la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

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

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

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

## Transmisión



## Mecanismo CSMA/CD

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

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

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

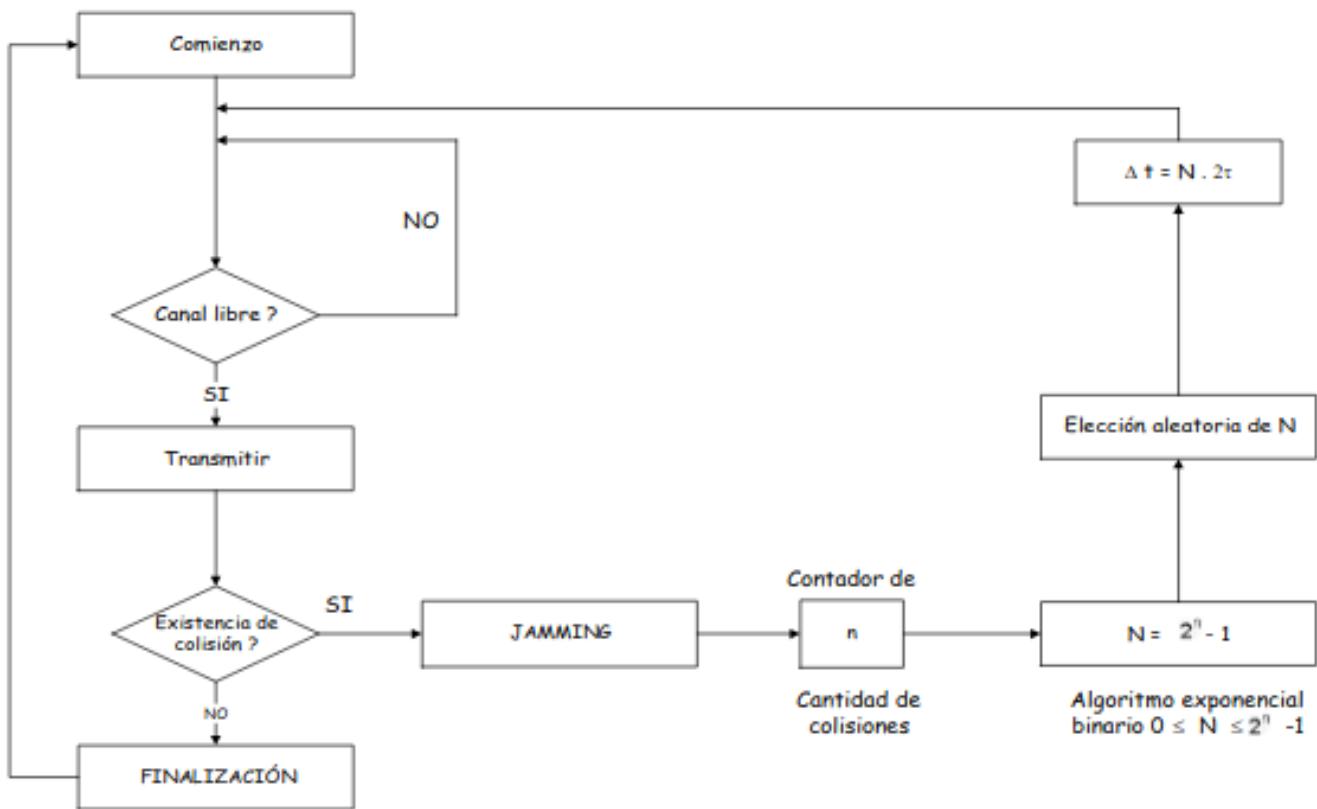
Al detectar una colisión, la estación pone en funcionamiento el algoritmo de **backoff exponencial**, que lo hace es ordenar las transmisiones en el tiempo.

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

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

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

## Algoritmo exponencial binario (backoff exponencial)



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

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

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

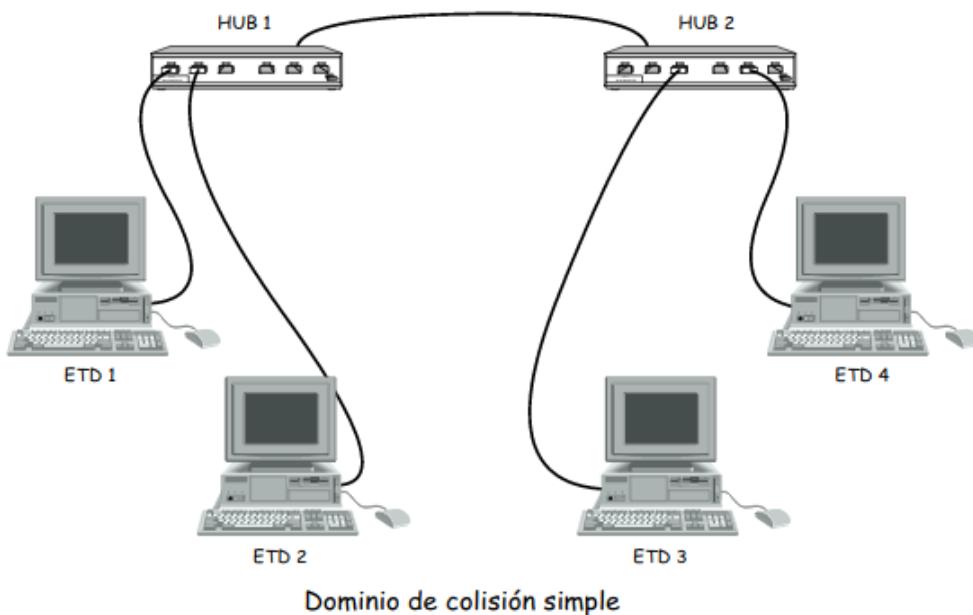
Luego de 10 colisiones consecutivas, se setea el máximo de ranuras a 1023 para bajar la probabilidad de colisión.

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

**Cuando logra transmitir, se resetea el contador de intentos.**

$N = (1/\text{probabilidad de colisión}) - 1$ .

## Dominio de colisión



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

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

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

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

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

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

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

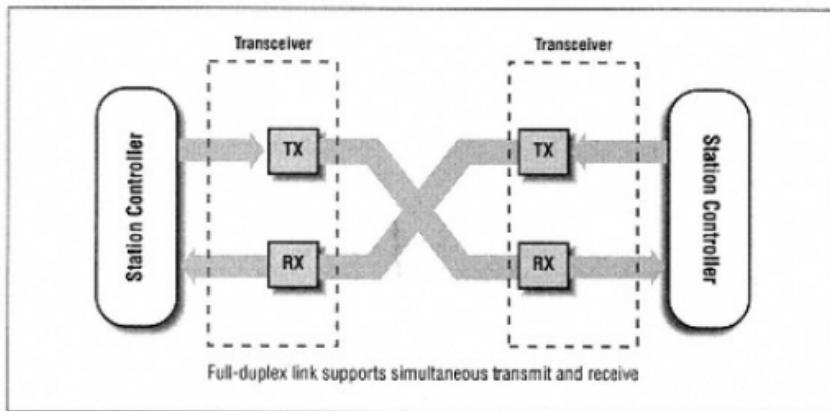
## Designación de tecnologías (configuraciones físicas)

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

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

## Interfaz full-duplex

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



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

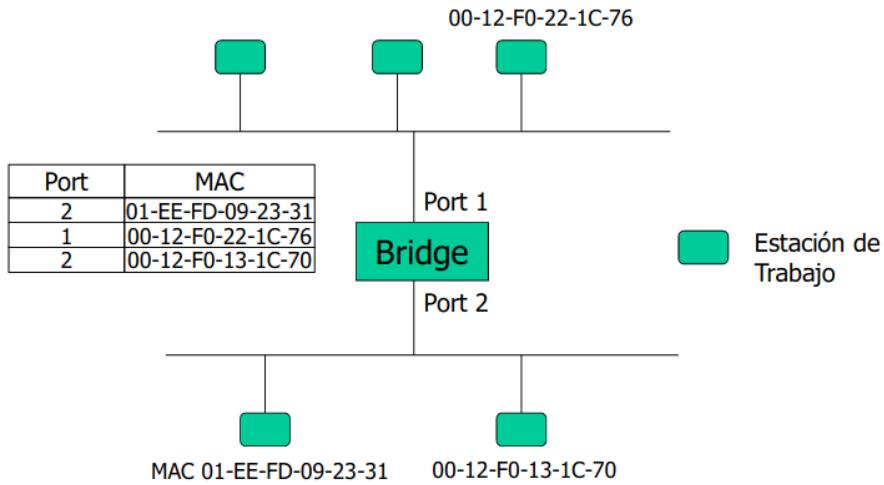
Como la **velocidad de transmisión** no se divide entre las estaciones del BUS, es **mucho mayor que en CSMA/CD** (además que las colisiones y la resolución de las mismas baja mucho el rendimiento).

## Bridging

Sirve para **mejorar el rendimiento**.

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

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



### Transparent Bridge

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

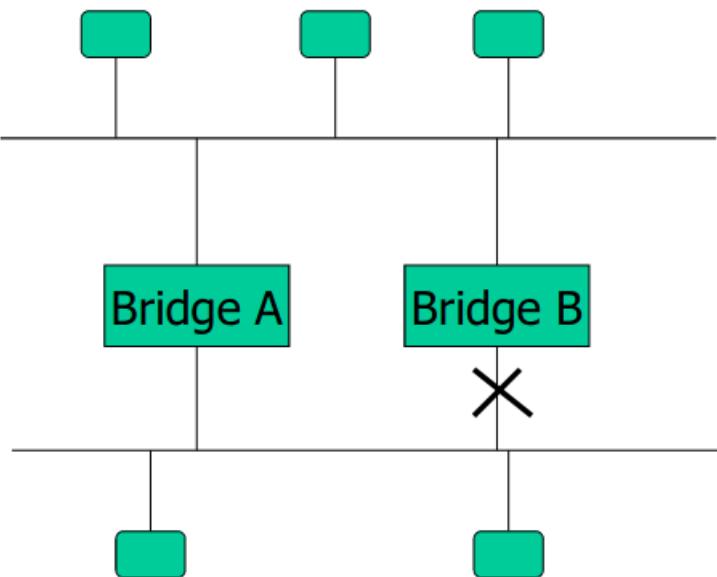
### Translating Bridge

- Realiza además conversión de protocolo y velocidad.

### Bridging Loop y STP

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

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



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

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

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

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

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

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

## Switch

Es un dispositivo de **capa 2**.

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

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

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

## Modos de operación

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

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

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

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

## Loop y STP

Sucede lo mismo que en los Bridges.

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

## VIRTUAL LANs (VLANs)

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

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

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

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

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

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

## Protocolo 802.1Q

### Problema que soluciona

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

## Protocolo 802.1Q

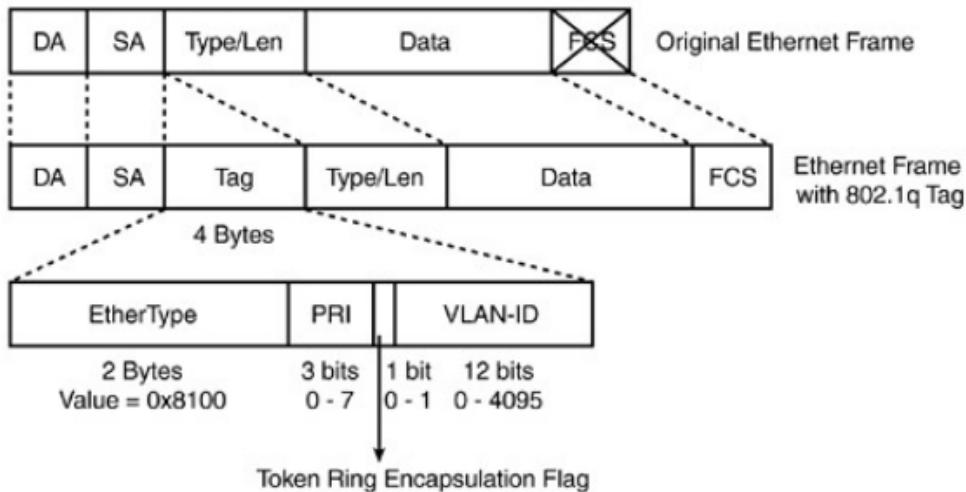
**Ambos switch deben soportar** el protocolo IEEE 802.1Q (protocolo de capa 2), es decir el VLAN trunking o VLAN tagging.

