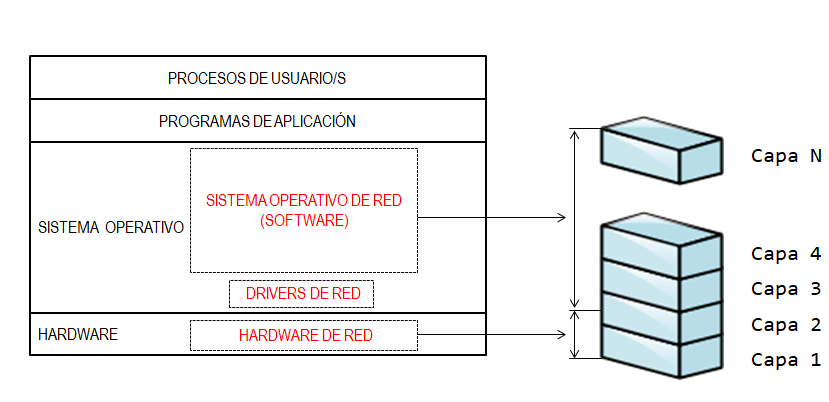
**Redes y Comunicaciones**

**HISTORIA Y EVOLUCIÓN DE LAS REDES**

El nacimiento de las redes, su concepción y filosofía de diseño estuvo marcada por la necesidad inmediata. Por ejemplo, una vez inventado el teléfono surgió la necesidad de interconectarlos y de este modo surgió la Red Telefónica. El objetivo fue transportar voz y el diseño y evolución de la Red Telefónica fue para este fin especifico. Inicialmente el ámbito geográfico fue acotado y sirvió para enlazar pocos usuarios acotados a nivel geográfico.

El mismo fenómeno ocurrió a partir de los avances en las tecnologías de los circuitos integrados y la fabricación del primer microprocesador en 1971 (INTEL) La factibilidad de fabricar computadoras personales a bajo costo era un hecho y la escala a nivel de mercado era todo el mundo. Distintas empresas comenzaron a construir distintos prototipos de computadoras personales y redes de área local para interconectarlas. Al igual que en las redes telefónicas las primeras redes de computadoras interconectaban pocas unidades en espacios reducidos (LAN Local Area Network – Redes de Área Local). Cada una de ellas servía a los propósitos de la organización propietaria de la red y no existían estándares. Proliferaron distintas tecnologías de red (incompatibles entre ellas) que competían entre sí para imponerse en un mercado de demanda creciente.

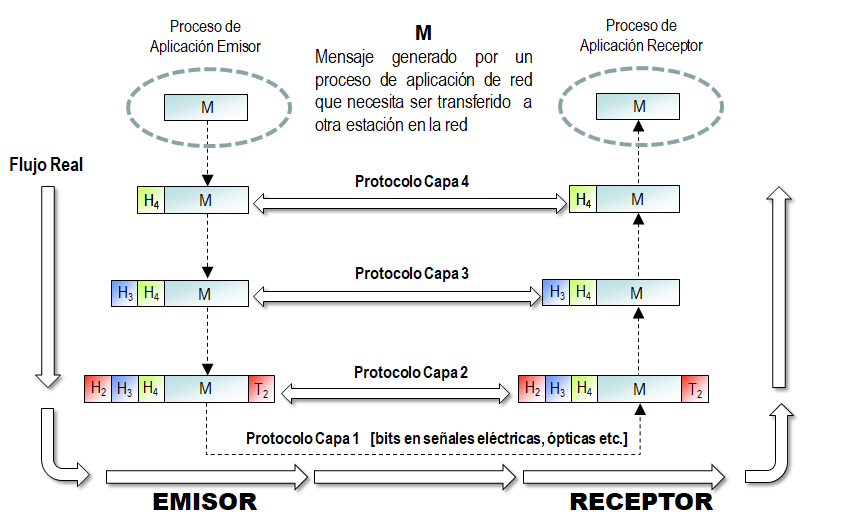
Entonces las redes fueron diseñadas e implementadas de acuerdo a las necesidades puntuales y ello originó una red especializada para cada tipo de tráfico (voz, datos, video, etc). La aparición de nuevos servicios de red fue el factor que hizo que tuvieran que ser continuamente reinventadas o rediseñadas. En la actualidad existen las redes de “servicios convergentes”. Una sola red para todos los servicios (voz, datos, TV, etc.).

La operación de las redes de computadoras modernas es un proceso muy complejo que envuelve la interacción de muchos subsistemas. Para reducir la complejidad de diseño que implica un sistema monolítico, las redes modernas están organizadas en forma modular e incluyen un número determinado de capas. El nombre de las capas, el número de capas, y su función, difieren dependiendo del tipo de arquitectura de red 

El propósito de cada capa es ofrecer ciertos **servicios** a la capa superior aislándola de los detalles de su implementación

En la figura, la Capa N utiliza los servicios que brinda la Capa N-1 sin preocuparse de los detalles de su implementación; por lo tanto, la Capa N-1 presta servicios a la Capa N. Del mismo modo la capa N-1 utilizara los servicios provistos por la capa N-2, y así sucesivamente.

La capa “N” mantiene una conversación con la capa “N” de la entidad remota y las reglas y convenciones usadas en este diálogo entre capas homólogas se conoce como **PROTOCOLO DE LA CAPA “N”.**



Una capa agrega un header a la información proveniente de un proceso de aplicación de red o de una capa superior. El *header* agregado por la Capa N es procesado en la Capa N homóloga y eliminado antes de ser transferido al proceso de aplicación receptor o a una capa superior.El diálogo entre capas homólogas se produce a través del *header* insertado por la capa (en la figura H2).

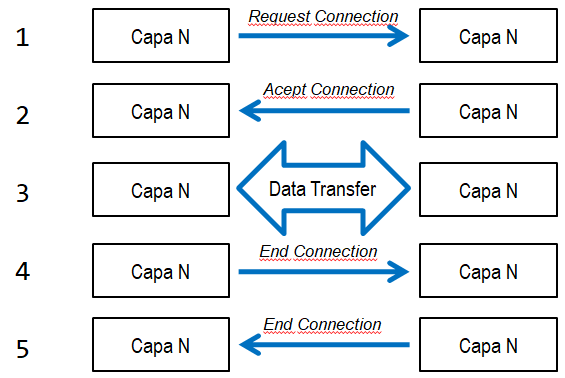
Al conjunto de Capas y Protocolos que definen una red se lo denomina “ARQUITECTURA DE RED” : Una Arquitectura de Red no incluye los detalles de implementación en cada una de las capas. Ello implica que distintas implementaciones de protocolos debidas a distintos fabricantes pueden responder a la misma Arquitectura de Red.Interoperación de redes a pesar de distintas implentaciones en cada una de las capas..Una capa puede brindar distintos servicios a una capa superior,por lo general una capa puede ofrecer todos o algunos de los siguientes tipos de servicios:

* Servicio Orientado a la Conexión
* Servicio sin Conexión sin reconocimiento
* Servicio sin Conexión con reconocimiento

La capa N escoge el tipo de servicio a utilizar en la capa N-1 a través del SAP (Service Access Point) correspondiente o puntos de acceso a servicio.

Un servicio orientado a la conexión (*service oriented connection*) posee las siguientes características básicas

1. Se establece una conexión
2. Se envían y reciben unidades de información numeradas
3. Se realiza control de errores, control de flujo, control de secuencia y control de congestión (si corresponde)
4. Se cierra la conexión

Recordar que el sevicio es ofrecido por una capa a la capa superior.

Un servicio orientado a la conexión en la capa N-1 garantiza a la capa N que las unidades de información llegaran:

* En orden de secuencia
* Libre de errores
* Libre de duplicados

Entonces la capa N ve una red simplificada a un solo enlace ideal (libre de errores) merced al servicio orientado a la conexión brindado por la capa N-1

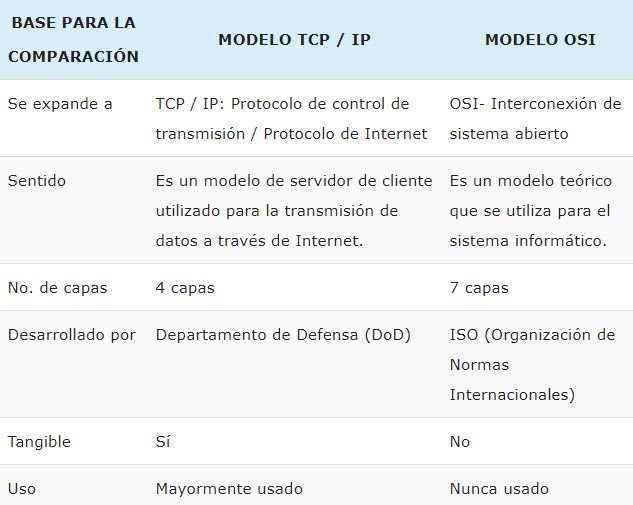
Sí una capa brinda un servicio sin conexión, los paquetes pueden: Perderse, duplicarse o llegar fuera de secuencia. No existen fases de apertura y cierre de la conexión por ende es mucho más eficiente que el servicio orientado a la conexión (¿siempre?). Hay algún mecanismo de detección de errores en los mensajes recibidos pero no se intenta recuperarlos.

Cada capa debe tener una función o un conjunto de funciones bien definidas y no duplicadas en otras capas,el número óptimo de capas debe ser aquel que minimice el flujo de información entre capas.Para determinar la función de cada una de las capas es necesario conocer a grandes rasgos la problemática que debe resolver una arquitectura de red genérica

Función de cada capa desde una perspectiva histórica:

* Capa Física
* Capa de Enlace de Datos
* Capa de Transporte
* Capa de Red
* Capa de Aplicación

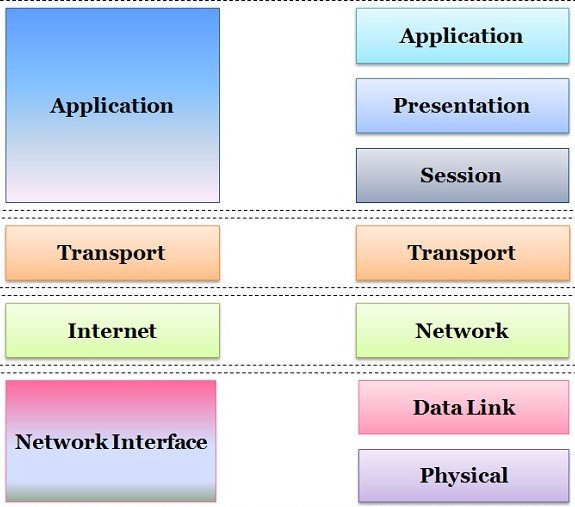
**Diferencia entre TCP/IP-OSI**



**Diferencias**

El modelo [TCP](https://pc-solucion.es/2018/04/04/tcp/)/[IP](https://pc-solucion.es/2018/04/10/ip/) fue desarrollado antes que el modelo OSI, y por lo tanto, las capas difieren. Con respecto al diagrama que vemos a continuación, se ve claramente que el Modelo TCP/IP tiene cuatro capas que son, Interfaz de Red, Internet, Transporte y Capa de Aplicación.

La capa de aplicación de TCP/IP es una combinación de la capa de sesión, presentación y aplicación del modelo OSI.



**Diferencias clave principales entre los modelos TCP/IP y OSI**

* **TCP/IP es un modelo cliente-servidor**, es decir, cuando **el cliente solicita el servicio es proporcionado por el servidor**. Mientras que OSI es un modelo conceptual.
* TCP/IP es un protocolo estándar utilizado para todas las redes, incluida Internet, mientras que **OSI no es un protocolo sino un modelo de referencia utilizado para comprender y diseñar** la arquitectura del sistema.
* TCP/IP es un modelo de 4 capas, mientras que OSI tiene 7 capas.
* TCP/IP sigue el enfoque vertical. Por otro lado, el Modelo OSI soporta el enfoque Horizontal.
* TCP/IP es tangible, mientras que OSI no lo es.
* TCP/IP sigue un enfoque de arriba hacia abajo, mientras que el modelo OSI sigue un enfoque de abajo hacia arriba.

**En Resumen**

Podemos resumir con que el modelo TCP/IP es fiable sobre el modelo OSI, **TCP/IP se utiliza para la conexión de extremo a extremo con el fin de transmitir los datos a través de Internet**. TCP/IP es robusto, flexible, tangible y también**sugiere cómo deben enviarse los datos a través de la web**. La capa de transporte del modelo TCP/IP comprueba si los datos han llegado en orden, si tiene un error o no, si se envían paquetes perdidos o no, si se recibe acuse de recibo o no, etc.

**Capa FIsica:**

Es la capa íntimamente ligada al medio físico

Su función principal es:

* Convertir bits en señales electromagnéticas susceptibles de ser transmitidas por el medio físico particular (pares de cobre, cable coaxial, fibra óptica, aire, etc)
* En el lado receptor, convertir señales electromagnéticas en bits

Implementar un código de línea apropiado a los efectos de:

* + - * No superar el ancho de banda disponible en el medio de transmisión
      * Facilitar la sincronización entre emisor y receptor
      * Proveer algún nivel de inmunidad ante condiciones de interferencia electromagnética

**Capa de Enlace de Datos:**

Debe poder prestar un servicio de comunicación confiable a la capa superior

Los medios de comunicación son imperfectos y susceptibles a interferencia electromagnética (ruido)

* + Control de errores, control de secuencia y control de flujo
  + Asignar una dirección a cada una de las estaciones en el enlace
  + En una red como la ilustrada debe proveer reglas para el acceso al medio

**Capa de Transporte:**

Capa encargada de efectuar el transporte de los datos (que se encuentran dentro del paquete) de la máquina origen a la de destino, independizándolo del tipo de red física que esté utilizando. La PDU de la capa 4 se llama Segmento o Datagrama, dependiendo de si corresponde a TCP o UDP. Sus protocolos son TCP y UDP; el primero orientado a conexión y el otro sin conexión. Trabajan, por lo tanto, con puertos lógicos y junto con la capa red dan forma a los conocidos como Sockets IP:Puerto (191.16.200.54:80).