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

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

**Ninguna estación conoce el vínculo entre los switches.**

## Etiqueta en la trama



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

## Conexión a internet

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

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

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

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

## Servicio storage

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

## LACP

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

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

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

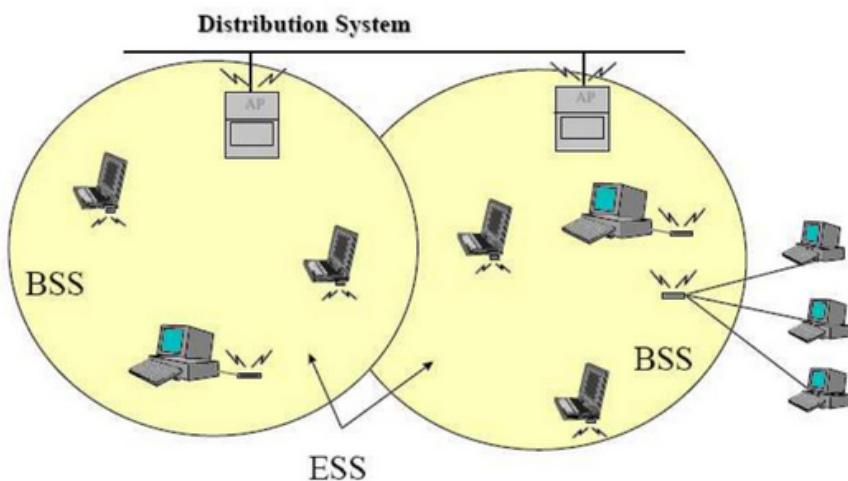
# WIRELESS LANs

Esta definida en la IEEE 802.11.

## Redes Ad-Hoc

Son aquellas que puedo formar entre por ejemplo dos laptop conectadas de forma wireless. En este caso uno de los dispositivos asume el rol de AP. La forma de operación es idéntica al del modo infraestructura.

## Modo infraestructura



## Composición

Posee una estructura celular, donde cada **celda (BSS)** contiene:

- **Distribution System (DS):** generalmente la red LAN cableada.
- **Access Point (AP):** estación base a la cual se conectan los terminales remotos.
- **Terminales.**

## Funcionamiento básico

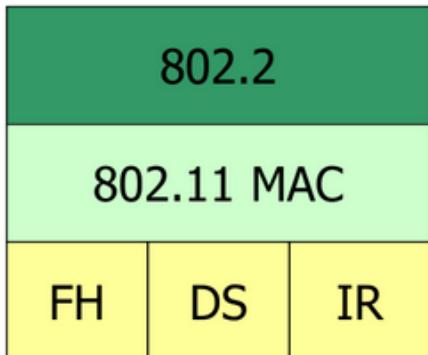
Las estaciones se conectan a un dispositivo conocido como **access point** (el de arriba). El **access point** posee una antena que cubre un área determinada llamada **BSS** (basic service set).

El access point da servicio de integración hacia un sistema de distribución. Es decir vincula los dispositivos a un switch conectado a recursos cableados.

Cuando hay dos o mas **access point (AP)**, sus BSS conforman un **extended service (ESS)**, cada dispositivo conectado al AP tiene acceso al mismo sistema de distribución.

Los dispositivos en una red wireless no se comunican entre ellos, sino que se comunican a través del AP.

## Modelo



802.11



Modelo OSI

Tres **variantes en la capa física**: infrarrojo (IR), salto de frecuencia (FH) y frecuencia directa (DS). Son **técnicas de modulación**.

La capa de **enlace** se encuentra **dividida en dos subcapas**.

## Donde opera wireless

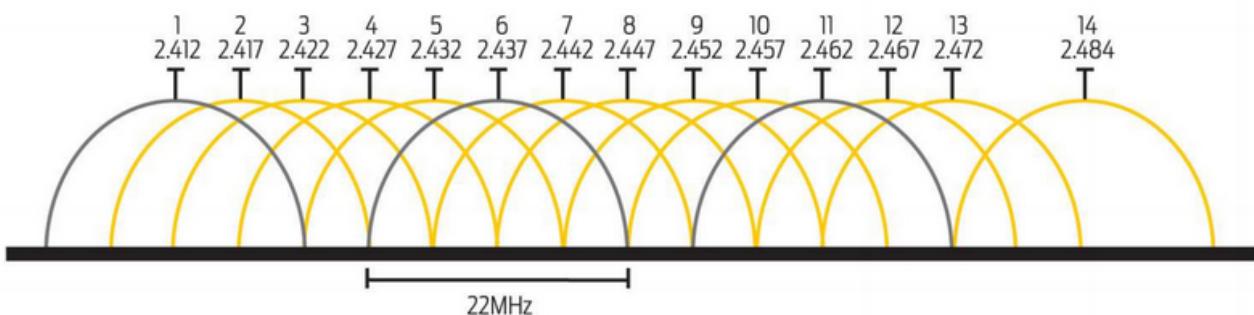
### Banda 2.4Ghz

La banda esta dividida en canal de **22Mhz** (depende de cada país).

Estos **canales se solapan**, pero **hay tres que no lo hacen: 1, 6 y 11**. Por lo que **si quiero en un lugar implementar 3 AP cubriendo una misma área, voy a tener que definir cada uno en canales diferentes, sino se van a estar interfiriendo y ocupando el mismo ancho de banda**.

### The 2.4GHz channels

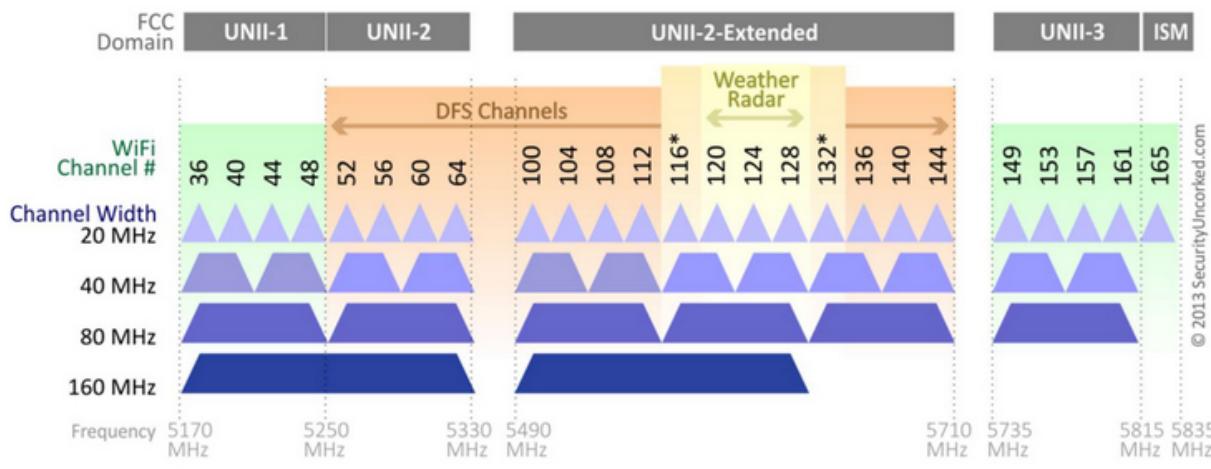
Channel Centre Frequency (GHz)



### Banda 5Ghz

Hay **muchos mas canales** (de **20Mhz c/u**) que no se solapan. La banda de **5Ghz** ademas permite hacer **backlink**, que consiste ir **agregando canales (uniendo)** de manera de obtener canales de **40Mhz hasta 160Mhz**.

## 802.11ac Channel Allocation (N America)



### Control del acceso al medio en wireless

En wireless **no podemos utilizar el mismo método de acceso al medio que en wired (CSMA/CD)** porque si dos transfieren probablemente **no se escucharían entre si** (o algo así). **Utiliza CSMA/CA.**

### CSMA/CA

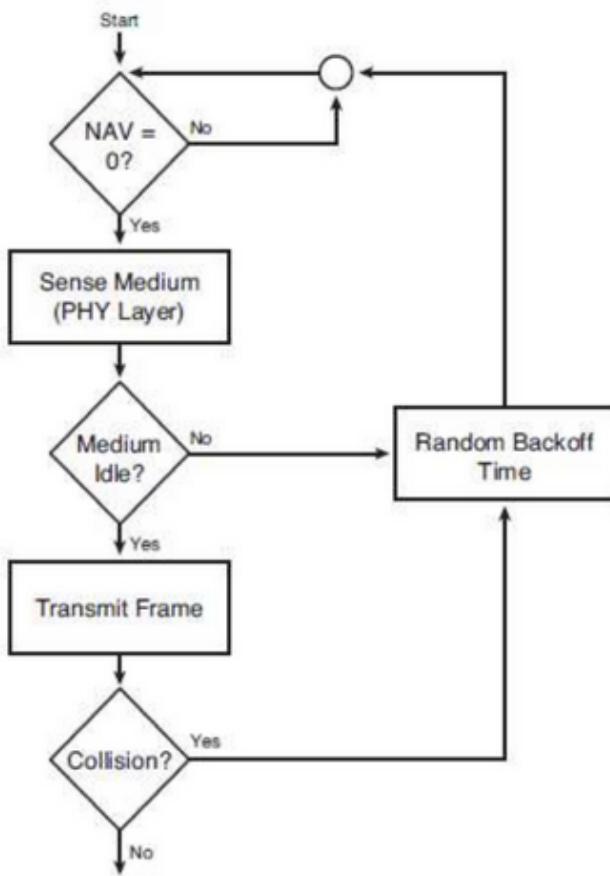
**Acceso múltiple por detección de portadora con evitado de colisiones.** Consiste en **escuchar antes de hablar, durante un periodo de tiempo**. Las implementaciones de este protocolo tratan de evitar las colisiones en todo lo posible:

### DCF (función de control distribuida) sin RTS/CTS

- La estación que necesita transmitir se fija si el **NAV** (vector de asignación de red) esta en cero osea, si hay alguien transmitiendo. Si no esta en cero, espera.
- Cuando el NAV llego a cero, escucha el medio por un intervalo **DIFS** (distributed inter frame space) que no haya nadie transmitiendo.
- Si esta libre, **transmite la trama.**
- Si el emisor recibe el **ACK** significa que la transmisión fue hecha correctamente. Podría no recibirla porque hubo colisión, la trama llegó dañada (NACK) o el ACK llegó dañado, en ese caso debo volver a retransmitir luego de un tiempo aleatorio.

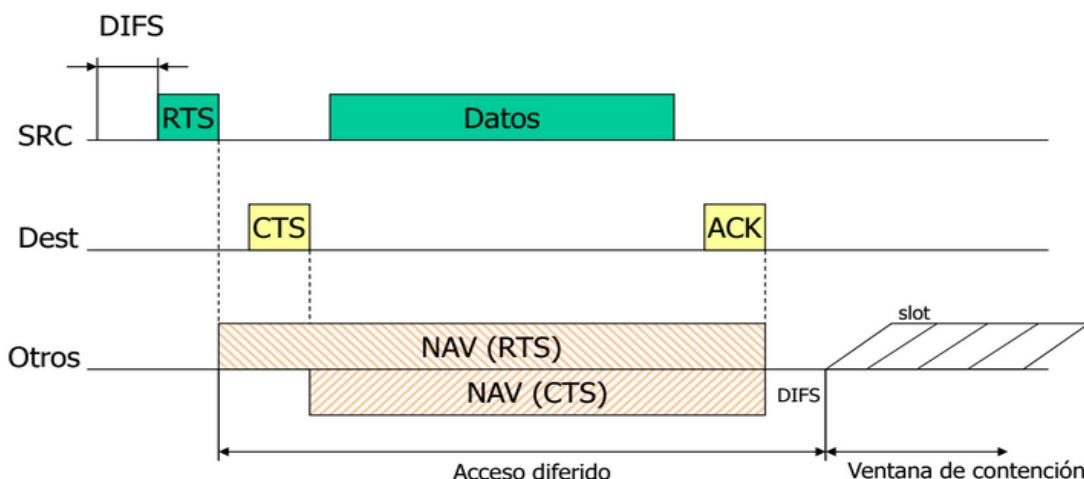
El **algoritmo exponencial binario (backoff**, tiempo de espera random para volver a transmitir) debe ejecutarse en cada uno de los siguientes casos:

- Cuando escucha el medio y este esta ocupado.
- Después de una retransmisión.
- Después de una transmisión exitosa.



## DCF con RTS/CTS

- Se fija si el **NAV** esta en cero. Caso contrario espera.
- Una estación transmite solo si el medio esta libre por un intervalo de tiempo determinado **DIFS** (conocido como DIFS, distributed inter frame space).
- Si el medio esta libre, la estación va a transmitir una trama conocida como **RTS** (ready to send) y “reserva el medio”.
- Recibe un **CTS** (clear to send) del AP.
- **Transmite los datos.**
- El receptor verifica el CRC y envíá un **ACK**.



- Dest es el access point y SRC quien transmite.
- Hay dispositivos que van a escuchar el RTS y otros que no.
  - El que lo escucha, van a settear una variable de estado **NAV** (network allocation vector) con un valor de tiempo que lo saca de la trama RTS. El **emisor incluyó en esa trama el intervalo de tiempo que necesita emitir** (recibir el CTS, transmitir los datos, recibir el ACK y x tiempo mas tarde libero el medio). La estación checkea el NAV antes de empezar a transmitir.
  - Los que no escuchan el RTS, escuchan el CTS que envía el AP. En el **CTS** el AP le indica cuanto **tiempo le queda al emisor** del tiempo que solicito en el RTS. Los dispositivos que escucharon el CTS y no el RTS, settean el NAV en el tiempo que dijo el CTS que le quedaba al emisor.

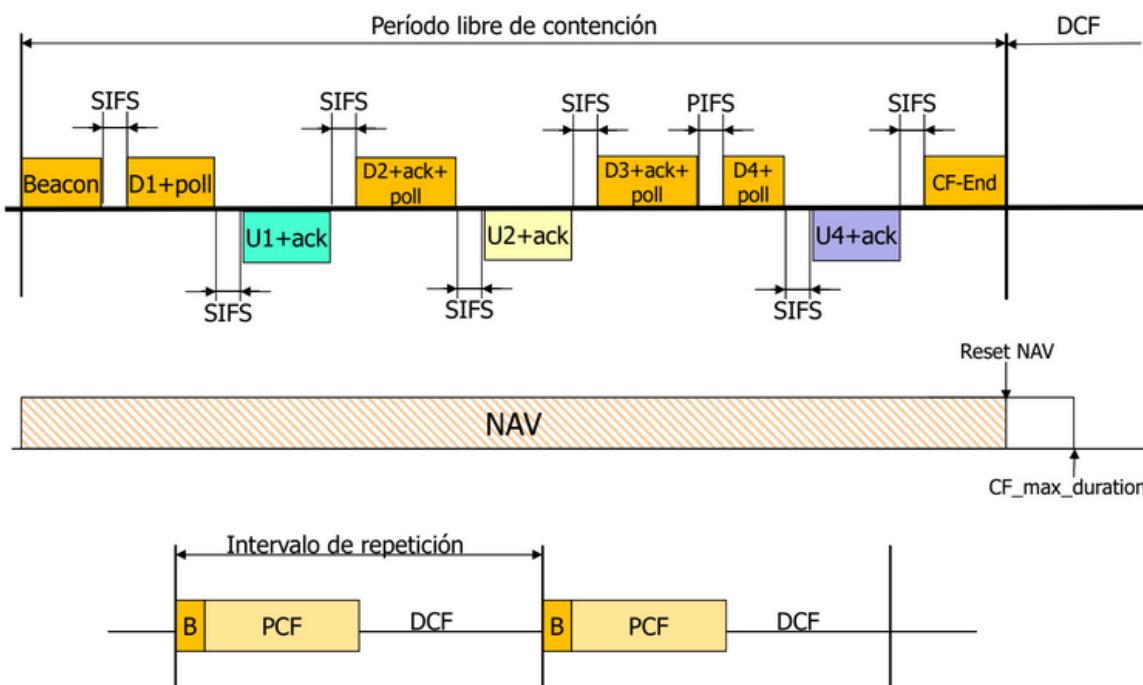
Este método con confirmaciones se lo llamo **DCF** (función de control distribuida) **con RTS/CTS**, y se realizaban por el **problema del nodo oculto** (una estación no escucha el RTS, por una interferencia por ejemplo una pared). Luego se dieron cuenta que ese problema no ocurría frecuentemente y que la solución **bajaba mucho la performance**, por lo que pasaron el mecanismo de **RTS/CTS opcional**.

### PCF (función de control puntual)

En las funciones anteriores, no tengo certeza cuando voy a poder transmitir (tengo que esperar que el medio esta libre).

En PCF el AP va a ir alternando periodos de DCF (distribuido) con periodos de PCF (puntual), entonces durante un intervalo de tiempo el acceso el medio funciona como vimos antes (el que puede transmitir transmite, sino espera, etc) pero **en un momento dado el AP determina quién habla (durante el periodo PCF)**. Una vez finalizado este periodo, vuelve al periodo DCF y así sucesivamente.

**Forma de decisión quien transmite:** los terminales se asocian al AP, por lo que **el AP toma registros de ellos y hace una lista de los que podrían transmitir**. El AP envía bits (en el dibujo son los poll) que **indica quien es el siguiente en la lista que puede transmitir**. Si el AP pasado un tiempo PIFS no recibe datos del elegido, le dice al siguiente de la lista que puede transmitir.

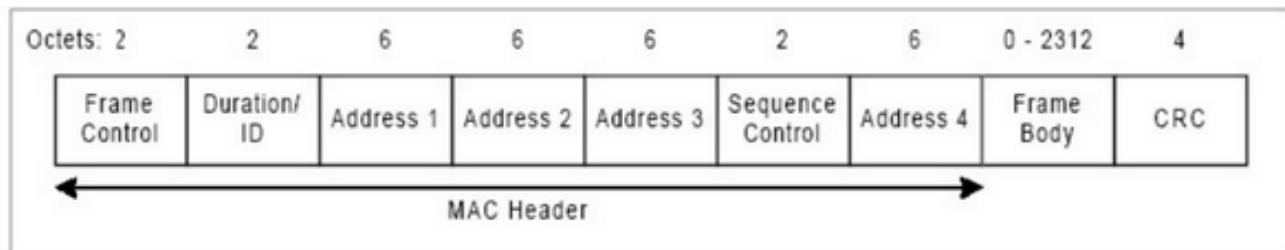


Entre una trama y otra en el intervalo PCF, la distancia de tiempo que hay es un SIFS (short interframe space). SIFS < DIFS porque de esta manera me aseguro que ningún dispositivo va a encontrar el medio libre durante un intervalo de tiempo DIFS y se va a poner a transmitir durante un intervalo de PCF.

**Como se indica el inicio y finalización del PCF:** El AP transmite unas tramas administrativas llamadas **Beacon**, una Beacon indica el inicio del intervalo PCF y otra indica el fin del intervalo. De esta manera todos los dispositivos en el rango BSS leyendo lo que dicen estos Beacon van a enterarse que comienza y finaliza el intervalo PCF.

Para optimizar el tiempo, el AP envía una trama con toda la información necesaria (por ejemplo, si recibió una trama para retransmitir, envíá estos datos, el ACK y el poll).

## Trama wireless

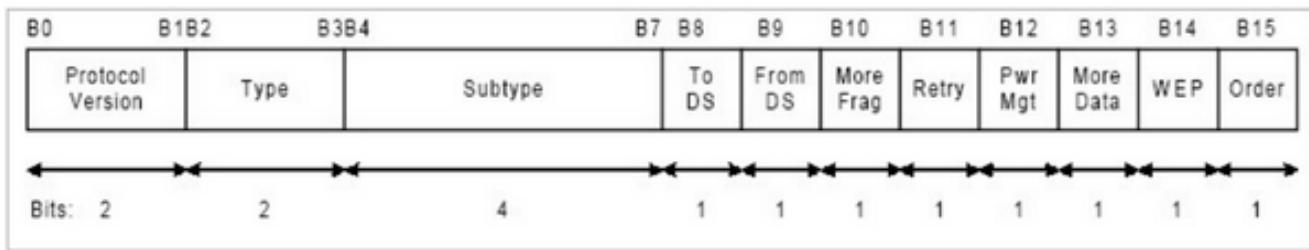


Los campos son:

- **Duration/ID:** intervalo de tiempo utilizado para calcular el **NAV time**.
- **Address fields:** ver cuadro. Los valores de las address MAC dependen de la combinación de to DS y from DS. Mayor explicación mas abajo.
  - DA: Dirección de destino final real (donde culmina esa transmisión).
  - SA: Dirección de origen real (la que empieza esa transmisión).
  - BSSID: ID del BSS.
  - RA: Receiver address (address del dispositivo que va a recibir esa trama, por ejemplo TA es un terminal y RA el AP o al revés, TA el AP y RA el terminal o de AP a AP).
  - TA: Transmitter address.
- **Sequence Ctrl:** identifica el fragmento.
- **CRC:** 32-bit.

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

## Frame control



Las tramas pueden ser de tipo/subtipo:

- **Administración:** petición/confirmación de asociación, autenticación, Beacon.
- **Control:** RTS, CTS, ACK.
- **Datos.**

Otros campos:

- **ToDS:** vale 1 cuando la trama se envía al AP (DS es distributed system, refiere la linea que esta conectado el AP). From 0 to 0 significa que el host envía autenticación al AP (no pasa al cableado, por ej la contraseña). 1 1 es para por ejemplo, wireless bridge.
- **FromDS:** vale 1 cuando la trama viene de un AP (1 a de un terminal).
- **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. Si esta en 1 significa que se fue a dormir la estación y que por un tiempo no le envíe nada el AP. Si alguien se quiere comunicar con el dormido, el AP guarda la trama en un buffer para enviarla despierte. El AP puede pedir que todas estén despiertas en cierto tiempo.
- **More Data:** idem, el AP indica a la estación que tiene mas fragmentos para ella. Sirve para indicarle a la estación que no se duerma porque el AP tiene mas datos que enviarle.
- **WEP:** Privacidad equivalente al cableado. Indica que el campo de datos está encriptado.

## Diferencias con el Ethernet cableado

### Fragmentación y reensamblado

La capa de red, en función a la tasa de error (BER, de capa 1), va a ir controlando el tamaño de la trama a transmitir (**cuan mas grande es el BER, mas fragmentada debe estar la trama para retransmitir el menor tamaño posible**). Se transmite un fragmento a continuación del otro.

## **Proceso de asociación a una red wireless**

Para incorporarse a una celda, una estación debe completar los siguientes **pasos**:

- **Sincronización:** por medio de los “**Beacon frames**” transmitidos por el AP. El AP envía 10 Beacon frames por segundo.
- **Autenticación:** intercambio de información (**clave**) entre el AP y el terminal.
- **Asociación:** se vinculan la terminal al AP. A partir de este momento puede comenzar a transmitir.

## **Protección**

Se utiliza el protocolo WEP (wired equivalent privacy) para evitar la intercepción de la información. Y para limitar el acceso de los terminales a un determinado AP.

## **Comparación de normas 802.11 wireless LAN standards**

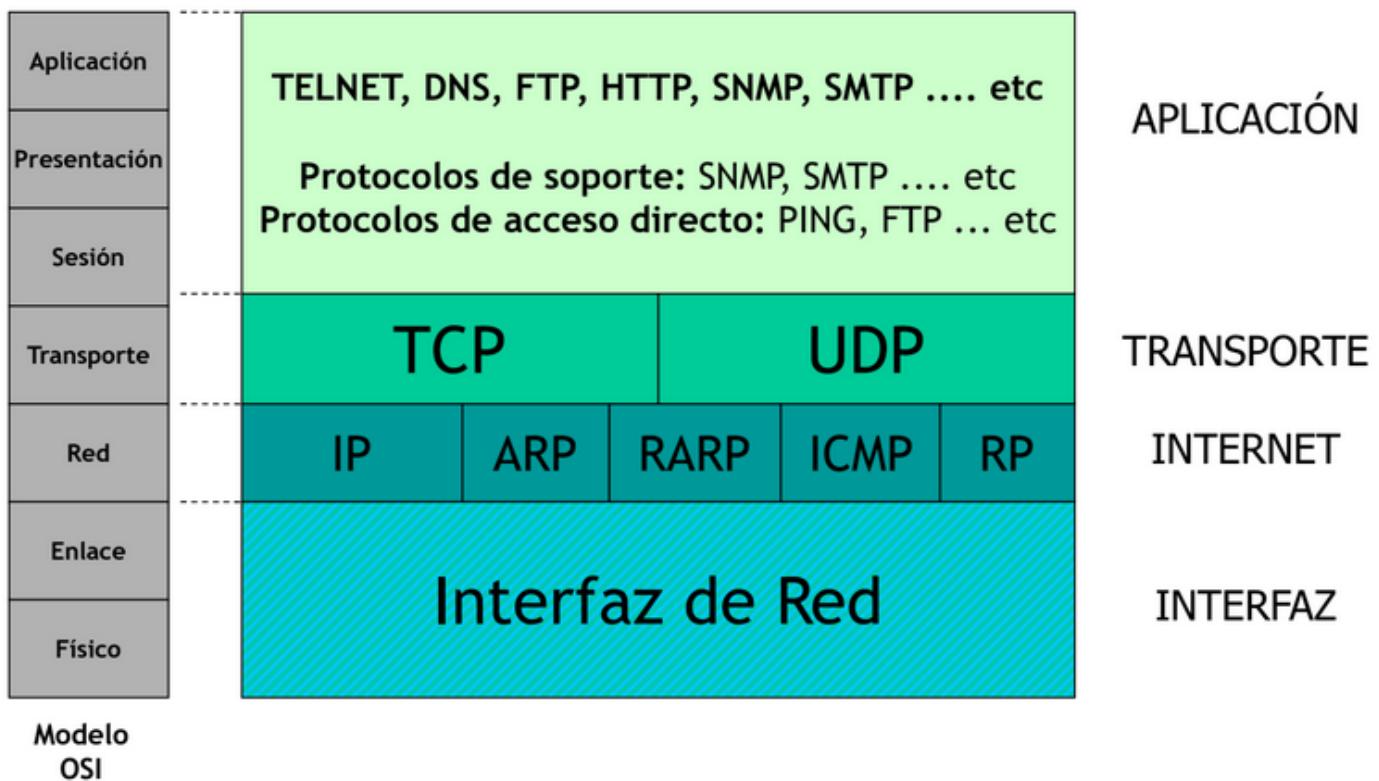
	<b>802.11a</b>	<b>B</b>	<b>G</b>	<b>N</b>	<b>AC</b>
Velocidad (Mbps)	54 Mbps	11	54	Hasta 600	433 / 867 / 1.69 Max 3.39 Gbps
Frecuencia de operación (GHz)	5	2.4	2.4	5 & 2.4	5 & 2.4
Modulación	OFDM	DSSS	OFDM, DSSS	OFDM	OFDM

## SUITE TCP/IP

Suite transmission control protocol / internet protocol.