**FUNCIONES DE LA CAPA DE TRANSPORTE**

La capa de Transporte permite la segmentación de datos y brinda el control necesario para reensamblar las partes dentro de los distintos streams de comunicación. Las responsabilidades principales que debe cumplir son:

* seguimiento de la comunicación individual entre aplicaciones en los hosts origen y destino,
* segmentación de datos y gestión de cada porción,
* reensamble de segmentos en flujos de datos de aplicación,
* identificación de las diferentes aplicaciones.

**Seguimiento de Conversaciones individuales**

Cualquier host puede tener múltiples aplicaciones que se están comunicando a través de la red. Cada una de estas aplicaciones se comunicará con una o más aplicaciones en hosts remotos. Es responsabilidad de la capa de Transporte mantener los diversos streams de comunicación entre estas aplicaciones.

**Segmentación de datos**

Debido a que cada aplicación genera un stream de datos para enviar a una aplicación remota, estos datos deben prepararse para ser enviados por los medios en partes manejables. Los protocolos de la capa de Transporte describen los servicios que segmentan estos datos de la capa de Aplicación. Esto incluye la encapsulación necesaria en cada sección de datos. Cada sección de datos de aplicación requiere que se agreguen encabezados en la capa de Transporte para indicar la comunicación a la cual está asociada.

Reensamble de segmentos

**Reensamble de segmentos**

En el host de recepción, cada sección de datos puede ser direccionada a la aplicación adecuada. Además, estas secciones de datos individuales también deben reconstruirse para generar un stream completo de datos que sea útil para la capa de Aplicación. Los protocolos de la capa de Transporte describen cómo se utiliza la información de encabezado de dicha capa para reensamblar las secciones de datos en streams y enviarlas a la capa de Aplicación.

**Identificación de las aplicaciones**

Para poder transferir los streams de datos a las aplicaciones adecuadas, la capa de Transporte debe identificar la aplicación de destino. Para lograr esto, la capa de Transporte asigna un identificador a la aplicación. Los protocolos TCP/IP denominan a este identificador número de puerto. A todos los procesos de software que requieran acceder a la red se les asigna un número de puerto exclusivo en ese host. Este número de puerto se utiliza en el encabezado de la capa de Transporte para indicar con qué aplicación está asociada esa sección de datos.

**Los requerimientos de datos varían**

Debido a que las distintas aplicaciones poseen distintos requerimientos, existen varios protocolos de la capa de Transporte. Para algunas aplicaciones, los segmentos deben llegar en una secuencia específica de manera que puedan ser procesados en forma exitosa. En algunos casos, todos los datos deben recibirse para ser utilizados por cualquiera de las mismas. En otros casos, una aplicación puede tolerar cierta pérdida de datos durante la transmisión a través de la red.

**Separación de comunicaciones múltiples**

Considere una computadora conectada a una red que recibe y envía e-mails y mensajes instantáneos, explora sitios Web y realiza una llamada telefónica de VoIP de manera simultánea. Cada una de estas aplicaciones envía y recibe datos en la red al mismo tiempo. Sin embargo, los datos de la llamada telefónica no se direccionan al explorador Web y el texto de un mensaje instantáneo no aparece en el e-mail.

**PROTOCOLO TCP**

**Protocolo de control de transmisión (TCP)**

TCP es un protocolo orientado a la conexión, descrito en la RFC 793. TCP incurre en el uso adicional de recursos para agregar funciones. Las funciones adicionales especificadas por TCP están en el mismo orden de entrega, son de entrega confiable y de control de flujo. Cada segmento de TCP posee 20 bytes de carga en el encabezado, que encapsulan los datos de la capa de Aplicación, mientras que cada segmento UDP sólo posee 8 bytes de carga. Ver la figura para obtener una comparación.

Las aplicaciones que utilizan TCP son:

* exploradores Web
* e-mail
* transferencia de archivos

**PROTOCOLO UDP**

**Protocolo de datagramas de usuario (UDP)**

UDP es un protocolo simple, sin conexión, descrito en la RFC 768. Cuenta con la ventaja de proveer la entrega de datos sin utilizar muchos recursos. Las porciones de comunicación en UDP se llaman datagramas. Este protocolo de la capa de Transporte envía estos datagramas como "mejor intento".

Entre las aplicaciones que utilizan UDP se incluyen:

* sistema de nombres de dominios (DNS)
* streaming de vídeo
* Voz sobre IP (VoIP).

**Capa de Red:**

La Capa de red o Capa 3 de OSI provee servicios para intercambiar secciones de datos individuales a través de la red entre dispositivos finales identificados. Para realizar este transporte de extremo a extremo la Capa 3 utiliza cuatro procesos básicos:

* direccionamiento
* encapsulamiento
* enrutamiento
* desencapsulamiento

**Direccionamiento**

Primero, la Capa de red debe proveer un mecanismo para direccionar estos dispositivos finales. Si las secciones individuales de datos deben dirigirse a un dispositivo final, este dispositivo debe tener una dirección única. En una red IPv4, cuando se agrega esta dirección a un dispositivo, al dispositivo se lo denomina host.

**Encapsulación**

Segundo, la capa de Red debe proveer encapsulación. Los dispositivos no deben ser identificados sólo con una dirección; las secciones individuales, las PDU de la capa de Red, deben, además, contener estas direcciones. Durante el proceso de encapsulación, la Capa 3 recibe la PDU de la Capa 4 y agrega un encabezado o etiqueta de Capa 3 para crear la PDU de la Capa 3. Cuando nos referimos a la capa de Red, denominamos paquete a esta PDU. Cuando se crea un paquete, el encabezado debe contener, entre otra información, la dirección del host hacia el cual se lo está enviando. A esta dirección se la conoce como dirección de destino. El encabezado de la Capa 3 también contiene la dirección del host de origen. A esta dirección se la llama dirección de origen. Después de que la Capa de red completa el proceso de encapsulación, el paquete es enviado a la capa de enlace de datos que ha de prepararse para el transporte a través de los medios.

**Enrutamiento**

Luego, la capa de red debe proveer los servicios para dirigir estos paquetes a su host destino. Los host de origen y destino no siempre están conectados a la misma red. En realidad, el paquete podría recorrer muchas redes diferentes. A lo largo de la ruta, cada paquete debe ser guiado a través de la red para que llegue a su destino final. Los dispositivos intermediarios que conectan las redes son los routers. La función del router es seleccionar las rutas y dirigir paquetes hacia su destino. A este proceso se lo conoce como enrutamiento.

**Desencapsulamiento**

Finalmente, el paquete llega al host destino y es procesado en la Capa 3. El host examina la dirección de destino para verificar que el paquete fue direccionado a ese dispositivo. Si la dirección es correcta, el paquete es desencapsulado por la capa de Red y la PDU de la Capa 4 contenida en el paquete pasa hasta el servicio adecuado en la capa de Transporte.

**DIRECCIONAMIENTO IP**

El direccionamiento es una función clave de los protocolos de capa de Red que permite la transmisión de datos entre hosts de la misma red o en redes diferentes. El Protocolo de Internet versión 4 (IPv4) ofrece direccionamiento jerárquico para paquetes que transportan datos. Diseñar, implementar y administrar un plan de direccionamiento IPv4 efectivo asegura que las redes puedan operar de manera eficaz y eficiente. Cada dispositivo de una red debe ser definido en forma exclusiva. En la capa de red es necesario identificar los paquetes de la transmisión con las direcciones de origen y de destino de los dos sistemas finales. Con IPv4, esto significa que cada paquete posee una dirección de origen de 32 bits y una dirección de destino de 32 bits en el encabezado de Capa 3.

En cada dirección IPv4, alguna porción de los bits de orden superior representa la dirección de red. En la Capa 3, se define una red como un grupo de hosts con patrones de bits idénticos en la porción de dirección de red de sus direcciones.

A pesar de que los 32 bits definen la dirección host IPv4, existe una cantidad variable de bits que conforman la porción de host de la dirección. El número de bits usado en esta porción del host determina el número de hosts que podemos tener dentro de la red.

**Capa de Aplicación:**

**Nivel de aplicación**

Ofrece a las aplicaciones la posibilidad de acceder a los servicios de las demás capas y define los protocolos que utilizan las aplicaciones para intercambiar datos, como correo electrónico (Post Office Protocol y SMTP), gestores de bases de datos y servidor de ficheros (FTP). Hay tantos protocolos como aplicaciones distintas y puesto que continuamente se desarrollan nuevas aplicaciones el número de protocolos crece sin parar.  
Cabe aclarar que el usuario normalmente no interactúa directamente con el nivel de aplicación. Suele interactuar con programas que a su vez interactúan con el nivel de aplicación pero ocultando la complejidad subyacente.

**Nivel de presentación**

El objetivo es encargarse de la representación de la información, de manera que aunque distintos equipos puedan tener diferentes representaciones internas de caracteres los datos lleguen de manera reconocible. Esta capa es la primera en trabajar más el contenido de la comunicación que el cómo se establece la misma. En ella se tratan aspectos tales como la semántica y la sintaxis de los datos transmitidos, ya que distintas computadoras pueden tener diferentes formas de manejarlas. Esta capa también permite cifrar los datos y comprimirlos. Por lo tanto, podría decirse que esta capa actúa como un traductor.

**Nivel de sesión**

Esta capa es la que se encarga de mantener y controlar el enlace establecido entre dos computadores que están transmitiendo datos de cualquier índole. Por lo tanto, el servicio provisto por esta capa es la capacidad de asegurar que, dada una sesión establecida entre dos máquinas, la misma se pueda efectuar para las operaciones definidas de principio a fin, reanudándolas en caso de interrupción. En muchos casos, los servicios de la capa de sesión son parcial o totalmente prescindibles.

**2.1.2. PROTOCOLOS Y FUNCIONALIDAD**

**PROTOCOLOS DE APLICACIÓN**

**-NFS:** es un protocolo de nivel de aplicación, según el Modelo OSI. Es utilizado para sistemas de archivos distribuido en un entorno de red de computadoras de área local.

**-Telnet:** Es el protocolo que define el conjunto de reglas y criterios necesarios para establecer sesiones de terminal virtual sobre la red. Telnet define los mecanismos que permiten conocer las características del computador destino. Así mismo, permite que los dos computadores (cliente y servidor) negocien el entorno y las especificaciones de la sesión de emulación de terminal.

**-FTP:** File Transfer Protocol

**-HTTP:** Hypertext Transfer Protocol  
**-POP3:** Post Office Protocol version 3

**-SMTP:** Simple Mail Transfer Protocol  
**-SNMP:** Simple Network Management Protocol

## ****Ventajas del modelo TCP/IP****

## TCP/IP ofrece ventajas significativas respecto a otros protocolos de red. Una de esas ventajas es que es capaz de trabajar sobre una extensa gama de hardware y soporta muchos sistemas operativos (es multiplataforma). Internet está repleto de pequeñas redes con sus propios protocolos por lo que el uso de TCP/IP se ha estandarizado y es posible utilizarlo como protocolo de comunicación entre redes privadas intranet y extranet, facilitando una red más homogénea.

* TCP/IP es adecuado tanto para grandes y medianas redes como para redes empresariales o domésticas.
* TCP/IP está diseñado para enrutar y además presenta gran compatibilidad con las herramientas estándar para analizar y monitorizar el funcionamiento de una red.
* Es el protocolo estándar que se utiliza a nivel mundial para conectarse a internet y a los servidores web.

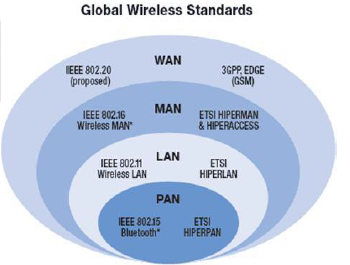
## ****Desventajas del modelo TCP/IP****

* No distingue bien entre interfaces, protocolos y servicios lo cual afecta al desarrollo de nuevas tecnologías basadas en TCP/IP-
* En redes con bajo volumen de tráfico puede llegar a ser más lento (en redes con mayor volumen de tráfico, que necesiten gran cantidad de enrutamiento, puede ser mucho más rápido).
* Cuando se utiliza en servidores de ficheros o servidores de impresión no ofrecen un gran rendimiento.

**Topologías y arquitecturas**

Modelos arquitecturales de redes de acuerdo al tamaño

Las redes pueden ser clasificadas de diferentes maneras de acuerdo al contexto sobre el cual se esté estudiando. La siguiente es la clasificación de redes de acuerdo a su tamaño:



**Figura 4 -[9]Clasificación de redes según tamaño**

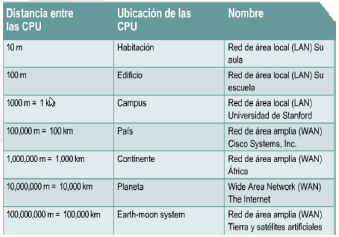
PAN

*Personal Area Network* (PAN), consideradas como redes de computadores pequeñas organizadas alrededor de un único dispositivo. Este tipo de redes es utilizado para compartir información entre dispositivos móviles, portátiles, etc. El rango de cobertura es de aproximadamente 10 metros con tecnologías con Bluetooth e Ingrarojos.

LAN

*Local Area Network* (LAN), son las redes utilizadas en [organizaciones](https://www.monografias.com/trabajos6/napro/napro.shtml), [instituciones](https://www.monografias.com/trabajos13/trainsti/trainsti.shtml), universidades y diferentes tipos de organizaciones para establecer comunicación entre diferentes equipos ubicados en un área geográfica determinada y compartir recursos como , internet, [Bases de Datos](https://www.monografias.com/trabajos11/basda/basda.shtml), espacio de almacenamiento, etc.

La estructura de [red LAN](https://www.monografias.com/trabajos11/reco/reco.shtml#ti) está compuesta por uno o más switches interconectados entre sí. Los esquemas de conexión pueden ser en pila, cascada, etc., e interconectan los diferentes dispositivos. En los casos en que se requiere conexión a otras redes, es necesario el uso de un [10]*router* provisto por el [11]ISP (Internet Service Provider) bajo el modelo de conexión de última milla.



**Figura 5 - Tamaños de las Redes**

MAN/WAN

*Metropolitan Area Networks* (MAN) son redes amplias que cubren áreas geográficas del tamaño de una ciudad, como por ejemplo, sede de las oficinas de una misma [organización](https://www.monografias.com/trabajos6/napro/napro.shtml) en distintas zonas de la ciudad.

Una MAN puede usarse para interconectar diferentes LAN o para conectar con una *Wide Area Network* (WAN) como lo es Internet. Las conexiones entre LAN y MAN son efectuadas a través de tecnologías de alta [velocidad](https://www.monografias.com/trabajos13/cinemat/cinemat2.shtml#TEORICO) como son [ATM](https://www.monografias.com/trabajos/atm/atm.shtml)/SDH, ATM/SONET, FrameRelay/xDSL, ATM/E1, etc.alcanzando velocidades desde 10Gb hasta 1.3Tb con tecnologías como DWDM.

Las WAN proporcionan conexiones entre ciudades o países, por lo que pueden interconectar redes MAN entre si. Requieren tecnologías de alta velocidad como las MAN y de equipos como [12]switches capa 2 y [13]*routers* capa 3 para logra interconectividad.

Las [topologías](https://www.monografias.com/trabajos15/topologias-neural/topologias-neural.shtml) describen la forma de interconexión tanto de los equipos [activos](https://www.monografias.com/trabajos11/contabm/contabm.shtml) como de los dispositivos que interconectan. A continuación se detallan los diferentes tipos de topologías físicas presentes en las redes.

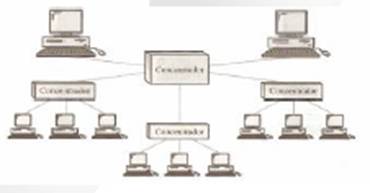
Topología de Estrella

Esta topología utiliza un equipo activo denominado concentrador o Switch al cual los equipos son conectados con un enlace punto a punto. El concentrador es el que modera la comunicación y sobre él cruza toda la comunicación realizada por los equipos que interconecta. Esta topología no permite el tráfico directo entre dispositivos.



Topología en Árbol

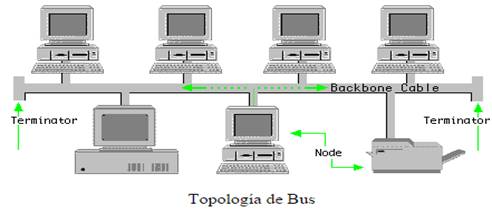
Es una modificación de la topología de estrella en la cual switches de borde o secundarios confluyen en switches de core o primarios para [poder](https://www.monografias.com/trabajos35/el-poder/el-poder.shtml) conectar a equipos que se encuentran a distancias mayores. Dispone de repetidores que generan los mensajes recibidos y los retransmite a distancias mayores.



**Figura 9 - [17]Topología en Árbol**

Topología de Bus

En esta [topología](https://www.monografias.com/trabajos15/topologias-neural/topologias-neural.shtml) la conexión de los equipos es multipunto, es decir, un solo cable actúa como red y es el punto de interconexión entre los dispositivos de la red los cuales quedan dispuestos de manera serial. La conexión de los dispositivos a la red, es realizada mediante el uso de conectores hembra y macho dependiendo de la [tecnología](https://www.monografias.com/Tecnologia/index.shtml) que se emplee. La ventaja de este tipo de interconexión es la sencillez, aunque tiene un límite de longitud.

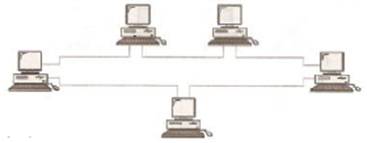


**Figura 6 -[14]Topología de Bus**

Topología de Anillo

En esta topología cada equipo tiene una conexión punto a punto con máximo dos equipos. La comunicación va en una solo sentido, transmitiendo los datos de un equipo a otro hasta que llegue a su destino.

Esta topología es fácil de implementar y configurar ya que cada equipo tiene solo dos conexiones físicas, sin embargo, el número de nodos (equipos) es restringido debido a la latencia que genera el [transporte](https://www.monografias.com/trabajos/transporte/transporte.shtml) de datos.



**Figura 7 - [15]Topología de Anillo**

Topología en Malla

Una red en malla interconecta todos los equipo de red entre si mediante canales dedicados estableciendo un tipo de conexión Punto a Punto o P2P. Esta topología y tipo de conexión generada es benéfico en [seguridad](https://www.monografias.com/trabajos/seguinfo/seguinfo.shtml), latencia y altas velocidades de transferencia.

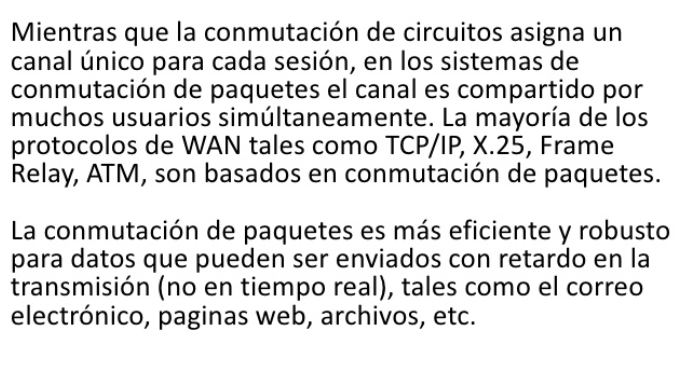
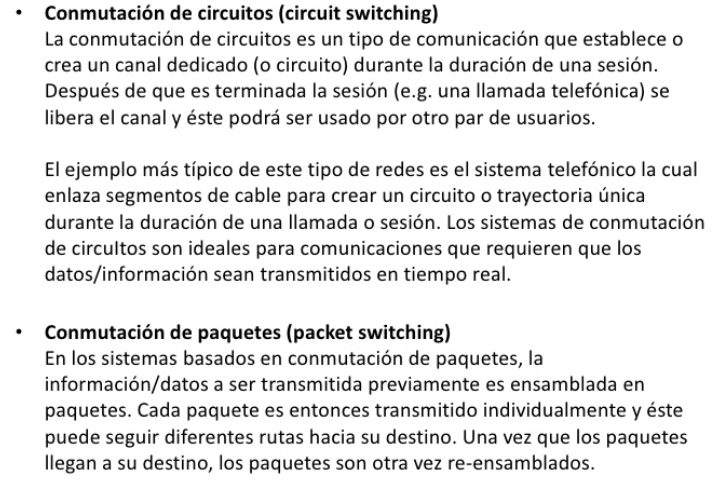


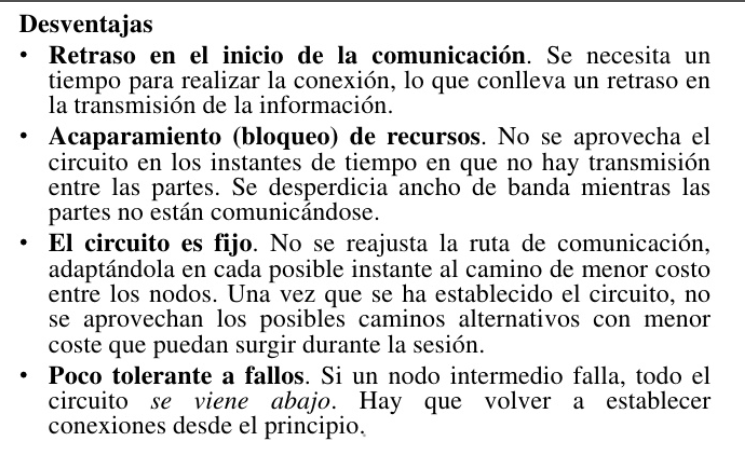
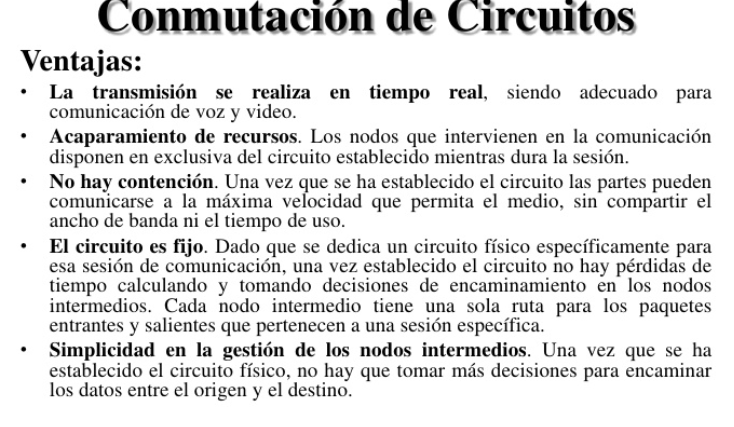
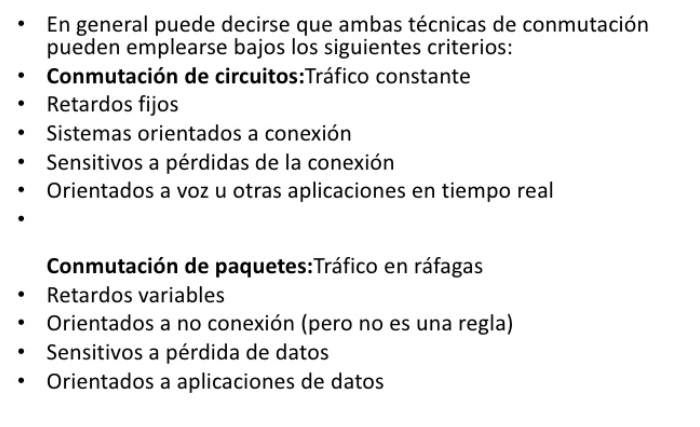
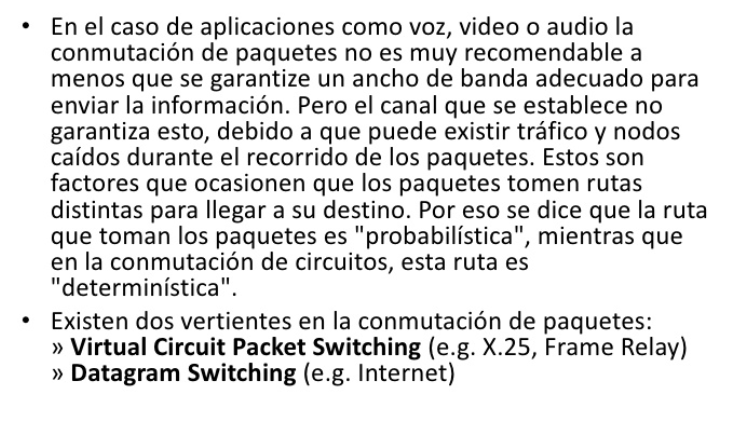
**Figura 10 - [18]Topología en Malla**

Las redes son utilizadas para despachar diferentes tipos de servicios a través de diferentes topologías y arquitecturas, para ello, hace uso de una gran variedad de equipos y tecnologías. Por lo anterior es de vital importancia obtener [el conocimiento](https://www.monografias.com/trabajos/epistemologia2/epistemologia2.shtml) y entendimiento de características y [comportamiento](https://www.monografias.com/trabajos16/comportamiento-humano/comportamiento-humano.shtml)las redes para hacer un uso adecuado de ellas para utilizarlas como [herramientas](https://www.monografias.com/trabajos11/contrest/contrest.shtml) que permitan el [diseño](https://www.monografias.com/trabajos13/diseprod/diseprod.shtml) y [arquitectura](https://www.monografias.com/trabajos6/arma/arma.shtml) de una estructura de comunicaciones adecuada a las necesidades de una corporación.