Decimos que es una suite de protocolos porque hay una serie de ellos, incluido TCP/IP.

### Arquitectura TCP/IP



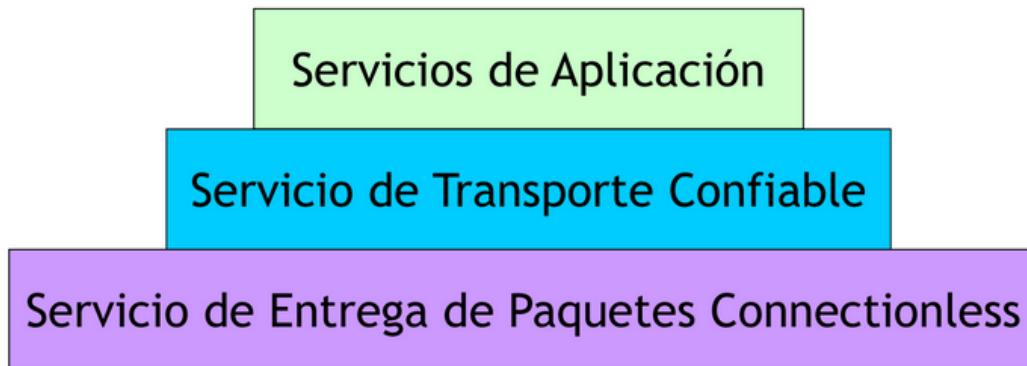
**TCP/IP no especifica la interfaz de red**, es decir no especifica sobre que protocolo de enlace debe ser encapsulado sino que esta abierto a ser encapsulado en cualquier protocolo de capa dos.

En la **capa de red o protocolo de internet** el único es IP, el resto son de **soporte** para IP (ARP, RARP, etc).

En la de **transporte tenemos dos versiones**, la **confiable que es la TCP** y la **no confiable o ligera que es la UDP**.

Por encima del protocolo de transporte tenemos aplicaciones corriendo sobre estos protocolos.

## Filosofía



La **filosofía** con la cual se creo esta arquitectura fue la de crear servicios sobre servicios de aplicación, sobre un servicio de transporte confiable (o adrede como UDP), sobre un servicio de paquetes no orientados a la conexión.

**Protocolo orientado a la conexión:** es un protocolo en el cual **para establecer una conexión**, pasa por tres estados: **etapa de establecimiento de conexión, otra de intercambio de información y otra de cierre o desconexión.**

**Protocolo no orientado a la conexión:** no pasa por la etapa de establecimiento ni de cierre, **sólo por el intercambio de información**. Ethernet es no orientado a la conexión (le basta con solo tener la dirección MAC destino, **no pregunta si quiere intercambiar al destino**).

### IP = Servicio connectionless

Que sea no orientado a la conexión **implica**:

- **No es confiable:** significa que **los paquetes pueden ser:**
  - **Perdidos:** puede pasar que la capa de transporte le pase una PDU a IP para que la transmita, e IP la pierda.
  - **Duplicados:** entregue dos copias del mensaje.
  - **Desordenados.**
  - **Demorados:** no esta garantizado cuanto puede demorar en llegar.
- **Connectionless: paquetes tratados independientemente.**
  - **No existe un “estado” en los routers acerca** de cómo fueron tratados los **paquetes anteriores**, ni que contenían. Cada paquete va a buscar como llegar al destino, no tiene un camino de routers predefinido.
- **Entrega best-effort:** el software realiza un **serio intento por entregar el paquete** (sin garantía).

## Datagrama IP

Datagrama es la cabecera del protocolo mas los datos.

0	4	8	16	19	24	31							
Vers.	HLEN	ToS	Longitud Total										
Identificación		FLAGS	Desplazamiento del Fragmento										
TTL	Protocolo	Checksum del encabezado											
Dirección IP Fuente													
Dirección IP Destino													
Opciones (si las hay)				Relleno									
DATOS													
...													

### Versión (4bits)

Puede variar entre (0100) o (0110) dependiendo si se utiliza IP versión 4 (IPv4) o IP versión 6 (IPv6). Este campo describe el formato de la cabecera utilizada. En la tabla se describe la versión 4.

### Tamaño de cabecera (HLEN. 4Bits)

Longitud de la cabecera, en palabras de 32 bits. Se utiliza debido a que los últimos campos de la cabecera son de longitud variable, por lo que el origen debe poner la longitud total de la cabecera. Su valor mínimo y común es de 5 palabras ( ) para una cabecera correcta, y el máximo de 15 palabras ( $15 \times 32 = 480$  bits, 60 bytes).

### Tipo de servicio (ToS, 8bits)

Indica una serie de parámetros sobre la calidad de servicio deseada durante el tránsito por una red. Algunas redes ofrecen prioridades de servicios, considerando determinado tipo de paquetes "más importantes" que otros (en particular estas redes solo admiten los paquetes con prioridad alta en momentos de sobrecarga).

Solo a 4bits se les dio significado, los otro 4bits son reservados para el futuro.

De los 4bits solo uno podía estar encendido, los bits significan:

- Bit 0: Minimize Delay (minimizar retardo, envialo por el camino que tarde menos).
- Bit 1: Maximize throughput (enviarlo por el camino que permita la mejor velocidad real de transferencia de datos).
- Bit 2: Maximize reliability (maximizar la confiabilidad, enviarlo por el camino mas confiable).
- Bit 3: Minimize cost (enviarlo por el camino mas barato).

Para saber el camino más rápido, etc pasa lo siguiente: se envía el dato con el bit de minimizar delay, pasa por el proveedor que tiene un conjunto de routers y lo hace pasar por el camino mas rápido de sus routers, este proveedor se lo pasa a otro y así. Hoy en día no suelen mirar el campo de ToS porque no discriminan entre clientes (ademas de las conveniencias de la empresa).

Valores recomendados para el ToS según 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

### Longitud total (16bits)

Es el **tamaño total, en octetos, del datagrama, incluyendo el tamaño de la cabecera y el de los datos**. Al ser 16bits el **tamaño máximo del datagrama es 65.536 bytes**.

### Identificación (16bits)

**Identificador único del datagrama.** Se utilizará, **en caso de que el datagrama deba ser fragmentado, para poder distinguir los fragmentos de un datagrama de los de otro**. El originador del datagrama debe asegurar un **valor único para la pareja origen-destino y el tipo de protocolo durante el tiempo que el datagrama pueda estar activo en la red**. El valor asignado en este campo debe ir en formato de red.

### Flags (3bits)

Actualmente utilizado sólo para especificar **valores relativos a la fragmentación** de paquetes. Los 3 bits (por orden de mayor a menor peso) son:

- **bit 0: Reservado; debe ser 0.**
- **bit 1: 0 = Divisible, 1 = No Divisible (DF).**
- **bit 2: 0 = Último Fragmento, 1 = Fragmento Intermedio (le siguen más fragmentos) (MF).**

### Desplazamiento del fragmento (13bits)

En **paquetes fragmentados** indica la **posición**, en unidades de 64 bits, **que ocupa el paquete actual** (offset de la data CREO) **dentro del datagrama original**. El primer paquete de una serie de fragmentos contendrá en este campo el valor 0.

### Tiempo de vida (TTL, 8bits)

Indica el máximo **número de enrutadores que un paquete puede atravesar**. Cada vez que algún nodo procesa este paquete disminuye su valor en, como mínimo, una unidad. **Cuando llegue a ser 0, el paquete será descartado**. Se utiliza porque es un protocolo no orientado a la conexión.

### Protocolo (8bits)

Indica el **protocolo de las capas superiores al que debe entregarse el paquete** (TCP = 6, ICMP=1 o UDP = 17).

## Suma de Control de Cabecera (16bits)

Suma de control de la **cabecera (no datos)**. Se **recalcula** cada vez que algún nodo cambia alguno de sus campos (por ejemplo, el Tiempo de Vida). **Si encuentra errores, descarta el datagrama**, no se generan mensajes de error.

## Dirección IP de origen (32bits)

Debe ser dada en formato de red.

## Dirección IP de destino (32bits)

Debe ser dada en formato de red.

## Opciones (bits variables)

Aunque **no es obligatoria** la utilización de este campo, cualquier nodo debe ser capaz de interpretarlo. Puede contener un **número indeterminado de opciones**. Las opciones están definidas y estandarizadas pero la gran mayoría están en **desuso**.

## Relleno (bits variables)

Utilizado para **asegurar** que el **tamaño, en bits, de la cabecera es un múltiplo de 32 y para llegar al tamaño mínimo del datagrama**. El valor usado es el 0.

## Datos (bits variables)

El **máximo**, al igual que los demás variables, sale de **restar la longitud total de la ocupada**.

## Fragmentación y reensamblado

Cuando un router recibe un paquete, lo des-encapsula para leer la dirección destino y si no era para él, y **lo vuelve a encapsular pero en el protocolo/tecnología que tenga el enlace entre el y el router al que se lo debe pasar**. Cada tecnología tiene su propia **limitación respecto al tamaño del mensaje**, que es lo que se conoce como **Maximum Transfer Unit (MTU)**.

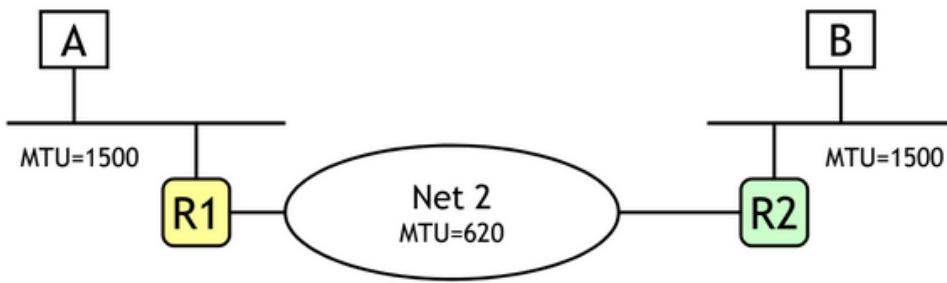
**MTU:** Cada tecnología de conmutación de paquetes, fija un **límite máximo para la cantidad de datos que pueden transmitirse en una única trama**.

Red	MTU (Bytes)
Token Ring 16 Mbps	17914
IEEE 802.3	1500
X.25	576

Cuanto **mas bytes puede enviar por mensaje, mas eficiente es la transmisión** (bits datos/ bits totales).

Si el router 1 que trabaja con IEEE 802.3, debe pasarle una trama de 1500bytes al router 2 que trabaja con X.25 (MTU menor), debe **fragmentar esa trama en tramas más pequeñas**.

## Funcionamiento



- “A” que esta conectada a una interfaz con una MTU=1500, arma un datagrama de 1500bytes, lo encapsula en una trama Ethernet y se lo pasa a R1 (router).
- El router lee la dirección MAC que sea para el, va a leer el EtherType (va a ver que es un datagrama IP), va a ir al campo de datos y va a retirar de ahí el datagrama IP y se lo va a entregar al módulo IP. El módulo IP va a calcular el checksum y va a ver la dirección destino y toma la decisión de por cuál interfaz lo envía.
- Lo que sigue es encapsular el datagrama en el protocolo Net 2. Tiene un datagrama IP de 1500bytes y Net 2 tiene un MTU de 620bytes. Como **IP ofrece un mecanismo de fragmentación y reensamblado**, por lo que R1 va a fragmentar el mensaje.
- IP oculta los detalles de tecnología subyacente. **Divide los datagramas en fragmentos que deben ser re-ensamblados en la dirección destino.**

Datagram Header	Data1 600 bytes	Data2 600 bytes	Data3 280 bytes
-----------------	--------------------	--------------------	--------------------

Fragment1 Header	Data1	Fragment 1 (Offset 0)
Fragment2 Header	Data2	Fragment 2 (Offset 600-75)
Fragment3 Header	Data3	Fragment 3 (Offset 1200-150)

La fragmentación la realiza la capa IP (puede hacerlo un router o un terminal).

El reensamblado lo realiza el destino final en la capa IP.

El **fragment header** contiene:

- **Dirección origen y dirección destino.** Es el mismo que el del datagrama original.
- **Campo identificación:** todos los fragmentos tienen el mismo para ser identificados como una misma unidad. Es el mismo que el del datagrama original.
- **Campo desplazamiento del fragmento (offset):** con el offset **reconozco el orden de los fragmentos**. Como el tamaño del datagrama original es demasiado grande, no puedo tomar el offset real, por lo que se elige un múltiplo de 8bytes más próximo al MTU del trayecto (en el ejemplo, 600 se representa con  $600/8 = 75$ ).

- **Flags.**
  - Bit 1: bit sin uso.
  - Bit 2: **no fragmentar.** Si se encuentra en 1 impide la fragmentación, por lo que si no es posible pasarlo por el MTU se descarta y se genera un mensaje ICMP (reporte de error al origen que necesita fragmentar y no lo dejan).
  - Bit 3: **mas fragmentos.** Si esta en 0 es el ultimo. Si se fragmenta un datagrama con el bit de mas fragmentos en 1, ninguno de sus fragmentos va a tener el bit en 0 debido a que ese datagrama también era un fragmento.