La humanidad se ha enfrentado a diferentes paradigmas sobre las cuales se ha tenido una reacción positiva y aportando grandes avances en la evolución de las redes. Partiendo del [paradigma](https://www.monografias.com/trabajos16/paradigmas/paradigmas.shtml#queson) de MainFrame, esquema centralizado con super-cómpto con la capacidad de realizar [operaciones](https://www.monografias.com/trabajos6/diop/diop.shtml) de red aunque a [costos](https://www.monografias.com/trabajos4/costos/costos.shtml) altos, pasando por PCs independientes representados en islas independientes de cómputo y almacenamiento y, llegando finalmente a las tecnologías de redes que integran esas islas de cómputo independiente del [sistema operativo](https://www.monografias.com/Computacion/Sistemas_Operativos/) o [hardware](https://www.monografias.com/Computacion/Hardware/) sobre el cual corren.

Tipos de redes





Metcalfe pensaba que los segmentos Ethernet debían vincularse entre sí, por algo más inteligente que un simple repetidor de señales (capa L1); y ese algo tiene que ver con el concepto de “Puentes Transparentes”. ¿Cuál es la diferencia entre interconectar segmentos LAN a través de repetidores o hacerlo a través de puentes transparentes? Todo esto lleva a definir un concepto muy importante que es “DOMINIO DE COLISIÓN”.

Sí se conectan 5 segmentos de red a través de 4 repetidores, se obtiene un enlace de 2500 metros de extensión (recordar que la primer tecnología Ethernet fue 10Base5). Bajo la premisa anterior, cuando una computadora transmita una trama, esta se difundirá por “todo el medio” (cable coaxil). Cuando una o más computadoras intenten transmitir cuasi simultáneamente empleando CSMA/CD, existirá UN único dominio de colisión que es TODO el enlace (los 2500 metros). En otras palabras, la colisión se propagará por todo el enlace. Decimos en este caso que sólo existe un dominio de colisión.

Ahora que tal si en vez repetidores, se conectan los segmentos mediante 4 dispositivos que llamaremos PUENTES TRANSPARENTES. ¿Qué es un puente transparente? En primer lugar diremos que es un dispositivo que funciona en L2 (capa 2). Puente, porque interconecta 2 segmentos Ethernet. El concepto de transparencia tiene sentido porque es transparente a las computadoras que se comunican en el enlace Ethernet.

Funciona de la siguiente manera:

El dispositivo puente transparente posee dos interfaces de red (Ethernet) para conectar dos segmentos. También puede analizar las tramas con el sólo propósito de guardar las direcciones MAC fuentes) de las tramas que recibe por sus (dos) interfaces (en modo promiscuo). Ello le permite construir una tabla en “su memoria” relacionando direcciones MAC fuentes, con el número de interfaz por donde recibió la trama. Sabemos que una dirección Ethernet se conforma con 6 bytes, y cada una de las computadoras en la LAN tiene que tener una dirección ÚNICA que a partir de ahora llamaremos DIRECCIÓN UNICAST.

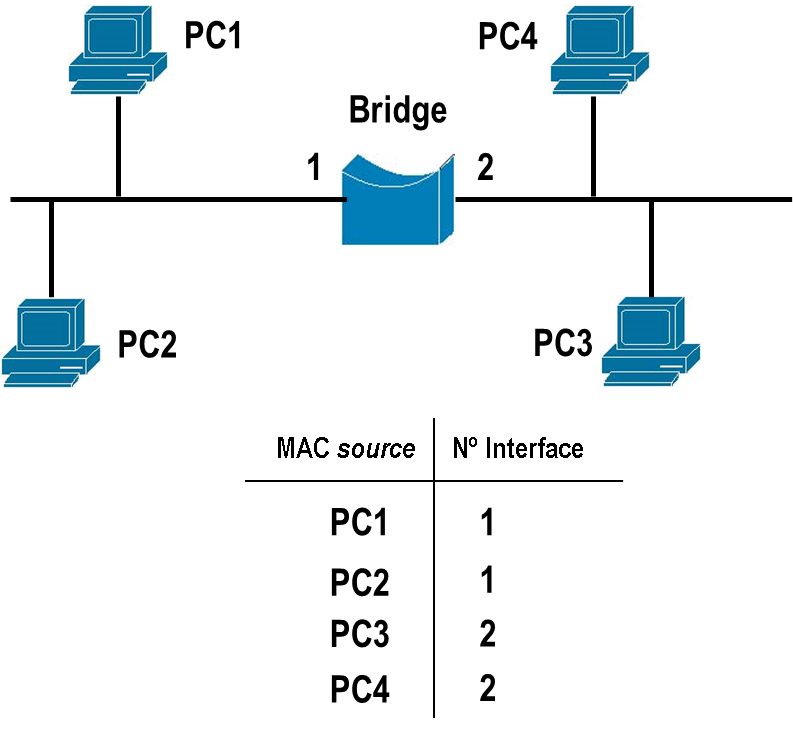


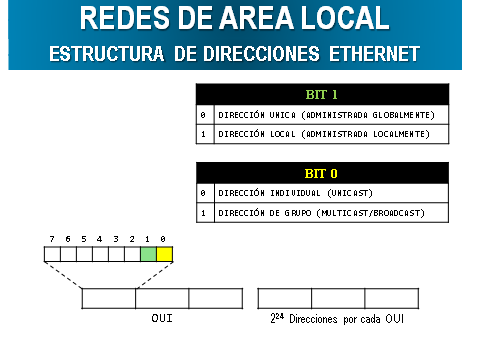
Figura 1: puente transparente conectando dos segmentos Ethernet

Suponer que se reinicia el bridge. Luego del reinicio, su tabla estará en cero. Bajo esa condición, por ejemplo, consideremos que PC1 le envía una trama a PC2, el bridge entonces aprenderá que PC1 está conectada del lado se su interfaz 1. Entonces escribe en su memoria MAC PC1, Interfaz 1, pero no sabe nada de PC2, entonces reenviará la trama que recibió en modo promiscuo por la interface 2 (no tiene sentido que el bridge la reenvíe por la interfaz l, pues por esa interfaz la recibió, al igual que todas las PC´s conectadas al segmento del lado izquierdo de la figura). Seguramente, y luego de haber recibido la trama de PC1, PC2 le contestará y generará una trama hacia PC1; entonces ahora el bridge aprende que PC2 está en el segmento del lado de su interfaz 1 (escribe en su memoria PC2; Interfaz 1). El proceso continúa hasta que finalmente, el bridge o puente llena la tabla.

Lo importante del elemento “bridge”, es que separa dominios de colisión. Los aísla. En la figura claramente se observan dos dominios de colisión. PC1 puede colisionar con PC2 y con un intento del propio bridge en enviar tramas de computadoras asociadas a su interfaz 2, por su interfaz 1; o viceversa.

Estructura de Direcciones Ethernet (Direcciones MAC)

Sabemos que una dirección MAC consta de 48 bits o 6 bytes



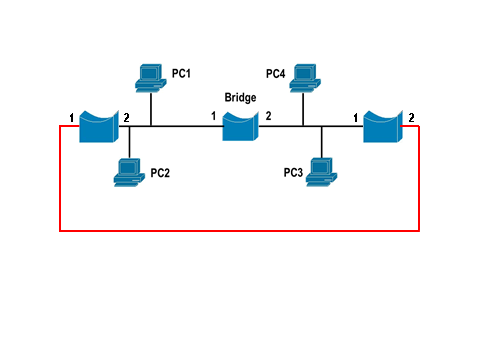
Notar que en la Figura 5 se ha representado de exprofeso una dirección MAC dividida en dos partes de 3 bytes cada una. ¿Cuál es la razón? La razón es que el estándar de la IEEE 802.2, reservó 22 bits de los primeros 3 bytes, para asignar un código para los fabricantes de interfaces Ethernet. Qué es un OUI? Es un identificador único de la organización que fabrica interfaces de red.

Observando el primer byte en la Figura 5, hay dos bits que son muy importantes. El bit 0 y el bit 1. Cuando el valor del bit “0” es igual a “0 lógico”, la dirección es UNICAST (individual). Recordar que cada computadora en la red Ethernet necesita obligatoriamente de una dirección MAC unicast.

Sí por el contrario, el valor del bit “0” fuese “1 lógico” se trata de una dirección MULTICAST o BRADCAST. Explicaremos que significan estos términos.

Dirección Multicast: se trata de una dirección que identifica a “un grupo de dispositivos” que se están comunicando mediante un protocolo para efectuar una tarea definida. Ojo! Está dirección no remplaza a la dirección Unicast, sino que coexisten.

Para evitar estos lazos que en LANs modernas son comunes se emplea un protocolo de capa 2 (L2) llamado Spanning Tree Protocol (STP) cuyo objetivo es: a partir de una topología mallada (real) construye un árbol lógico, deshabilitando interfaces redundantes. Ante una falla, el protocolo STP puede reconstruir otro árbol lógico para que no se pierda conectividad en la infraestructura LAN.

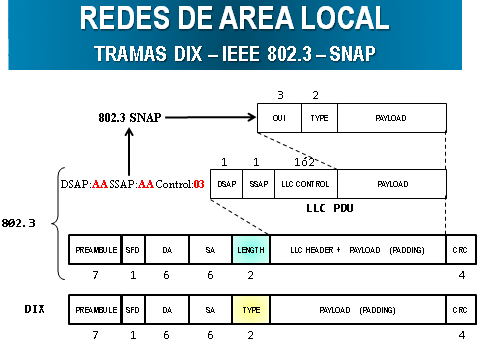


¿Y una dirección broadcast que es? En Ethernet se trata de la dirección MAC destino, FF:FF:FF:FF:FF:FF. Cuando se genera una trama cuya dirección es la dirección destino es broadcast, TODAS las computadoras y dispositivos en la red deben leer la trama obligatoriamente.

Respecto al bit 1 de una dirección MAC destino, si su valor es “0 lógico” significa que la dirección es globalmente asignada (por la IEEE) y única en el mundo. Sí el bit 1 es un “1 lógico” significa que es localmente administrada. Ello significa que por más que la placa Ethernet tenga la dirección unicast “fijada” en hardware (imposible de borrar), un administrador puede decidir reemplazarla por otra localmente asignada (por él administrador) para propósitos que sólo el puede justificar.

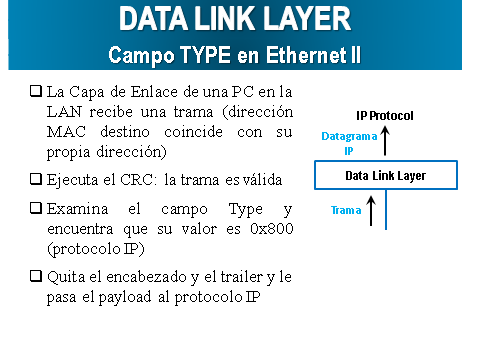
DEBE QUEDAR CLARO QUE LAS DIRECCIONES MULTICAST Y BROADCAST TIENEN SIGNIFICADO EN EL CAMPO DE LA TRAMA ETHERNET “DESTINATION ADDRESS”. LA DIRECCIÓN FUNTE DEBE SER SIEMPRE UNA DIRECCIÓN UNICAST.

Tipos de Tramas Ethernet (Direcciones MAC)



Se observan dos campos de bytes que no los inserta la Capa de Enlace; se trata de los campos PREAMBULE Y SFD (Start Frame Delimiter). Entre los dos suman un total de 8 bytes y los inserta la capa física Ethernet para proveer sincronización en la recepción de tramas. ¿Cómo funciona? El PREAMBULE consta de 7 bytes que alternan entre 1´s y 0´s en forma consecutiva: 101010…1010 (56 bits) y SDF es 10101011 (campo que indica el inminente comienzo de la trama Ethernet al/los receptores. Con una tasa de transmisión de 10Mbps, el tiempo de un bit es 0,1µs, ello significa que un receptor tiene 64 x 0,1µs o 6,4 µs para sincronizarse con la trama que se envía. Los campos PREAMBULE y SFD los inserta la capa física Ethernet para fines de sincronización.

Como se observa en la figura 1, el campo Type en Ethernet DIX (2 bytes), sirve para indicarle al receptor a que protocolo debe entregar el payload de esta trama.



Luego de la estandarización, IEEE cambió el significado al campo Type, y lo transformó en Lenght (cantidad de bytes en el payload). El campo Length sirve para indicar al receptor el número de bytes en el payload de la trama

Lenght especifica la cantidad de bytes en el campo de datos de la trama Ethernet (payload).

Como Ethernet DIX debía coexistir con la trama IEEE 802.3, se llegó a una solución:

1. Cuando el valor del campo Type/Lenght es menor a 0x600 se trata de una trama IEEE 802.3, pues la máxima longitud en bytes del payload no puede superar los 1500 bytes (campo Lenght).
2. Valores superiores a 0x600: trama Ethernet DIX o Ethernet II (campo Type)

IEEE 802.3 debía incluir el campo Type, pues: ¿cómo entregar el payload de la trama al protocolo correcto en el extremo receptor? intentaron resolverlo con los campos DSAP (Punto de Acceso al Servicio Destino) y SSAP (Punto de Acceso al Servicio Origen).

La lógica indica, que el protocolo que escribió datos en el payload de la trama, será el mismo protocolo al que se le debe entregar el payload en la parte receptoraPosteriormente surgió un nuevo inconveniente, 1 byte permite especificar 256 protocolos, y ese aspecto resulto ser insuficiente.

En la Figura 1 también se observa el encabezado de la subcapa denominada Logical Link Control – o por su acrónimo LLC, en español Control de Enlace Lógico. Está subcapa fue estandarizada como IEEE 802.2.

**SUBCAPA LLC**

La Figura 1 muestra las funciones obligatorias y no obligatorias en Ethernet (Capa de Enlace). Las funciones no obligatorias se añadieron luego del proceso de estandarización de Ethernet cuando la IEEE agregó la subcapa 802.2 o LLC.