Cada fragmento conforma un datagrama independiente para la red, es decir que si un router fragmenta, el router destino no se entera. Los tres conforman un datagrama fragmentado solamente para el destino, que es el que lee el bit “mas fragmentos”.

### Desventajas de la fragmentación

- Duplica la **probabilidad de pérdida** de un datagrama.
- Genera **mayor carga de procesamiento** en los routers.

## Direccionamiento

Un protocolo de capa tres debe ser capaz de proveer direcciones que permitan identificar a los miembros de la red y mecanismos que permitan el encaminamiento (es decir como llegar de un nodo a otro).

En el protocolo IP **a cada host se le brinda una dirección IP con las siguientes características:**

- Es **única en internet**, no hay dos direcciones IP iguales.
- Tiene **32bits de longitud**.
- Se suelen representar como 4bytes separados por un “.” con notación decimal (ejemplo 24.323.218.197).
- **Parte de la dirección IP identifica a la red, la otra parte identifica al host dentro de la red.** Que parte corresponde a la red y que parte corresponde al host lo determina la **Mascara de Subred**.
  - **Identificador o prefijo de red:** parte de la dirección IP que es igual para todos los miembros de la red. Se lo llama prefijo de red. **Los proveedores de red deben pedir este prefijo al organismo** (único mundial) **que los gestiona**.
  - **Identificador de host:** parte de la dirección IP que **permite identificar unívocamente al host dentro de la red**. El identificador de host **se debe pedir a la red**.
    - No puede ser 0 porque es reservado para identificar la red.
    - No puede ser 255 porque es el numero reservado para broadcast en la red.
- La **netmask** o mascara de subred **identifica red / host**. Un bit igual a 1 significa que ese bit de la dirección IP corresponde a la red y un 0 al host. (255 son 8bits en 1. Ejemplo 255.255.255.0, recién el ultimo octeto o separación corresponde al host).

Si el host quiere enviar un mensaje a una dirección destino que se encuentra en su misma red, va a averiguar la MAC address del destino, va a armar una trama de Ethernet con las direcciones MAC address e IP y va a enviar directamente el mensaje.

En caso de que no este en la misma red, no puedo averiguar la dirección MAC. Por ende, el host debe enviar el mensaje al router:

- Para ello el host averigua la dirección MAC del router, encapsula el datagrama IP que le quiero enviar al destino, dentro de una trama Ethernet cuya dirección MAC address es del router (que tiene una dirección IP).
- El router recibe la trama Ethernet con su dirección MAC como destino. El EtherType es 0800 por lo que lo hay adentro es un datagrama IP. Lo analiza si no esta en su red y envía el mensaje por el camino donde se debería encontrar la dirección destino.
- El destino al recibir la trama, responde a la IP address del host si era una request a gmail por ejemplo, sino no hace nada debido a que TCP/IP no es orientado a la conexión.

## Subredes

### Por que surgen

Cuando se pide un prefijo de red a la IANA (entidad que las asigna), la misma brinda el prefijo de red. Que las direcciones tengan el mismo prefijo de red significa que desde la .1 a la .254, los hosts son vecinos (conectados al mismo enlace físico). Esto podría no suceder, por ejemplo en el caso de que quiera tener una red local y una remota (IANA no va a proveer a la misma empresa otro prefijo de red porque todavía tiene un montón de IP's de la anterior sin utilizar). Para solucionar este problema se “parte” la red para poder implementar dos redes, es decir obtengo subredes.

### Definición

Las subredes son un método para maximizar el espacio de direcciones IPv4 de 32 bits y reducir el tamaño de las tablas de enrutamiento en una interred mayor. En cualquier clase de dirección, las subredes proporcionan un medio de asignar parte del espacio de la dirección host a las direcciones de red, lo cual permite tener más redes. La parte del espacio de dirección de host asignada a las nuevas direcciones de red se conoce como número de subred.

### Cantidad de subredes

Cuando hacemos una partición la mínima cantidad de bits que puedo agarrar para formar subredes es 1bit, formando dos subredes pero, por estándar (creo que igual ya no se le da bola al estándar), la mínima cantidad de bits del ultimo octeto que podemos agarrar para formar subredes es 2 bits, por lo tanto obtenemos 4 subredes (las que empiezan con 00 01 10 y 11) . Si tomo 2bits como mascara de subred, me quedan 6 bits para las direcciones que serian 64 direcciones, de las cuales puedo asignar 62 por subred (la primera es de la subred y la ultima de broadcast).

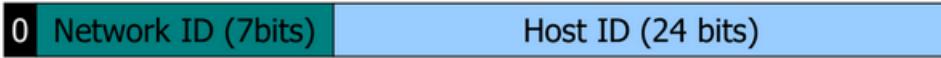
Para formar estas máscaras de 26bits (24 de red y 2 de subred), debo modificar la máscara de la IP a 255.255.255.192 (192 es 8unos.8unos.8unos.2unosy6ceros).

Generadas las subredes, el direccionamiento es el mismo que el explicado anteriormente, se tratan como dos redes distintas.

## Clases de direcciones

Para mejorar el uso de la tabla de direcciones, la **IANA** decidió separar los prefijos de red en bloques **A, B, C, D y E**, siguiendo un orden de **mayor soporte de host a menor**.

CLASE A



CLASE B



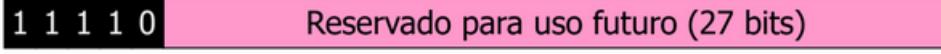
CLASE C



CLASE D



CLASE E



Las tres clases que se pueden utilizar para host son la clase **A, B y C**, mientras que la **D** esta reservada para multicast y la clase **E** para un uso futuro. La capacidad de soporte de host es la siguiente:

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	$2^8(256)$

Debido a que inicialmente las personas solicitaban IP's no para conectarse a internet, sino porque utilizaban **programas que corrían sobre el protocolo IP (uso privado)**, surgió el **RFC1819** para mejorar la asignación de las direcciones.

Para que dos direcciones IP estén dentro de una misma red, su prefijo debe ser igual.

## RFC1819 classfull

### Dirección privada

Se reservo **un bloque A, uno B y uno C para direccionamiento privado** con los cuales **no se puede acceder a internet** porque **NO garantizan la unicidad** (pero si debe ser única cada IP dentro de la red privada), a muchos se les asigna esos bloques.

**Esos bloques son:**

- 10.0.0.0 a 10.255.255.255 (10/8 prefix).
- 172.16.0.0 a 172.32.255.255 (172.16/12 prefix).
- 192.168.0.0 a 192.168.255.0 (192.168/16 prefix).

### Dirección publica

**A los routers se le asigna una dirección publica**, es decir con posibilidad de acceso a internet debido a que no existe la duplicidad. Esta dirección del router nos la asigna (presta) nuestro proveedor de internet que a su vez pidió a la IANA un bloque para sus clientes.

**Accedemos a internet a través de la IP publica del router. La trama viene del dispositivo con la dirección IP privada del mismo, cuando la recibe el router, reemplaza esta dirección privada de mi dispositivo por la IP publica del router y envíá la trama.** El router se guarda el **registro** de esta operación para que, si recibe una respuesta, envíe la misma al host correspondiente.

### Dirección loop

El **bloque 127.0.0.0** de la clase A esta reservado para looping, es decir, **todo lo que se le envíe a ese puerto, me lo devuelve**. También se conoce como **localhost**.

Ejemplo de ejercicio subred

172.30.0.0 /16      ↗ 55 subred  
                         ↙ 1K host

$\begin{matrix} S & H \end{matrix}$   
 000000.00.00000000  
 $\overbrace{2^6 = 64} \quad 2^{10} = 1024 \text{ hosts}$

Mask: 255.255.252.0

### Classless interdomain routing (CIDR) y VLSM

Intenta resolver el problema de tamaño fijo de los bloques de las clases.

Consiste en **olvidar la classfull (la división por clases)** y poder **darle a un cliente la cantidad de direcciones que pida asignándole un segmento de ese tamaño** (por ejemplo si pide 1000 direcciones, se le asigna una dirección de red de /22 quedando 10 bits para los hosts). Es decir, consiste en **máscaras de tamaño variable (VLSM variable length subnet mask)**.

Se implementa este método para todos los bloques restantes (libres) a los ya asignados con el método anterior.

# ARP (Address Resolution Protocol)

## Definición

Es un **protocolo de comunicaciones** de la **capa de enlace de datos**, responsable de **encontrar la dirección de hardware** (Ethernet MAC) que corresponde a **una determinada dirección IP**.

ARP es **imprescindible** para la **transmisión de datos en redes Ethernet** por dos razones: por un lado, las tramas de datos (también **tramas Ethernet**) de los paquetes IP solo pueden enviarse con **ayuda de una dirección de hardware** a los hosts de destino, pero el protocolo de Internet no puede obtener estas direcciones físicas por sí mismo. Por el otro, y debido a su limitada longitud, el protocolo **IPv4 carece de la posibilidad de almacenar las direcciones de los dispositivos**. Con un mecanismo de caché propio, el protocolo ARP también es, aquí, la solución más adecuada. IPv6, por su parte, adopta las funciones del Neighbor Discovery Protocol (NDP).

## Problema de resolución

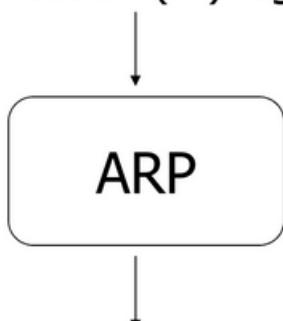
### Problemas:

- Las **aplicaciones de alto nivel solo trabajan con direcciones IP**.
  - Ilusión de una única red virtual.
  - La **comunicación es realizada por redes físicas**, reales.
- Los **datagramas IP son encapsulados en tramas MAC** → se necesitan direcciones de hardware MAC.

### Resolución

- Mapear direcciones IP de alto nivel a direcciones MAC físicas.

Dirección (IP) lógica



Dirección (MAC) física

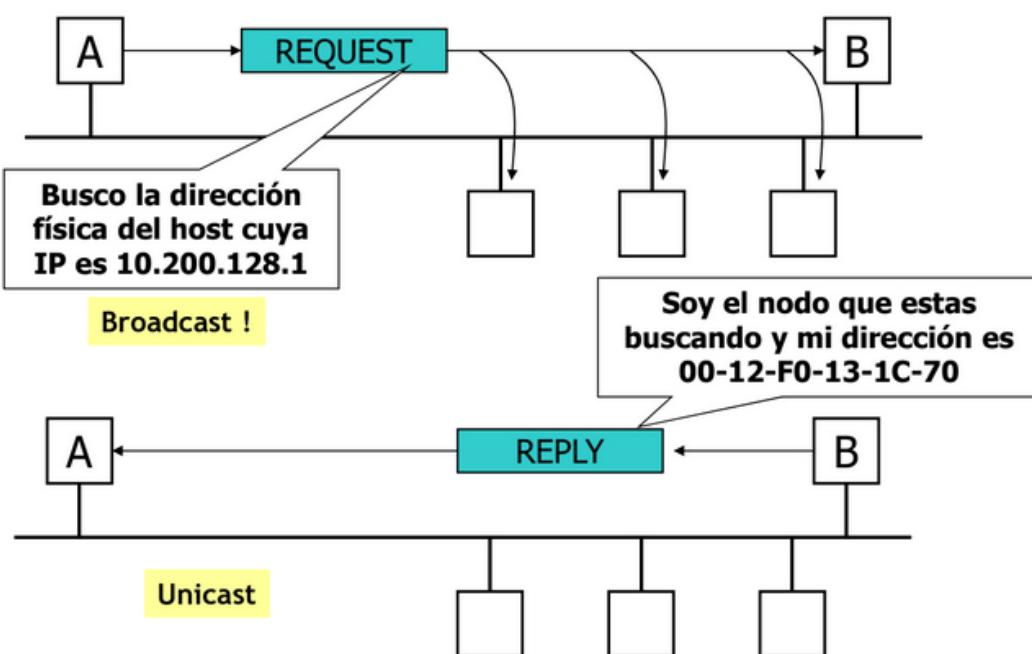
## Como funciona

- A la hora de asignar direcciones por medio del Address Resolution Protocol **hay que distinguir si la dirección IP del host de destino** se encuentra en la **misma red local o en otra subred**. Así, en caso de asignar una dirección MAC a una determinada dirección IP, antes de nada se lleva a cabo una **revisión de la máscara de subred**.

- Si la IP se encuentra en la **red local**, el primer paso es **controlar si ya existe una entrada para ella en la caché del ARP**.
  - Si una **dirección IP ya tiene asignada una dirección física**, es esta la que se utiliza para el direccionamiento.
  - En **caso contrario**, el remitente **envía una solicitud ARP (ARP Request)** con la dirección IP de destino **a todos los hosts de la red**. Para tal fin, el emisor utiliza la dirección de broadcast de ARP FF:FF:FF:FF:FF como dirección del destinatario. **Cada una de las estaciones compara la dirección IP indicada en la petición con las suyas propias** y rechaza la solicitud si no hay coincidencia. Si una estación percibe que **se trata de la dirección propia**, reacciona con una **respuesta ARP (ARP Reply)** en la que, entre otros datos, también **transmite la dirección MAC**. Ambas partes pueden **incorporar la dirección MAC y la IP de la otra parte en la memoria caché**, sentando las bases para el intercambio de datos.
- Si el **host de destino no se encuentra en la misma subred**, el remitente se dirige a la **puerta de enlace estándar** (en la mayoría de los casos un **router**).
  - Puede acceder a ella mediante la combinación de dirección MAC e IP, por lo que aquí también se necesita el Address Resolution Protocol.
  - Una vez resueltas las direcciones, la **puerta de enlace recibe el paquete de datos y a continuación lo envía al host de destino**. Para ello esta pasarela de enlace analiza la cabecera IP para obtener los datos necesarios.
  - El proceso se repite tantas veces como sea necesario hasta que el paquete de datos llegue a su destino o hasta que el campo TTL (Time to Live) haya adoptado el valor 0 en la cabecera IP.