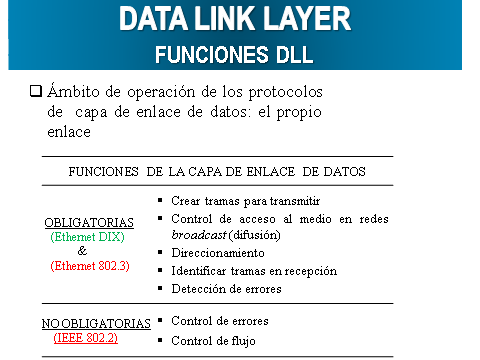


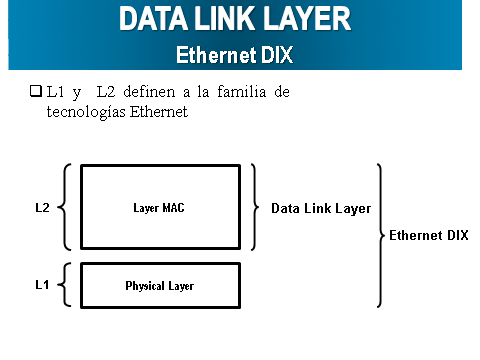
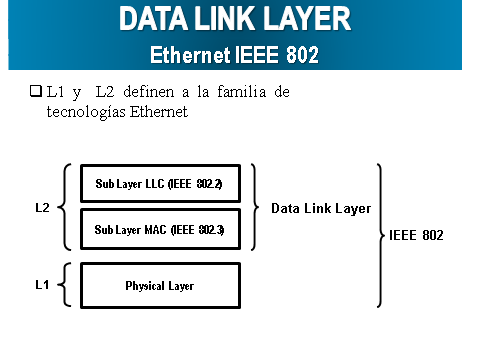
Tabla 1: Funciones de la Capa de Enlace

Ethernet DIX (Figura 2) sólo implementó la capa MAC (funciones obligatorias mostradas en la Tabla 1).

Mientras que la subcapa IEEE 802.2 (Figura 3) ofrece 3 servicios a la capa superior.

* Servicio orientado a la conexión (ejecuta funciones no obligatorias)
* Servicio no orientado a la conexión (no ejecuta funciones no obligatorias)
* Servicio no orientado a la conexión con reconocimiento (no utilizado)

Cuando un protocolo utilice el servicio de IEEE 802.2, orientado a la conexión se llevará a cabo en capa 2 (L2), control de errores y control de flujo (a nivel del enlace).

Figura 2: Ethernet original DIX (trama Ethernet II)Figura 3: Ethernet IEEE

Lo que debe quedar en claro es que coexisten los tres tipos de tramas en Ethernet. Cuando se trata de Ethernet cableadas (UTP o Fibra Óptica) se utilizan tramas según el formato DIX, o tramas Ethernet II. Cuando se trata de tecnologías Ethernet inlámbricas se requiere de tramas IEEE 802.3 (ver Figura 3). Es fácil justificarlo, los medios guiados tienen una alta inmunidad a la interferencia electromagnética, mientras que no ocurre lo mismo con Ethernet inalámbricas.

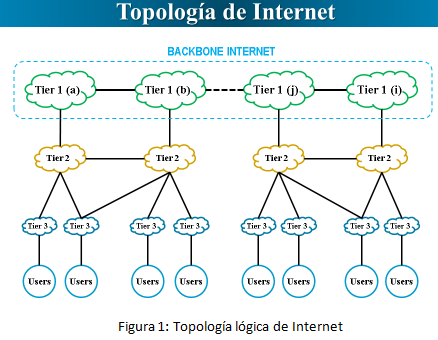
**Capa de Red en Internet**

La capa de red en el modelo TCP/IP aloja a uno de los protocolos que definió el nombre de la arquitectura de red: el protocolo IP. En la actualidad coexisten dos versiones del protocolo: IPv4 e IPv6. Comenzaremos con el estudio de la primera versión: IPv4.

Infraestructura de Internet

Es importante antes de enfocarnos en IPv4 conocer como es la topología “lógica” de Internet. El núcleo o backbone (columna vertebral) de Internet la conforman grandes corporaciones que despliegan su infraestructura de red sobre amplios territorios. Se trata de grandes WAN diseminadas en el planeta a las que se denomina TIERS 1 (nivel 1). Los TIERS 1 son jugadores de primer nivel en el negocio de Internet. Los TIERS 1 se interconectan entre sí, y no intercambian U$S por el servicio de tránsito; a esta relación de la llama entre “peering”. La razón de esta estrategia es simple, para una Internet global se debe poder alcanzar toda la geografía del mundo.

Los TIERS 1 forman un grupo selecto; ellos no se caracterizan por aceptar nuevos socios al gran negocio. Es necesario aclarar que, en países con extensa geografía, elevada densidad poblacional, y con leyes antimonopólicas, tienen que coexistir más de un TIER 1.



Los que siguen en la jerarquía, son los TIERS 2, que necesitan conectarse por lo menos a un TIER 1 y para ello tienen que pagar U$S de acuerdo al volumen de tráfico que inyectan al backbone. Para ser considerado Tier 2 hay que tener una envergadura importante Los Tiers de nivel 2 con características similares en envergadura, y que coexisten en lo que a territorio se refiere; por lo general realizan acuerdos para conectarse entre sí (sin que ello implique costos), con el único objetivo de evitar accesos (evitables) vía Tiers 1 y así disminuir costos.

Por último, en lo que respecta a jerarquías de Tiers se encuentran los de nivel 3. Son los que conectan abonados a Internet (ISP – Proveedores de Servicios de Internet). Los Tiers 3 por lo general se conectan a más de un Tier 2, y ello responde a dos propósitos: tener redundancia cuando un TIER 2 presenta problemas, y distribuir la capacidad en lo que se refiere al volumen de tráfico que contratan.

En el nivel inferior se encuentran los usuarios que contratan a uno o más ISP`s (Tier 3). Este nivel es el único generador y consumidor de servicios de Internet.

**Protocolo IPv4**

El protocolo IPv4 definido por el RFC 791 en el año 1981 definió un esquema de direccionamiento global (a nivel de Internet). Los autores del estándar consideraron que 32 bits era una cantidad suficiente para definir las direcciones IPv4 en forma global.

Tal como se puede leer, el texto de la Figura 3 hace referencia a “flexibilidad” en la asignación de direcciones IPv4 y se refiere a “muchas” redes pequeñas (con pocas computadoras), redes intermedias (con moderada cantidad de computadoras) y “pocas” redes grandes (con gran cantidad de computadoras).

La estructura de direccionamiento en Internet tiene que ser jerárquica. Sí observamos la Figura 1 los router´s (dispositivo que opera a nivel de capa 3) en el backbone tienen que tener una tabla de rutas a todas las redes IP del planeta.

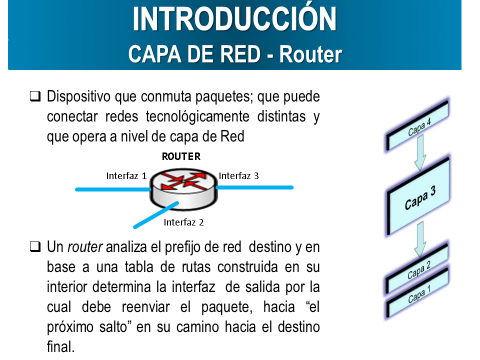


Figura 4: Router, dispositivo de conmutación en capa de red.

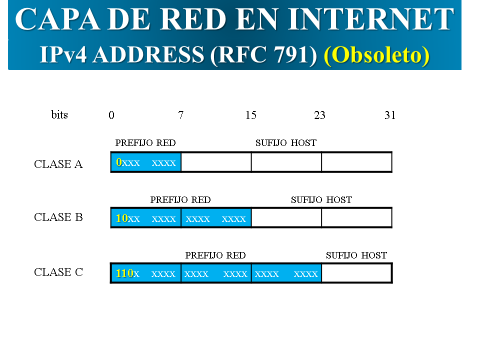
Un datagrama IPv4 debe tener en su encabezado, entre otros campos, un campo para la dirección IPv4 de la computadora que generó el paquete IP, y la dirección IPv4 destino.

los router´s conmutaran paquetes en relación a red destino a la cual va dirigido un paquete IPv4 y no en direcciones IPv4 individuales. De este modo, los routers en el backbone de Internet deberán consultar tablas que contengan “direcciones de redes” y “no direcciones IPv4 individuales”.

se definieron 3 clases de redes IPv4, las denominaron: clase A. clase B y clase C.

A una red de Clase A, se la identifica por el primer bit en cero (0). Notar que con esta decisión las redes de Clase A ocupan la mitad del espacio de direccionamiento total de IPv4.

El total de redes IPv4 de clase A (ver prefijo de red) es de 127. En otras palabras, 127 redes de clase A y cada red puede contener hasta 2^24 (aproximadamente) computadoras: : la friolera cantidad de 16.777.216 direcciones IPv4 para cada red de esta Clase.



El RFC 791 definió redes de Clase B (ver Figura 5). Con los dos primeros bits de la dirección en 1-0, se identifica una red como de Clase B. El prefijo de red es se extendió a 2 bytes, por lo que pueden existir 2^14 = 16.384 redes de Clase B; cada una de ellas con capacidad para conectar (aproximadamente) 2^16 = 65.536 dispositivos en red (16 bits es la extensión del sufijo para hosts).

Por último, el estándar definió un número enorme de redes de Clase C (2^21 = 2.097.152) cada una de ellas con capacidad de conectar (aproximadamente) 2^8 = 256 host. Las cuentas se realizan considerando el “prefijo de red” y el “sufijo para host”

Capa de Red en Internet

Las redes de Clase “A” rápidamente fueron adjudicadas, con las redes de Clase “B” ocurrió lo mismo, aunque en un transcurso de tiempo mayor. Luego de ello, no quedó otra alternativa que comenzar a adjudicar redes de Clase “C”. Hay que recordar que el total de redes de Clase “C” era un número superior los 2 millones de redes. De este modo los *routers* en el núcleo de Internet *(backbone)* comenzaron a incrementar el tamaño de sus tablas de rutas en forma exponencial. Otro problema crucial fue la gran cantidad de direcciones IPv4 infrautilizadas en redes IPv4 de Clase “A” y de Clase “B”. En ese contexto se definió el concepto de *subnet* (subred).

**Definición de Subredes (SubNets)**

crear subredes en una red IPv4 permite sortear el impedimento en cuanto al número de hosts que puede conectar una tecnología LAN.

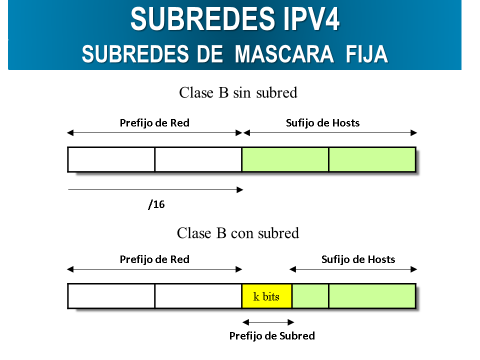
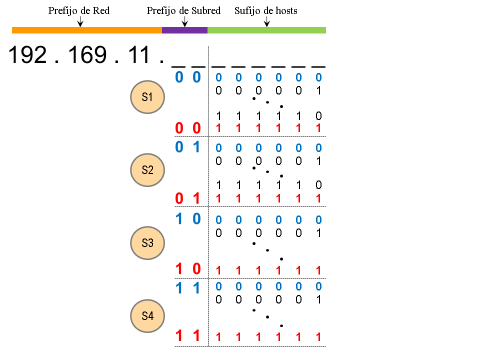
El concepto de subredes consiste en utilizar *k* bits del espacio “sufijo para hosts” para crear 2^k subredes. Este mecanismo es válido para cualquier tipo de clase de red (A, B o C).

Figura 1: Creación de Subredes

Crear 4 subredes a partir de la red 192.169.11.0/24:

Para crear 4 subredes en la red de Clase “C” el valor de *k* debe ser 2 bits ya que 2^2=4 subredes.

Figura 3: Procedimiento para crear las subredes

Es importante recalcar que el prefijo /24 debe quedar inalterable, y el prefijo de subred (k=2 bits) se forma a partir de los bits más significativos del sufijo para hosts (ver Figura 3). A cada una de las subredes se las llamó S1, S2, S3, y S4.

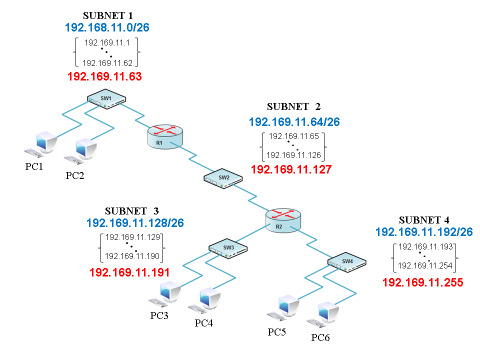
Notar que con k=2, la subred S1 la definimos con 00, la S2 con 01, la S3 con 10, y la S4 con 11. Cada subred tiene su dirección IPv4 que no se puede asignar a ningún host. También cada subred tiene una dirección de broadcast que no es asignable a ningún host. ¿Cómo se forma la dirección de subred y la dirección de broadcast de cada subred?

Consideremos la subred S3; notar que el prefijo de subred es 10. Sí se completa el sufijo para host con ceros, se obtendrá la dirección de la subred (10|00 0000); 192.169.11.128. La dirección de difusión de S3 será 10|11 1111 (192.168.11.191). Ver Figura 4.

Dentro de cada subred el sufijo para host ya no es de 8 bits de longitud, los k bits del prefijo de subred, restaron 2 bits al sufijo de host de manera que ahora cada subred tendrá un espacio de direcciones IPv4 de 64 direcciones (62 de ellas para ser asignadas a hosts).

Lo importante de todo esto, es que a la red 192.169.11.0/24 se la puede alcanzar desde el exterior (desde otra red IPv4) sin conocer la complexidad de las subredes internas. La división interna en subredes y el ruteo implícito a que conlleva sólo lo conoce el administrador de la red.

La Figura 5 ilustra la topología resultante de la división de la red 192.169.11.0/24 en cuatro (4) subredes.Noten que entre los routers R1 y R2 sólo se necesitan 2 direcciones IPv4, por ser un enlace punto a punto y le estamos asignando un bloque de 64 direcciones. No tiene sentido. Esto sucedió porque hemos utilizado un prefijo de red (*k* bits) común a todas las subredes; por lo que todas tendrán la misma capacidad en lo que a direcciones IPv4 se refiere. A este procedimiento se lo denomina creación de subredes con “máscara fija”.



Para que los routers R1 y R2 puedan “encaminar” datagramas IPv4 entre subredes, la máscara de subred (Genmask en la tabla de rutas) será /26; porqué ahora tiene que comprobar el prefijo de subred para tomar decisiones de encaminamiento.

Capa de Red en Internet

La subdivisión de una red IPv4 en subredes del mismo tamaño (máscara de subred fija), impide la correcta distribución de direcciones IPv4 y no se ajusta a las necesidades del administrador en una infraestructura de red.

**Subredes de Máscaras Variables (VLSM)**

Con VLSM, cada subred se puede dimensionar de acuerdo a los requerimientos de número de hosts o interfaces IPv4 que el administrador de la infraestructura ha planificado en la etapa de diseño.

Conceptualmente lo que se debe entender, es que el RFC791 (protocolo IPv4) fijó sólo tres máscaras: /8 para la Clase A, /16 para la Clase B, y /24 para la Clase C). En ese contexto, los router´s no admitían otros prefijos distintos a los mencionados. Con la introducción del concepto de subredes, se flexibilizó el concepto de máscaras y su extensión a nivel de subred. Uds. han comprobado que, para el caso de subredes de máscara fija, el ejercicio propuesto contempló máscaras /26 para el propósito de encaminamiento dentro de la subred.

Comenzaremos con VLSM, contemplando la misma topología de red utilizada cuando se trató la creación de subredes con máscara fija. Tal como se observa en la Figura 1, se han estipulado criterios de diseño que se listarán a continuación:

Red IPv4: 192.169.11.0/24

Subred Address IPv4 por Subred

S1 28

S2 2

S3 45

S4 52

Lo primero a verificar es que el total de direcciones IPv4 en las subredes no supere en cantidad a los que provee la red IPv4 192.169.11.0/24. El /24 indica la longitud del prefijo de red; ello significa que el sufijo para hosts es de 8 bits, entonces el número de direcciones disponibles es 2^8 – 2 = 254 direcciones IPv4 (hay que restar dos direcciones: la de la red y la de broadcast de la red).

A partir de los requerimientos de la Tabla 1, se calculará el número total de direcciones IPv4 en las subredes y al resultado se lo contrastará con 254.

S1 requiere 28 direcciones IPv4, eso significa que esta subred en el espacio de “sufijo para host” requerirá de 5 bits pues 2^5 = 32 direcciones IPv4.

S2 requiere tan sólo 2 direcciones IPv4 (enlace punto a punto entre R1 y R2) por lo tanto requerirá de 2 bits en el espacio “sufijo para host” (recordar que además de las direcciones de las interfaces en el enlace, se necesita la dirección de la subred y la dirección de *broadcast* de la subred; por ese motivo se necesitan 2 bits)

S3 contempla 45 direcciones IPv4; siguiendo el mismo razonamiento, necesita de 6 bits en el espacio “sufijo para hosts” pues 2^6 = 64 direcciones IPv4.

Por último, S4, necesitará 6 bits en el espacio “sufijo para host” pues 2^6 = 64 direcciones IPv4.

La Tabla 2 resume todo lo expresado en cuanto a direcciones IPv4 en las subredes.

|  |  |  |  |
| --- | --- | --- | --- |
| Subred | Addr´s IPv4 | Bits sufijo hosts | Genmask |
| S1 | 32 | 5 | /27 |
| S2 | 4 | 2 | /30 |
| S3 | 64 | 6 | /26 |
| S4 | 64 | 6 | /26 |

Notar que el número REAL de direcciones IPv4 en las subredes suman 164, número menor a las 254 direcciones disponibles, por lo tanto, la división en subredes de factible de realizar.

Otro aspecto importante para destacar es que se ha introducido la máscara para cada una de las subredes (Genmask). Esos números son fáciles de obtener; por ejemplo, analicemos el caso de S1, que requiere 5 bits de “sufijo para hosts”; la Figura 2, muestra cómo obtener la máscara para esta subred. Con las demás subredes se procede el mismo modo.

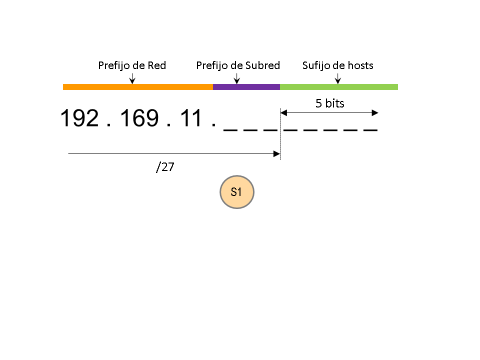


Figura 2: Obtención del prefijo de red para la subred S1

La Figura 3 ilustra el procedimiento para crear las subredes. La única precaución a tomar es que no se repitan direcciones IPv4 en los distintos bloques; y ese aspecto se administra con distintos prefijos de subred. La solución ilustrada es una de las alternativas, podría haber comenzado con la definición de S3 en vez de S1.

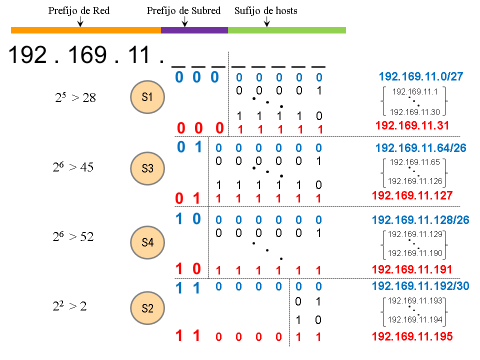


Figura 3: Subredes de máscara variable (VLSM)

La Figura 5 muestra la topología que Uds. deberán configurar, considerando los números de direcciones IPv4 mostrados en las interfaces. Comen el ejercicio de máscara fija, consideré una sola PC por subred (PC1, PC3 y PC5) y compruebe conectividad utilizando el comando ping desde PC1.

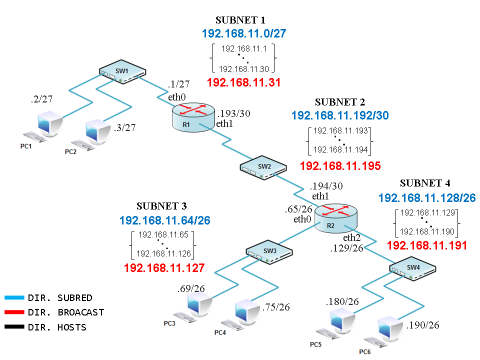


Figura 5: Topología para experimentación de VLSM

Los routers en el backbone de Internet tienen una ruta a todas las redes IPv4 en el planeta. Cuando se agotaron las redes de Clase A y B, se comenzaron a adjudicar redes de clase “C” (aproximadamente 2^21). En ese contexto se produjo un crecimiento exponencial de las tablas de rutas; y quedó en evidencia que el sistema de asignación de redes “clasfull” o basado en clases, no podía continuar. Surgió de este modo Classless Interdomain Routing (CIDR) en septiembre de 1993 (RFC 1519).

**Ruteo entre Dominios sin Clases (CIDR)**

El RFC ha dejado en evidencia que pronto se enfrentaran varios problemas de gran escala:

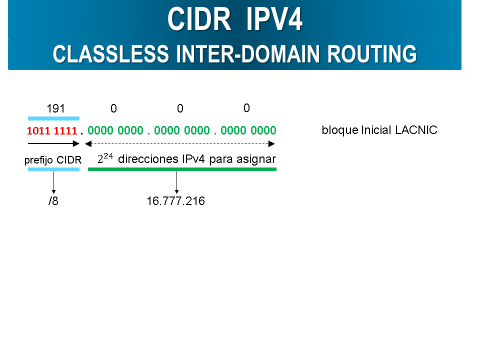
1. Agotamiento del espacio de redes de Clase A y Clase “B”

2. Crecimiento de las tablas de enrutamiento en los router´s del núcleo de Internet (backbone)

3. Eventual agotamiento del espacio de direcciones IPv4.

El diseño y despliegue de CIDR tenía la intención de resolver los problemas escritos y proporcionar un mecanismo para frenar el crecimiento global de las tablas de enrutamiento en el backbone de Internet, y a la vez reducir la tasa de consumo de direcciones IPv4. CIDR no intentó resolver el tercer problema considerado como de largo plazo.

LACNIC debió asignar bloques de direcciones IPv4 a los proveedores de servicios de Internet (ISP) a partir de los bloques administrados (/8 mostrados en la Tabla 1). Notar que, por cada bloque se puede asignar un total de 16.777.216 direcciones IPv4

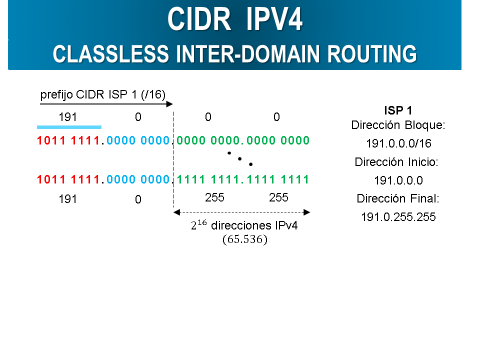


Para que entiendan, vamos a suponer que LACNIC comenzó a administrar el bloque mostrado en la Figura 2 y recibió peticiones desde los ISP según la Tabla 2. Mostraremos como es el proceso de asignación considerando que la adjudicación de los bloques IPv4 responden a un orden estrictamente cronológico (primero ISP 1, luego ISP 2, etc.).

|  |  |
| --- | --- |
| ISP 1 | 65.536 |
| ISP 2 | 32.768 |
| ISP 3 | 16.384 |
| ISP 4 | 2.048 |
| ISP 5 | 4.096 |
| ISP 6 | 2048 |

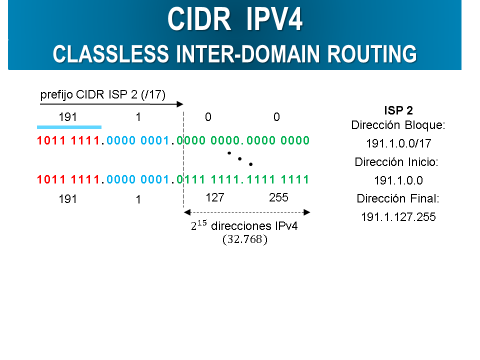
Tabla 2

Para proveer 65.536 direcciones IPv4 al proveedor ISP 1, LACNIC debe partir de un prefijo /167

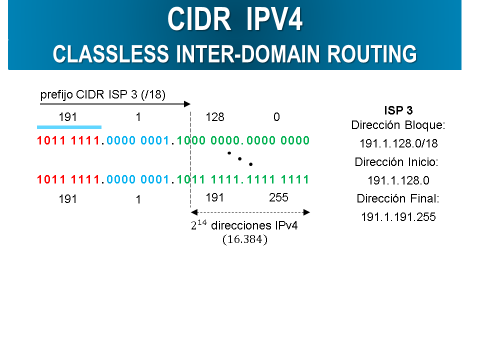


LACNIC asigna 65.536 direcciones a ISP 1

El proveedor de servicios de Internet (ISP 2) requiere de 32.768 direcciones IPv4. Sí se lo compara con el bloque asignado al ISP 1; requiere la mitad de direcciones. Ese detalle ya nos indica que el prefijo de red CIDR para ISP 2 será de 17 bits o /17 y quedarán 15 bits para que disponga de 2^15 = 32.768 direcciones IPv4 (tal como lo solicitó a LACNIC). La Figura 4 ilustra el proceso de asignación.



El proveedor de servicios de Internet (ISP 3) requiere de 16.384 direcciones. Ello significa que LACNIC necesitará de un prefijo /18 para obtener los 14 bit que le proporcionen al ISP 3 las 2^14 = 16.384 direcciones (ver Figura 5).



ISP 4 sólo requiere 2048 direcciones IPv4 (2^11), entonces el prefijo CIDR del bloque será /21 (ver Figura 6).

ISP 5 (4096 direcciones IPv4, 2^12), LACNIC deberá asignar un prefijo CIDR /20 (ver Figura 7).