### Entrega directa o unicast

Caso en el que **busca una única dirección MAC**.

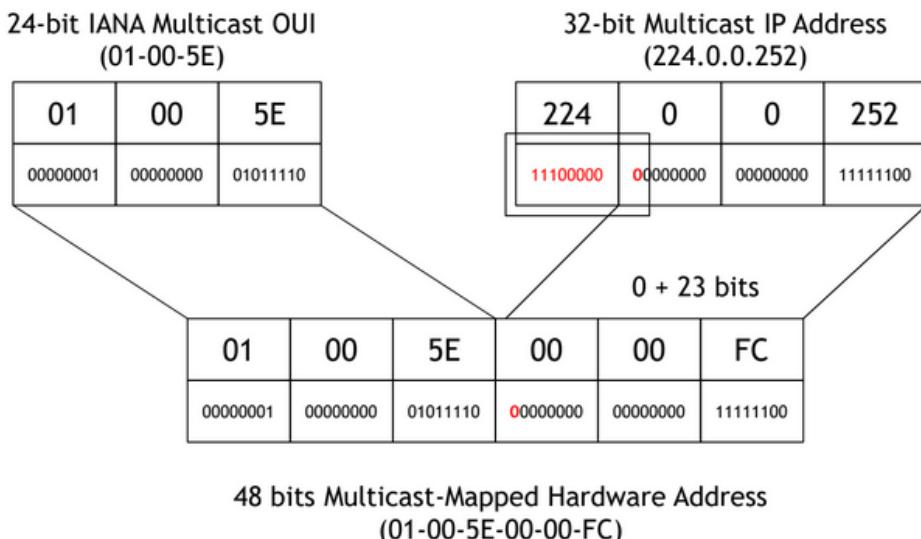


## Entrega broadcast

Caso en el que se envía un broadcast. **No se hace la ARP request**, la dirección IP todos los host, se mapea a la dirección MAC todos los dispositivos (la de broadcast).

## Entrega multicast

Caso en el que el **mensaje va dirigido a un grupo destino** (las direcciones de clase D). **No se hace la ARP request**. Se encapsula el datagrama IP con dirección destino multicast en una trama Ethernet cuya dirección **MAC destino se confecciona a partir de la información de grupo de multicast**. Entonces las estaciones que pertenezcan a ese grupo de multicast van a identificar esa dirección MAC y van a procesar los mensajes. Quienes no pertenezcan lo van a ignorar.



Se combina la **IANA multicast OUI** (identificador único de organización, es un prefijo de la MAC address para grupos) con la **dirección IP multicast** que identifica el grupo (todos los octetos menos el 224 que identifica a multicast en si), **esta combinación conforma la dirección MAC destino**.

## Formato de 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

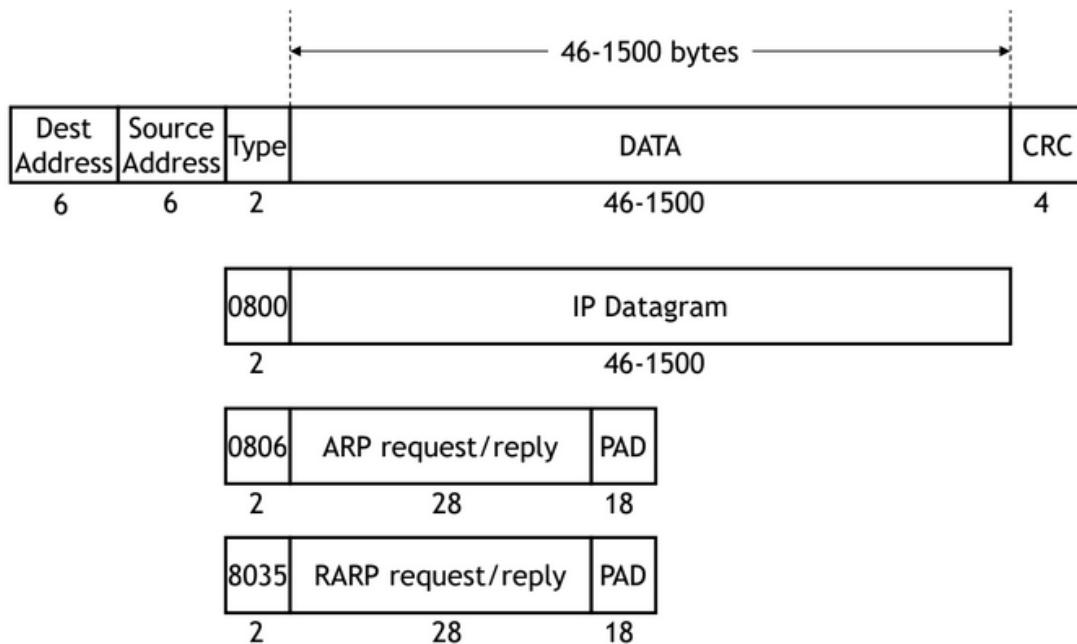
Tiene un **formato dinámico**, porque se aadecua a varios tipos (hardware/protocolo).

En longitud de dirección, el 6 y el 4 es la cantidad de bytes que se utilizan.

En la **request**, la dirección física del destino va en 0 porque es lo que queremos averiguar.

El datagrama va a ir encapsulado en una trama Ethernet cuyo EtherType es 0x0806.

### Estructura de la trama ethernet para arp



Como son solo 28bytes el ARP request/reply y la cantidad mínima de un campo de datos es 46, se rellena con 18bytes.

### Gratuitous ARP

Antes de activar la interfaz con cualquier dirección IP lo que va a hacer el host es enviar un **ARP request gratuito**. Lo llamamos gratuito porque el host pregunta por si mismo (la dirección ip asignada), si alguien responde, hay un conflicto. De esta manera el host detecta si existe alguien con su misma IP.

Es una **variante del ARP**. Son útiles por las siguientes razones:

- **Permiten detectar conflictos en IP** (principal utilidad, por ejemplo cuando sale el "la dirección IP está en uso").
- **Actualizan el contenido del cache ARP.**
- **Informan a los switches** el MAC del cliente conectado.
- **Sucede a cada cambio de estado de la interfaz** → indicador de problemas.

## Routing Protocols

Es el **proceso de encaminamiento**. El proceso de ruteo ocurre en la **capa 3** (capa de red) del modelo OSI.

El **router** para enviar los datagramas al siguiente salto o hop **realiza dos funciones** básicas:

- **Determinar el mejor camino a destino:** Esta tarea consiste en **revisar** todos los caminos disponibles a la red destino **y elegir** el camino mas optimo.
  - Para **determinar el mejor camino**, se fija en la topología de la red, información que tiene almacenada en una tabla local llamada tabla de ruteo. Se fija la network destination que tenga mayor bits de coincidencia (si tiene algún bit que no va, no sirve; si no puedo usar ninguna uso la 0.0.0.0 pero es la mas larga). También se fija en la métrica (cuanto menor es el numero, mas confiable es la interface), sirve para desempatar la network destination.
- **Conmutar el datagrama:** Consiste en **cambiar la dirección destino física de la trama, por la del próximo salto.**
  - En caso de que **el router se encuentre en la red destino**, pregunta mediante un ARP quien es la dirección destino, la estación responde. **Encapsula el datagrama en una trama Ethernet** con la dirección MAC que le acabaron de dar y entrega el datagrama.

### Objetivos de un RP

Los **objetivos de diseño** de un protocolo de ruteo son:

#### Flexible

**Rápida adaptación a los cambios** en la topología de la red.

Cuando una red deja de estar disponible, el protocolo debe detectarlo y determinar el próximo mejor camino hacia esa red. Cuando la red vuelve a estar disponible, debe **actualizar su tabla para reflejar el cambio**.

Un protocolo flexible puede **adaptarse a cambios en las variables de red**, tales como ancho de banda y retardo.

#### Optimo

La optimidad de un protocolo consiste en la habilidad para **elegir la mejor ruta**. Está **directamente relacionada con la métrica** que utiliza **para calificar sus rutas**.

Un protocolo puede utilizar solo el número de saltos como métrica, mientras otro puede utilizar una combinación de estos y el retardo de la red.

#### Rápida convergencia

Convergencia: La **convergencia** se refiere al tiempo que tardan todos los routers de la red en **actualizarse** en relación con los cambios que se han sufrido en la topología de la red.

La convergencia ocurre cuando **todos los routers dentro de una red poseen tablas de ruteo consistentes**. Cuando ocurre un evento, todos los routers deben re-calcular las rutas óptimas. En ese momento existen inconsistencias en las tablas de ruteo y pueden producirse “routing loops”.

## Robustez

Un protocolo robusto es aquel que mantiene su **correcto funcionamiento aun en condiciones inusuales o impredecibles**:

- Alta utilización de los vínculos.
- Falla de hardware.
- Configuraciones incorrectas.

## Simplicidad

La simplicidad de un protocolo se refiere a la **habilidad de operar eficientemente**. Se busca que **consuma la menor cantidad de recursos posibles al router**.

Los protocolos obtienen y almacenan información de rutas. De esta manera compiten por los recursos físicos y limitados de un router. Un protocolo simple debe operar con el **mínimo impacto (overhead)**.

## Clasificación

- **Estáticos / Dinámicos:** Dependiendo si **conoce como llegar a un destino solo o si hay que especificarlo**.
- **Single-Path / Multi-Path:** Característica de los dinámicos.
  - Un protocolo de ruteo es **Single-Path** si ante dos caminos posibles para alcanzar una red destino, siempre va a elegir uno (solo define una ruta para comunicar un nodo origen con un nodo destino).
  - Un camino **Multi-Path** va a identificar estos dos caminos posibles, y es capaz de hacer **balanceo de carga**.
- **Flat / Hierarchical:**
  - Un **ruteo plano** hace referencia a que todos los **nodos** se encuentran en el **mismo nivel de jerarquía y todos intercambian información de enrutamiento**. Tiene problemas para escalar el proceso de ruteo.
  - Un **ruteo jerárquico** establece **grupos jerárquicos** al rededor del **backbone**. Designa grupos lógicos llamados **dominios, sistemas autónomos o áreas**. **Algunos enrutadores pueden comunicarse entre dominios y otros solo con su dominio**. Permite escalar.
- **Interior / Exterior.**
  - Un protocolo **interior** es aquel que **corre dentro de mi red**, bajo routers de mi administración y es el que utilizo para interconectar diferentes routers.
  - El protocolo **exterior** es un protocolo de ruteo diseñado para que **dos entidades o compañías se comuniquen entre si**. El mas conocido es el **RGP**.
- **Distance Vector / Link State:** Son dos formas de calcular la mejor ruta.

- **Link state:** intercambio de información de costos de enlace con **todos los routers**. Tiene la **configuración completa de la red**. Ej OSPF.
- **Distance vector:** Intercambio de información con los vecinos. Los nodos mantienen un vector por enlace para cada red conectada directamente. Ej RIP.

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

## RIP

- **Distance-vector, interior gateway protocol.**
  - Con “split horizon” y “poison inverse”.
- Optimiza la métrica.
  - Utiliza como **métrica el hop count (máximo 15 hops)**, tamaño de red limitado).
- Encapsulamiento en **datagramas UDP**.
  - Puerto 520.
  - **Entrega “no confiable”**.

## Funcionamiento



- Cada 30 segundos, envíá la tabla de ruteo completa a sus vecinos.
- Si una ruta no es actualizada en 3 minutos, su métrica es seteada a infinito (es decir, **inactiva**), y se informa a los vecinos.
- El borrado de una ruta de la tabla de ruteo, se demora 2 minutos.

**Inicialización:** Enviá un **request a todos los vecinos** (broadcast) **solicitando sus tablas** de ruteo completas. **No realiza Neighbor Discovery**, envíá broadcasts y no recibe confirmación.

**Confiabilidad:** Se basa en la **retransmisión periódica de toda la información**

**Subredes (versión 2):** Incluye información de subred en la tabla de ruteo y la informa en las actualizaciones a sus vecinos.

**Seguridad (versión 2):** Password opcional de 16 bytes (cleartext). Evita la existencia de black-holes (routers que informan todas las redes con métrica 0). Es fácil de quebrar.

## OSPF

Open (no propietario) Shortest Path First.

- Protocolo **interior** recomendado para TCP/IP.
  - **Link state**, utiliza el algoritmo de Dijkstra (arma un **grafo de la red** formando la topología).
- **Ventajas**
  - **Converge más rápido que RIP.**
  - **Intercambia menos información que RIP**, solo intercambia información cuando hay **cambios en la topología**.
- **Corre directamente sobre IP** (no UDP/TCP) protocolo número 89.

**Métrica optimizada:** Utiliza para el calculo de la métrica **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.

**Subnets:** Diseñado para trabajar con **VLSM y CIDR**.

**Seguridad:** Contraseña simple cleartext. MD5 – preshared key.

## Areas

RIP es un sistema plano que podía tener hasta 15 saltos.

En el caso de OSPF **se pueden configurar áreas**. En **cada área los routers corren una misma instancia de OSPF e intercambian información de topología entre ellos**.

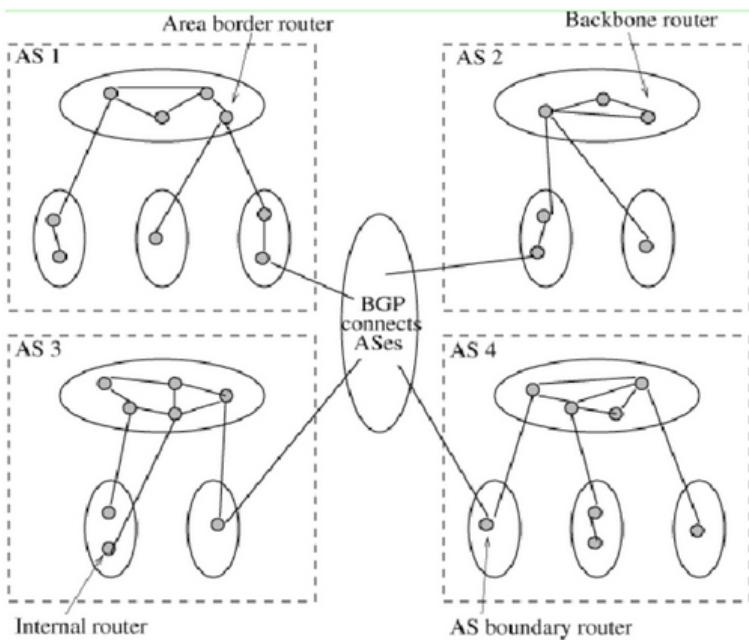
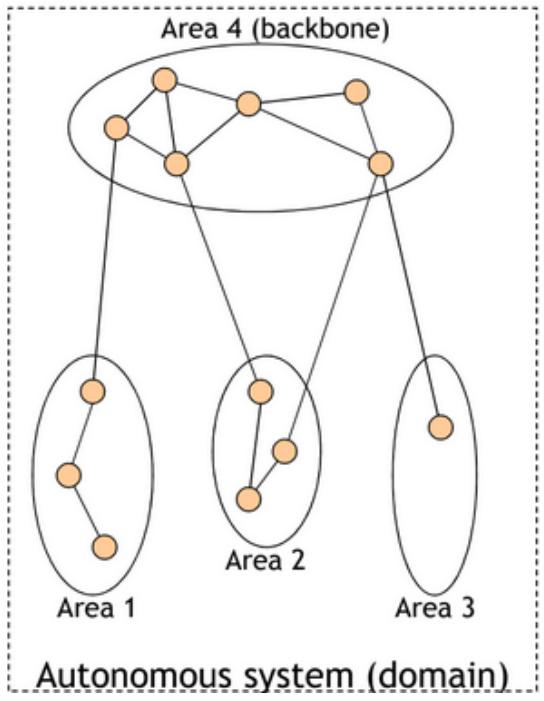
Luego hay un **router designado** que es el que se va a **conectar** con lo que se conoce como **área de backbone** y ahí si se va a consolidar la información de topología que después se va a distribuir a los vecinos.

**Fuera de un área, su topología y detalle no son visibles. Las rutas entre las diferentes áreas circulan siempre por el backbone, por lo tanto todas las áreas deben conectar con el backbone.**

Posee una **jerarquía de 2 niveles**. Permite mantener pequeñas las bases de SPF.

**Cada área corre una copia de Link-State protocol.**

**Los routers de borde realizan summarización de rutas e intercambian menos información.**



**BGP es el protocolo que permite la comunicación de distintos sistemas autónomos (de distintos proveedores).**

## ICMP

Internet Control Message Protocol (**Protocolo de mensajería de control de internet**).

Surge para **intentar resolver algunas incertidumbres de IP** originadas por no estar orientado a la conexión.

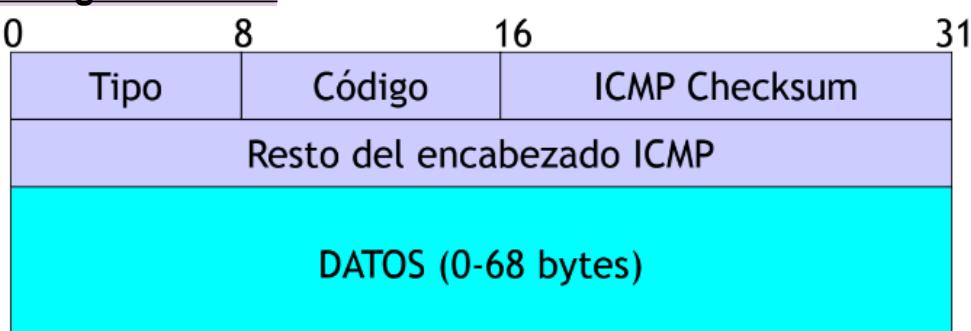
Para **intercambiar datos de estado o mensajes de error**, los nodos recurren al Internet Control Message Protocol (ICMP) en las redes TCP/IP. Concretamente, los servidores de aplicaciones y las puertas de acceso como los **routers**, utilizan **esta implementación del protocolo IP para devolver mensajes sobre problemas** con datagramas al remitente del paquete.

Por definición, ICMP es un protocolo autónomo aun cuando los diferentes mensajes están incluidos en paquetes IP tradicionales. Para tal fin, el protocolo de Internet trata a la implementación opcional como un protocolo de capas superiores. Los diversos servicios de red que se suelen utilizar hoy en día, como traceroute o ping, se basan en el protocolo ICMP.

### Que hace ICMP

- **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 que acción correctiva tomar**.

### Datagrama ICMP



**Tipo:** Informa que **tipo de mensaje de error** se está generando.

**Código:** Código del mensaje de error.

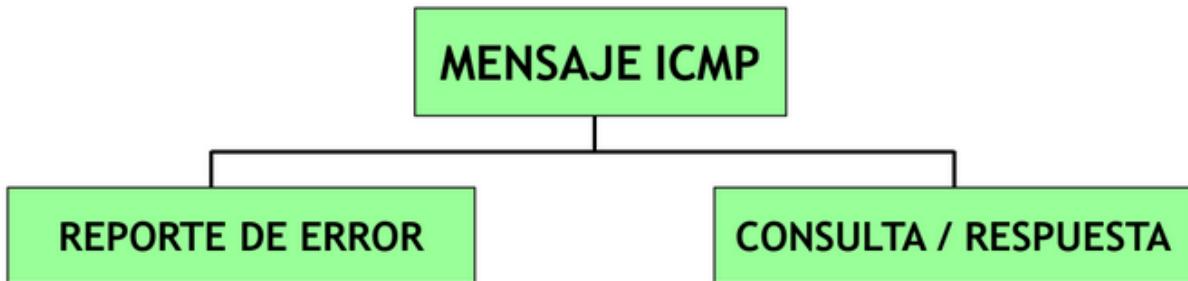
Por ejemplo, un mensaje ICMP del tipo 3 indica que no se ha alcanzado el objetivo del paquete de datos, mientras que el código de este dato precisa y ofrece información acerca de si la red de destino (0), el host deseado (1) o el puerto esperado (3) no ha respondido a la solicitud.

**ICMP Checksum:** Cubre todo el datagrama, garantiza la exactitud del mensaje.

**Datos:** Generalmente contiene

- **Encabezado IP del datagrama** que causó el error.
- **Primeros 8 bytes de datos** de este datagrama.
- Información necesaria para **identificar la raíz del error**.

## Tipos de mensaje ICMP mas utilizados



-**Destino inalcanzable**

-**Tiempo de espera agotado**

-Router redirect

-Problema de parámetro

-**Echo request/reply**

-Timestamp request/reply

-Router discovery

### Destino inalcanzable

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

#### Motivos:

- **Network unreachable:** esto nunca sucede si tengo el router default 0.0.0.0 seteado, significa que el **router no sabe como llegar a la red**.
- **Host unreachable:** sucede cuando el **router de la red destinataria envía un ARP request y nadie contesta**.
- **Protocol (TCP/UDP) not enabled.**
- **Port not bound to a service:** Puerto no vinculado a ningún servicio, respuesta cuando se trata de pegar a un puerto (ej 8080) y **no hay ningún servicio corriendo en el**.
- **Fragmentation needed, but DF flag set.**
- **Source route failed.**

### 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**.

### Echo Request / Reply

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

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

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

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

Si se envía un ping utiliza un echo request y echo reply.

Traceroute: me permite identificar si el ruteo es correcto (ver no llego, o si llegó y no me respondió, etc). Se suele utilizar luego de un ping fallido.

# DHCP

**Dynamic Host Configuration Protocol** (Protocolo dinámico para configuración de host).

Conectar dispositivos a una red TCP/IP ya no es tan difícil como antes, porque en lugar de tener que asignar las direcciones IP manualmente e introducirlas en los diferentes sistemas, hoy la gestión de direcciones tiene lugar automáticamente. Si los routers, hubs o conmutadores pueden asignar de forma automática una dirección individual a los dispositivos que solicitan conectarse a una red, es gracias al protocolo de configuración dinámica de host.

## Funciones

- Derivado de BOOTP, protocolo que **permite la inicialización de computadoras sin disco rígido.**
- **Centraliza y administra la asignación de direcciones IP.**
- **Mantiene un registro de la IP asignada a cada cliente.**

## En que consiste

La asignación de direcciones con DHCP **se basa en un modelo cliente-servidor**: el terminal que quiere conectarse **solicita la configuración IP a un servidor DHCP** que, por su parte, recurre a una base de datos que contiene los parámetros de red asignables. Este **servidor**, componente de cualquier router ADSL moderno, **puede asignar los siguientes parámetros** al cliente con ayuda de la información de su base de datos:

- **Dirección IP única.**
- **Máscara de subred.**
- **Puerta de enlace estándar.**
- **Servidores DNS.**
- **Configuración proxy** por WPAD (Web Proxy Auto-Discovery Protocol).

Todo host debe poseer una dirección de IP única, una máscara de subred, un default gateway que le permita salir de la red y un servidor DNS (para resolución de nombres).

**Default gateway o puerta de enlace:** es la **dirección IP del enrutador que conecta la red interna a una red externa** (interconecta redes). Se utiliza en hosts o terminales.

**Default route:** Se utiliza en routers. También conocida como ultimo recurso del gateway. Es usada para paquetes cuya dirección de destino no coincide con ninguna entrada en la tabla de enrutamiento del router. Es una ruta estática en el router en la cual se indica la IP mas baja posible (por ej 0.0.0.0, todas matchean con esta) y la IP del otro router por la cual acceder a ella o que va a saber como ir (por ejemplo el del ISP) o la interfaz por la cual salir .

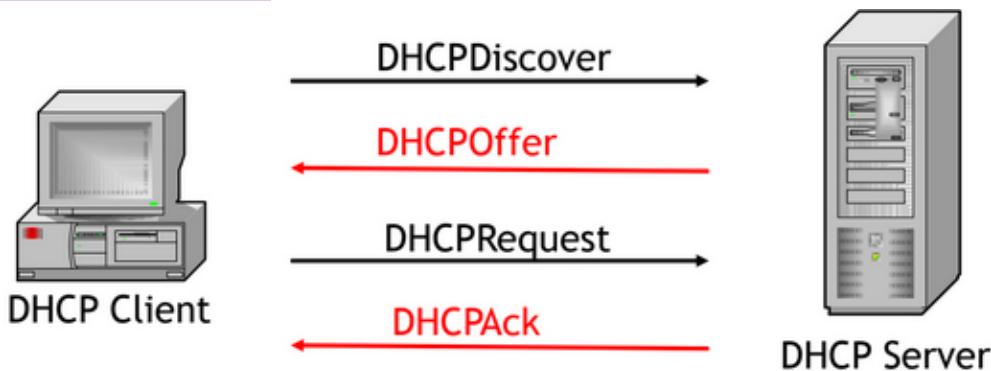
## Asignación estática vs dinámica de IP

**Ventajas de la asignación dinámica:**

- **Elimina la necesidad de llevar un registro** de direcciones asignadas (facilita la administración).

- **Facilita la modificación del espacio de direcciones de una red.** Cuando decido hacer subnetting de una red a la que ya le había asignado hosts, debo reconfigurar todas las IP (etc) de esos host, si estoy usando DHCP solo debo cambiar la mascara y el servidor va a repartir direcciones del nuevo rango, liberando el rango que quería tener vacante.
- Permite la **utilización eficiente de un espacio de direcciones** reducido (más hosts que direcciones IP disponibles – ISP). Puedo **asignarle dirección al host cuando se conecte** en vez de que tenga una fija todo el tiempo (es decir, ocupandola sin usarla).
- **Elimina la existencia de errores** en la configuración (**humanas**).
- Permite asignar a cada host todos los parámetros de configuración junto con la dirección IP.