ISP 6 (2048 direcciones IPv4, 2^11), LACNIC deberá asignar un prefijo CIDR /21. Notar que ha quedado un bloque de direcciones no asignadas entre el espacio otorgado por LACNIC al ISP 4 e ISP 5.

**Conclusión**

En total LACNIC asignó 122.880 direcciones IPv4, el equivalente a 480 redes de la ex Clase “C”. Ello hubiese requerido 480 entradas a las tablas de rutas en el núcleo de Internet; pero CIDR las resume en sólo 6 entradas:

191.0.0.0/16 ISP 1

191.1.0.0/17 ISP 2

191.1.128.0/18 ISP 3

191.1.192.0/21 ISP 4

191.1.208.0/20 ISP 5

191.1.200.0/21 ISP 6

**Protocolo de Resolución de Direcciones – ARP**

Desde la perspectiva del protocolo IPv4, un host tiene tres alternativas de comunicación bien definidas:

1. Con otro host en su propia red IPv4

2. Con un host en otra red IPv4

3. Con la dirección IPv4 de localhost.

¿Qué ocurre cuando una PC debe enviar un datagrama IPv4?

No existen muchas alternativas, el datagrama debe ser transferido a la capa de enlace de datos, para finalmente convertirse en una trama Ethernet y posteriormente ser enviada por la capa física. ¡Pero existe un problema! ¿Cómo hace PC1 para determinar la dirección MAC destino de la entidad receptora de la trama?

Para responder al interrogante planteado; el RFC 826 definió el protocolo ARP (en los años 80´). La operación del protocolo ARP se puede definir en pocas líneas:

a) El protocolo ARP generará un ARP request (solicitud) en PC1 que se encapsulará en una trama Ethernet; dirigida a la dirección de difusión de L2: FF: FF: FF: FF: FF: FF.

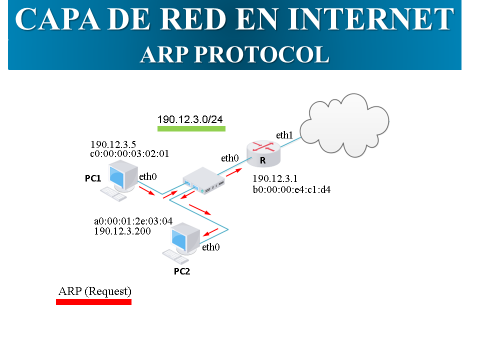
b) Todas las interfaces en la red (o subred) IPv4 deberán procesar la trama Ethernet y posteriormente entregar el payload al protocolo especificado en el campo Type (en Ethernet II) o al protocolo especificado en DSAP si se tratara de una trama IEEE (en cualquiera de los casos el protocolo es ARP).

c) La interfaz que reconozca su propia IP address en el mensaje ARP, generará un ARP reply (respuesta) especificando su dirección MAC. Es importante destacar que el ARP reply se encapsula en una trama Ethernet a la dirección “unicast” de PC1.

d) PC1 recibirá la trama Ethernet, y luego de quitar el encabezado y el tráiler (CRC) pasará el payload al protocolo ARP. De ese modo L2 de PC1 determina la dirección MAC destino para poder enviar el datagrama IPv4.

Las opciones de comunicación listadas al comienzo de esta sección las determina la capa de red, al evaluar las opciones de rutas.

Sea la topología de red mostrada en la Figura 1. Se considera en primer lugar que PC1 intenta enviar un datagrama IPv4 a PC2 (ambas PC´s coexisten en la red 190.12.3.5/24). Como la dirección IPv4 destino es 190.12.5.200 (dirección IPv4 de PC2), PC1 consulta su tabla de ruteo (ver Figura 2) y determina que es un host en su misma red. Para averiguar la dirección MAC de PC2, PC1 envía un ARP request (ver Figura 1) encapsulado en una trama Ethernet y dirigido a la dirección de difusión de capa 2. Notar que el switch reenvía por todos sus puertos la trama Ethernet porque está dirigida a la dirección difusión de capa 2.



PC2 al recibir la trama Ethernet, le transfiere el contenido (payload) al protocolo ARP y reconoce en el encabezad ARP su propia dirección IPv4. Inmediatamente genera un ARP reply, encapsulado en una trama Ethernet dirigido a la dirección unicast de PC1 (c0:00:00:03:02:01) en el que le contesta (reply) “190.12.3.200, mi dirección MAC es a0:00:01:2e:03:04”. Recién ahora PC1 puede envíale el datagrama a PC2.

¿por cada datagrama IPv4 que necesita ser enviado hay que acudir al protocolo ARP? La respuesta es NO

**HEADER IPv4**

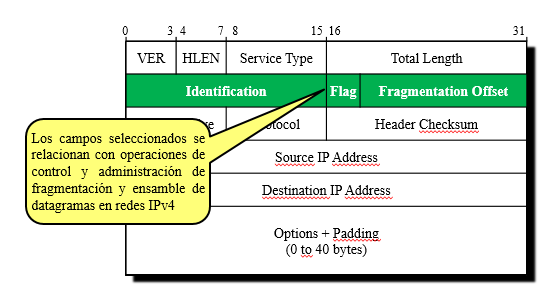


**El encabezado o header del datagrama o paquete IPv4.** El primer campo de 4 bits, indica la **VERsión** del protocolo de Internet. Sí el valor de VER es 0100 se tratará de un datagrama IPv4, sí el valor es 0110 se tratará de un datagrama IPv6.

El campo **Header Lenght** especifica la **longitud del encabezado** IPv4 . ¿Cómo hacerlo con tan sólo 4 bits? Es sencillo, noten que los campos **no opcionales** suman cinco (5) filas y cada fila tiene 4 bytes, entonces el campo HLEN **nunca** puede ser inferior a 20 bytes, cuando este sea el caso, HLEN tendrá el valor será 0101 (5 en decimal). Lo concreto es que para conocer la longitud en bytes del header IPv4 se tiene que multiplicar el valor de HLEN por 4. Ahora si, HLEN es 0101 (5) x 4= 20 bytes. Otra conclusión interesante es determinar la longitud máxima de un datagrama IPv4, y ello sucederá cuando HLEN alcance su máximo valor; 1111 (15 en decimal). Entonces 15 x 4 = 60 bytes. Como consecuencia en IPv4 no puede existir un header que supere los 60 bytes.

**Service Type**, este campo de 8 bits fue pensado para marcar datagramas y elaborar un sistema de prioridades que permitan a los sucesivos routers en al path (camino) hasta el destino final, clasificar los datagramas en base a su prioridad y reenviarlos en función de ella.

**Total Lenght,** longitud total del datagrama (campo de 16 bits). Incluye la longitud del encabezado y de los datos (payload) que transporta el datagrama. IPv4 fue pensado para conectar distintas tecnologías de red mediante routers. La máxima longitud de un datagrama IPv4 bajo Ethernet es de 1500 bytes. Otro aspecto importante es que el valor mínimo de Total Length obligatoriamente debe ser mayor a 20 bytes

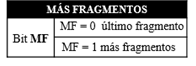


**IPv4 permite la fragmentación de datagramas**. Rx a partir del datagrama original, se generan fragmentos de acuerdo al MTU del enlace (512 bytes) y cada uno de los fragmentos tendrá su propio encabezado IPv4. Cuando los fragmentos alcancen el destino final, la capa de red tiene que ensamblarlos y convertirlos en el datagrama original. Los campos seleccionados sirven para este propósito

Fragmentos de un datagramas fragmentado poseen el mismo valor en el **campo identificación.**



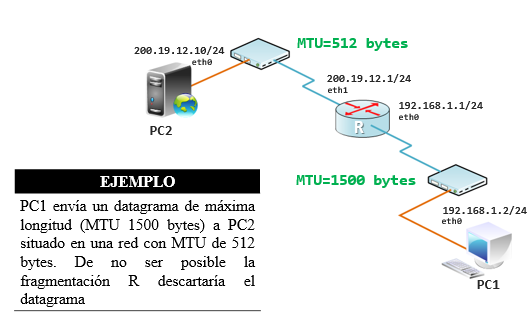
**Flag** para el control de fragmentación. se compone de 3 bits, pero sólo utiliza 2. En la generalidad de los casos DF = 0 (se permite la fragmentación). El bit MF sirve para indicar al receptor final del datagrama (dependiendo de su valor) la recepción del último fragmento (MF=0). Este último párrafo merece una especial atención, la capa de red en Internet es no orientada a la conexión, los fragmentos pueden llegar al destino final en una secuencia distinta a la que fueron emitidos.

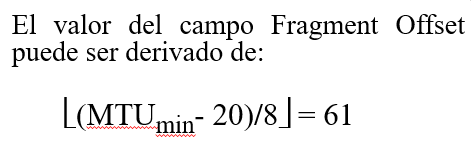
¿cómo hace IPv4 para reconstruir el datagrama original y transferir el payload al protocolo que corresponda?

Los diseñadores de IPv4 incluyeron el campo **“Fragmentation Offset“** para que el receptor “final” de los fragmentos IPv4 , pueda ordenarlos en la secuencia correcta para luego entregar el datagrama original (ensamblado) al protocolo especificado en el campo “Protocol”

Ejemplo:

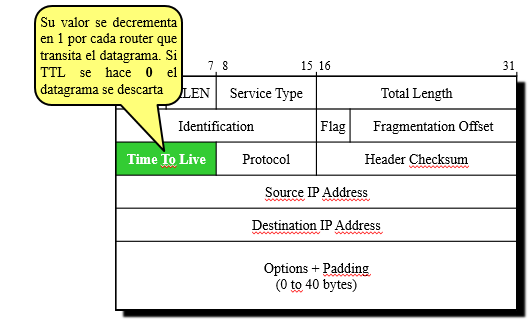
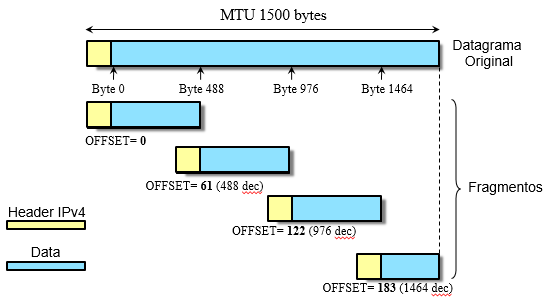


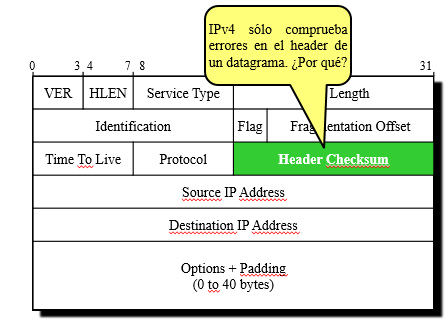
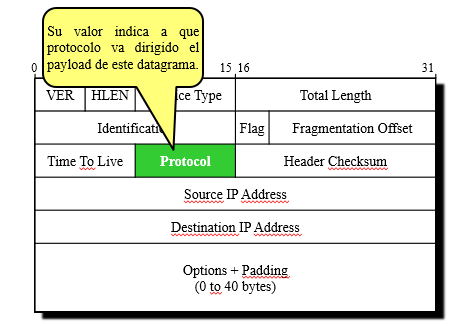
El router R vincula dos redes IPv4 con distintos MTU´s. La inferior (192.168.1.0/24) con un MTU de 1500 bytes, y la 200.0.0.0/24 con un MTU de 512 bytes. Desde PC1 se enviará a PC2 un datagrama IPv4 con un valor en el campo “Total Length” de 1500 bytes. R necesariamente deberá fragmentar el datagrama y PC2 recibirá los fragmentos del datagrama original.