## Funcionamiento



### 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 (no es un broadcast de subred, es un **broadcast total**). El cliente establece contacto con todos los integrantes de la red **con el propósito de localizar servidores DHCP disponibles e informar sobre su petición**.
- **Posee la dirección física del cliente y el nombre del host.**

Este proceso se **realiza cuando**:

- 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 la siguiente información:**

- **Dirección de hardware del cliente.**
- **Dirección IP destino 0.0.0.0.**
- **Una dirección IP ofrecida.**

- La máscara de subred.
- **Duración de la asignación (Lease).** Las IP no son fijas, por ejemplo te da esa IP por un día. Un admin puede darte tiempo infinito, siempre que te conectes a esa red vas a tener esa IP (para por ejemplo si la IP tiene asignado privilegios de Firewall).
- Una **identificación del servidor** (dirección IP).

El **DHCP Server reserva la dirección IP ofrecida** (junto al MAC address del host). El cliente DHCP selecciona la dirección IP de la primer oferta recibida.

### DHCP Request / IP lease selection

Luego de recibir al menos una oferta de un DHCP server, el **cliente envía un broadcast (DHCPRequest)** a todos los servers **indicando la oferta aceptada**.

El mensaje se envía como un request (DHCPRequest), **indicando la dirección IP del servidor** cuya oferta se está aceptando, de esta manera el **resto de servidores** también reciben este mensaje de forma que **quedan informados de la elección**.

Esta respuesta **también sirve para confirmar parámetros asignados** con anterioridad.

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

### DHCP Pack / IP lease acknowledge

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

Este mensaje **contiene la dirección asignada y otros valores de configuración**.

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

Si el servidor no contara con la dirección, entonces respondería con **DHCPNAK** (DHCP not acknowledged o «no reconocido»). Esto puede suceder por dos **razones**:

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

### Otros mensajes

**DHCPDecline:** El cliente indica al servidor que la dirección está en uso (lo sabe mediante **Gratuitous ARP**). Puede suceder si otro host tiene esa IP de manera estática (el DHCP server no tiene idea).

**DHCPRelease:** El cliente libera la asignación, cancelando el lease.

**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 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.

En caso de no recibir respuesta del servidor, cuando va el **75% del tiempo vuelve a hacer el intento de renovación** (2da vez y luego lo intenta una tercera vez (en 87,5% creo)). Una vez excedido el **87,5% del tiempo sin respuesta, o recibido un NACK, se inicializa el proceso DHCP**.

## Condiciones de diseño

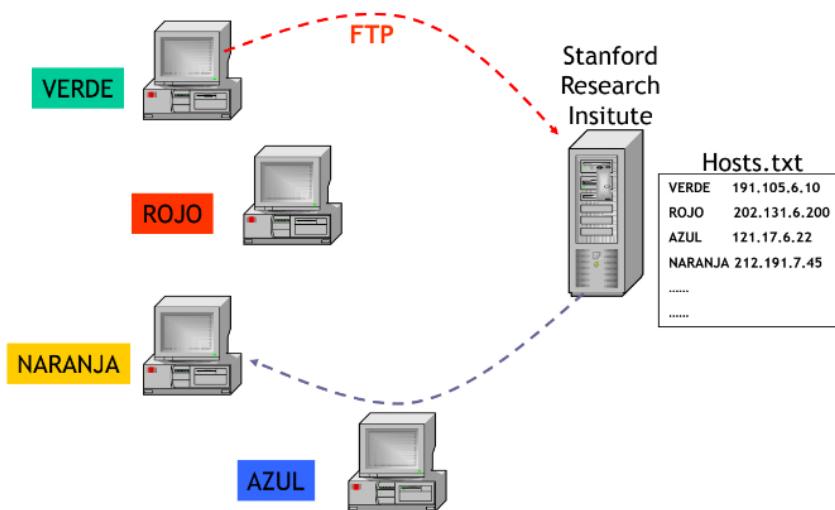
- Es común dividir el espacio de direcciones disponibles en **2 DHCP servers para aumentar la disponibilidad** (si un cliente no recibe una IP, no puede operar en la red). Considerar que el **rango de IPs se debe dividir entre los servers para que no asignen las mismas** (no comparten la memoria).
- El **pool de direcciones** que reparte el DHCP server **excluye un rango de IP reservado para asignación estática** (routers, impresores, etc).
- Es necesario configurar los routers para permitir el paso de DHCP requests (broadcast).
- DDNS.

## DHCP options

- 1 – Subnet mask.
- 3 – Router Option.
- 6 – Domain Name Server.
- 33 – Static route option (default gateway).

# DNS

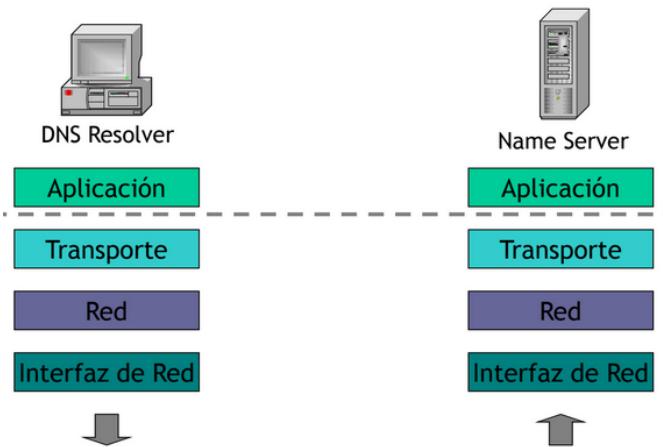
## Domain name system.



El Sistema de Nombres de Dominio o DNS es un **sistema de nomenclatura jerárquico** que **se ocupa de la administración del espacio de nombres de dominio** (Domain Name Space). Su labor primordial **consiste en resolver las peticiones de asignación de nombres**. Traduce un dominio en una IP. Por ejemplo la ip de [www.ejemplo.es](http://www.ejemplo.es) es 93.184.216.34.

El sistema es cliente-servidor y está compuesto por:

- **Resolvers:** envía el pedido de resolución entre la aplicación y el servicio de nombres. Es el cliente.
- **Name servers:** reciben el pedido y resuelven el nombre de Host a una dirección IP. Es el servidor.



## Espacio de nombres jerárquicos

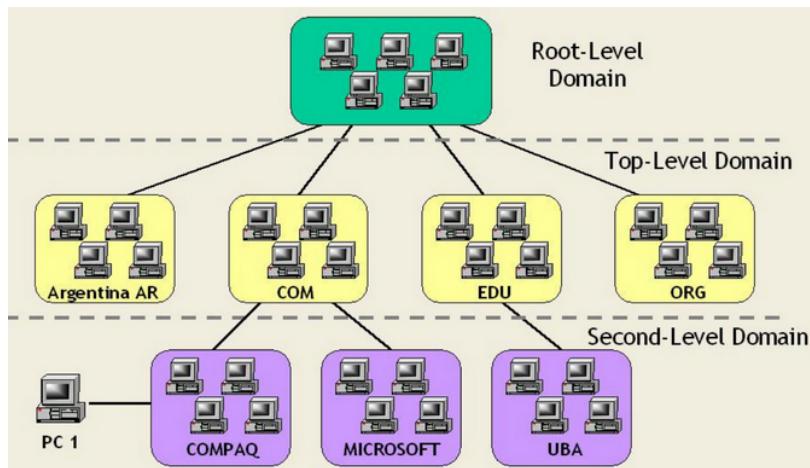
El espacio de nombres esta **construida en una estructura jerárquica de árbol**.

**Root-level domain:** es lo más alto de la jerarquía. Este nivel no utiliza una etiqueta, pero puede identificarse con el ". ". En realidad, aunque no lo escribamos, todo dominio tiene un punto al final. Por ejemplo "google.com." .

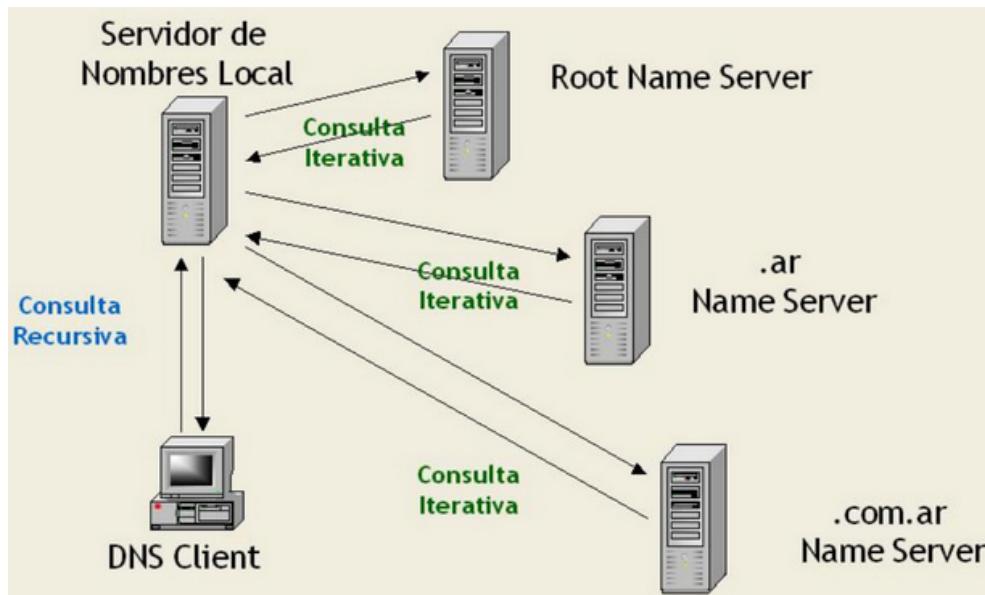
**Top-level domain:** es como termina, por ejemplo com, edu, ar, etc.

**Second-level domain:** pueden contener host u otros dominios llamados sub-dominios. Por ejemplo el dominio microsoft.com puede poseer hosts: [ftp.microsoft.com](http://ftp.microsoft.com) o subdominios como dev.microsoft.com. Para dar de alta un dominio se debe ir al proveedor top-level y registrar mi dominio.

**Host names:** Es el **nombre completo**, se le suele decir **Fully Qualified Domain Name (FQDN)**. Ejemplo sistemas.frba.utn.edu.ar



## Resolución de nombres



Si no tiene cacheado el dominio el servidor de nombres local, le pregunta por ejemplo por sistemas.utn.edu.frba.ar a root name server, este no lo sabe pero le dice que se fije en .ar name server y así (esto es una consulta iterativa).

### Tipos de consultas:

- **Consulta recursiva:** El servidor de nombres consultado está **obligado a responder con los datos o con un error**. No me da una “pista” como la iterativa. La hace el terminal al servidor de nombres local. Puede tener cacheada una respuesta o empezar a hacer consultas iterativas a otros servidores.
- **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 dada.

## Zonas de autoridad

Hay un servidor que es responsable del archivo de zonas.

Es una **porción de dominio por la cual un servidor es responsable**. Por ejemplo por edu.ar.

El DNS server responsable de la zona posee el archivo de la zona X que contiene la asociación nombre → IP para ese dominio.

**Un único server puede mantener múltiples zonas.**

## Roles de los servidores DNS

- **Primary name server:** Los archivos de información de la zona se **almacenan localmente**.
- **Seconday name server:** Obtiene la información de zona de master name server. **Replican para evitar un único punto de falla.**
- **Master name server:** Fuente de información para un secondary server. **Puede ser Primary o Secondary servers.**
- **Caching only:** No almacena información de zona. Lo que hacen es **atender a consultas**. Cachean las respuestas durante un tiempo que se lo da el servidor que le dio esa asociación.

## Caching y TTL

Los **DNS Servers cachean las consultas iterativas**. Cada entrada en cache tiene asociado un **tiempo de vida (TTL)**, cuando este expira, la **entrada es borrada**.

El **TTL remanente es enviado al resolver** cuando se responde una consulta recursiva.

## Protocolos y puertos

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

En el archivo que esta almacenada la asociación de nombre e IP hay diferentes entradas:

- **Host Record (A):** asocia estéticamente un **nombre de un host con una dirección IP**. Comprende la mayor parte del archivo y lista todos los hosts dentro de la zona. Ejemplos:
  - www IN A 200.69.225.145: dice que www. se traduce a esa IP.
  - rhino IN A 200.26.65.12: dice que rhino se traduce a esa IP.

- **Mail exchange (MX):** asocia un **dominio de email con la dirección de los servidores de correo.** Ejemplos:
  - @ IN MX [10] mailhost: le pregunta la ip al servidor DNS de un registro MX de tal casilla mailhost.
  - @ IN MX [20] mail1.infovia.com.ar
- **Canonical name (CNAME):** permiten asociar **más de un nombre de Host a una única dirección IP (alias).** Ejemplos donde muestra que puedo conectarme a Rhino con esos dos alias.
  - Fileserver1 CNAME Rhino
  - ftp CNAME Rhino

## **Registro de nombres**

La administración local de TLD .ar lo realiza Cancilleria (MRECIC) en [www.nic.ar](http://www.nic.ar).