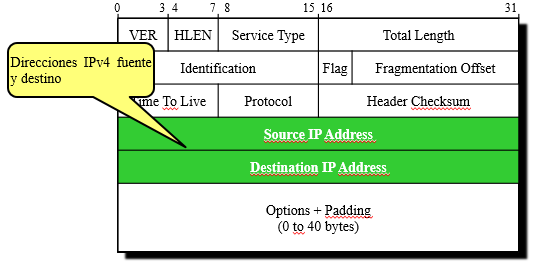
La fórmula presentada se deriva del siguiente razonamiento considerando los datos de la topología anterior:

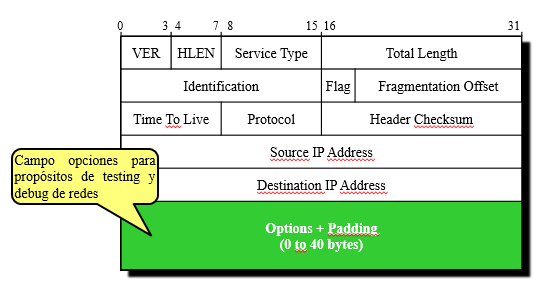
1. El MTU es la máxima cantidad de bytes que la capa de enlace puede encapsular
2. Se considera que la capa de enlace encapsula datagramas IPv4 cuyo header es de 20 bytes
3. Como el campo “Fragment Offset” expresa el desplazamiento (offset) de bytes de datos respecto del datagrama original, es necesario restarle al MTU (512 bytes) los 20 bytes del header IPv4
4. Al valor del paso 3) es necesario dividirlo por 8 por la razón de que el campo Fragment Offset tiene una longitud de 13 bits
5. Se debe aplicar la función “suelo” (aproximar al entero por debajo)
6. El resultado de la operación en 5) será 61
7. Si se considera que el datagrama IPv4 original tiene una longitud total de 1500 bytes (campo Total Lenght), el payload será de 1480 bytes (20 bytes de header IPv4)
8. El offset del primer fragmento será 0, y tendrá 488 bytes de datos (bytes 0 a byte 487)
9. El offset del segundo fragmento será 488 bytes (campo Fragment Offset =61)
10. El offset tercer fragmento será 976 bytes (campo Fragment Offset =122)
11. El offset del cuarto y último fragmento será 1464 (campo Fragment Offset =183) y tendrá un payload de apenas 16 bytes.
12. Se puede comprobar sí lo escrito es válido: 3 fragmentos de 488 bytes de payload más un fragmento de 16 bytes de payload. 488\*3 + 16 = 1480 bytes (el payload del datagrama original)



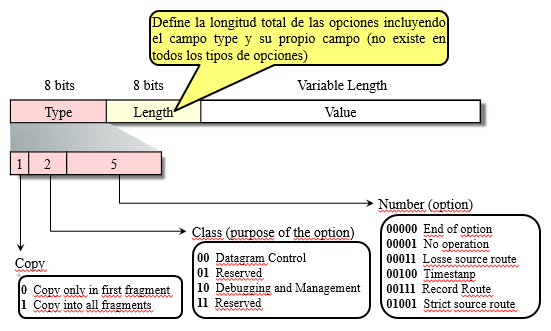
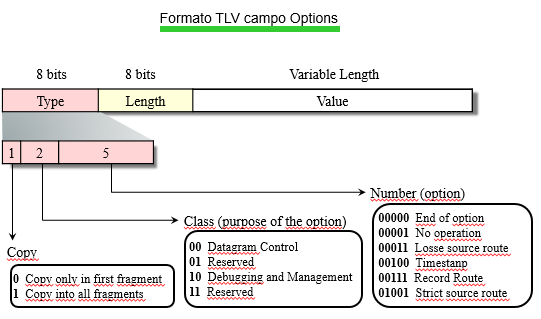


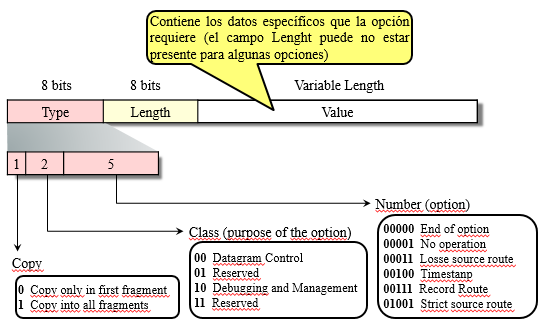
El método de comprobación de errores por **Cheksum** es menos robusto que el método de CRC. Se basa en aritmética binaria módulo 1 (significa que tienen sentido los bits de acarreo). Simplemente se aplica al header IPv4 (no a los datos encapsulados) por una razón muy sencilla. En capa de Transporte, TCP aplica un Checksum sobre el header y los datos TCP. Entonces; no tiene sentido realizar un nuevo checksum adicional.



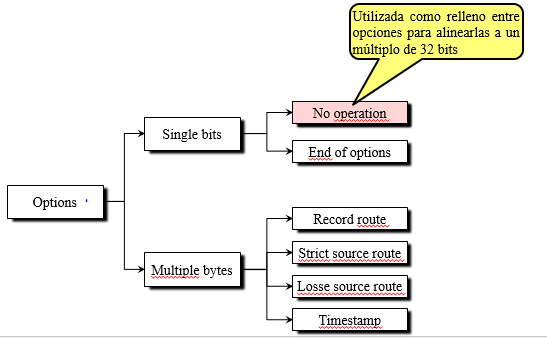
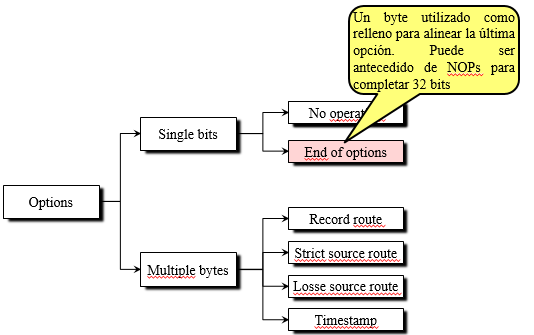


**La sigla TLV**.significa una ordenación según Type, Length y Value.

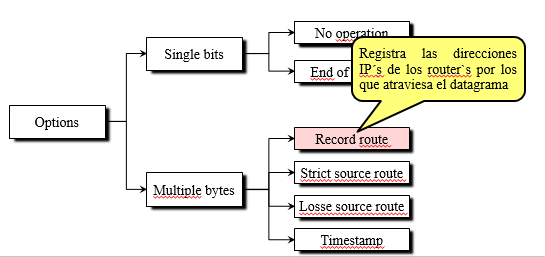
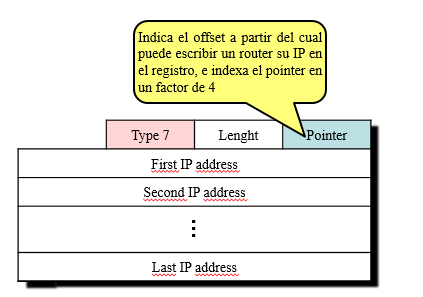


 Las opciones en IPv4, tal como se expresó; sirven para el propósito de testing y debugging. La operación NOP (No OPeration) está relacionada con el campo HLEN. Recuerden que HLEN es un campo de 4 bits que en el caso de un header de tamaño mínimo (sin opciones) su valor es 0101 (expresa que la longitud del header tiene 20 bytes: no puede existir un header IPv4 menor a 20 bytes). Sin embargo, hay opciones que no completan en su última fila 4 bytes y de ese modo se entra en conflicto con lo que especifica HLEN

Es importante destacar que en el campo opciones puede existir más de una opción. El campo NOP se utilizará cuando cada opción necesite ser alineada con palabras de 32 bits (4 bytes) . La opción NOP es un código de 1 byte.

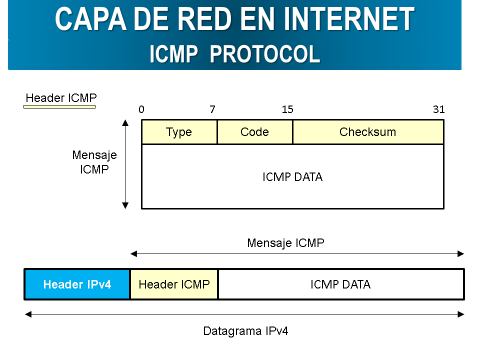
 

**Ninguna de las dos posee campo length.**

El protocolo IP provee un servicio “host to host” no orientado a la conexión. Los datagramas pueden perderse en router´s congestionados, pueden sufrir errores en enlaces defectuosos, pueden agotar su “tiempo de vida” (campo TTL=0) y ante cualquiera de las situaciones mencionadas; el o los datagramas afectados se descartarán. *Internet Control Message Protocol* (ICMP) es un protocolo de capa de red destinado a reportar o notificar errores al emisor original de datagramas IPv4. ICMP siempre se encapsula en un datagrama IPv4. Tenga presente que no se generan mensajes de error ICMP por la pérdida de mensajes ICMP (de error). En el caso de error en datagramas fragmentados, sólo se generará un mensaje ICMP para el fragmento cuyo valor del “fragment offset” sea 0 (primer fragmento).

**Protocolo ICMP**

Figura 1: Estructura de un mensaje ICMP y encapsulación en IPv4

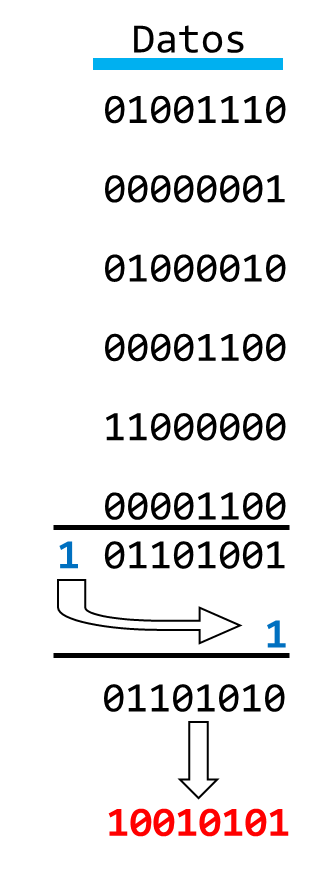
Del encabezado se deduce que hay distintos tipos de mensajes ICMP (campo type de 8 bits), y por cada tipo múltiples variantes contempladas en el campo “code” que brindan más detalles sobre el tipo considerado.

**Mensaje ICMP Unreacheable (Tipo 3 Código 0)**

El RFC 792 declara que en el área de datos (ICMP Data en la Figura 1) de un mensaje ICMP (Tipo 3) contendrá el encabezado del datagrama IPv4 que generó el error más 64 bytes de datos adicionales que siguen al encabezado IPv4. Los 64 bytes le sirven al host para entregarlos al proceso apropiado (notar que generalmente contendrá un encabezado de un protocolo de capa superior, por ejemplo, encabezado TCP o UDP).

El campo checksum servirá para todos los protocolos que implementan “detección de errores” (datagrama IPv4, ICMP, TCP, UDP, etc.). En el caso de ICMP el checksum (campo de 16 bits) se aplica al encabezado y a los datos.

En host emisor antes de transmitir un mensaje (ICMP, IPv4, TCP, UDP, etc.) lo agrupa en bloques de 16 bytes. Para simplificar la explicación consideraremos bloques de 8 bits, el procedimiento es el mismo y la explicación servirá para el caso general. La Figura 3 ilustra los datos a comprobar y el área donde se escribirá la suma de comprobación.

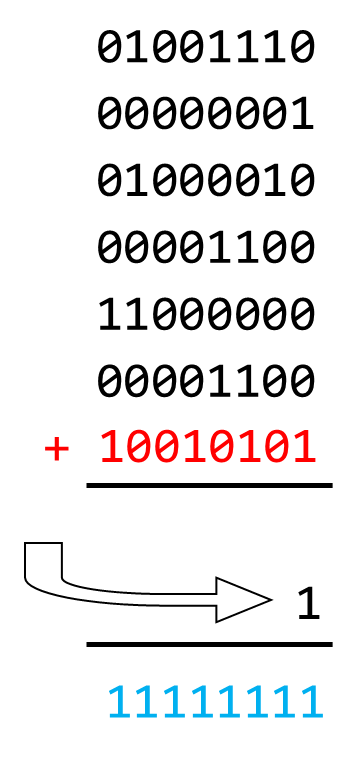


🡨Para calcular el ckecksum (en el emisor) se suman los bloques considerando bits de acarreo. Una vez hallada la suma se realiza el complemento de la suma obtenida (se invierten los bits) tal como se observa en la Figura 4 (color rojo).

El complemento de la suma se escribe en el campo checksum y se envía el mensaje al receptor (ver Figura 5). (abajo ).



El receptor del mensaje realizará la suma de todos los bloques (incluido el checksum) y al resultado le efectuará el complemento A1(no mostrado en la Figura 6). De no existir error lo que se obtiene en el receptor del mensaje es:



Aca falta hacer el complemento, la condición de no error ser 00000000 en vez de los unos.

El método checksum para detectar errores se implementa a partir de capa de red (inclusive) por la sencilla razón de que tiene mejor performance en software que el método CRC

**Mensaje ICMP Unreachable (Tipo 3 Código 1)**: En el caso general, un datagrama IPv4 será enviado a un host en una red remota. Si el host destino se encuentra fuera de servicio, se generará un mensaje ICMP desde la red remota y el encargado de enviarlo será el router en contacto con la red del host inactivo.

**Mensaje ICMP Unreachable (Tipo 3 Código 2)**: Un mensaje ICMP Tipo 3 código 2 se genera cuando se envía un datagrama IPv4 desde un host emisor a un host receptor. El protocolo IPv4 tiene que entregar su payload al protocolo descrito en el campo “protocol” del encabezado IPv4; pero sucede que ese protocolo no fue instalado en el host destino. Cuando este es el caso, se generará un mensaje ICMP indicándole al emisor original del datagrama “protocolo inalcanzable.

**Mensaje ICMP Unreachable (Tipo 3 Código 3) (**puerto inalcanzable)

**Mensaje ICMP Unreachable (Tipo 3 Código 4)** (Fragmentation needed)

**Mensaje ICMP Unreachable (Tipo 3 Código 5)**(“source route failed”)

**Mensaje ICMP Time Exeeded in Transit (Tipo 11 Código 0):** Cuando se abordó el encabezado IPv4 se trató el campo “Time to Live” (tiempo de vida) cuya extensión es de 8 bits. El propósito de TTL (acrónimo de Time to Live) es que un datagrama que es inyectado en Internet, como máximo puede ser reenviado (*forwarding*) 255 veces (en el caso extremo). Dependiendo de los protocolos que se encapsulan en IPv4, TTL puede tomar valores intermedios. Cada vez que un datagrama es reenviado por un router, el valor del campo TTL se le resta uno (1). En el supuesto de que un datagrama IPv4 ingrese por una interfaz de un router con TTL=1, ese router descartará el datagrama y generará un mensaje ICMP dirigido al emisor original para notificar *“Time Exceeded in-Transit”*.

**Mensaje ICMP Redirect (Tipo 5 Código 0):** *“redirect”* con códigos que señalan *“redirect for network”* (code 0), *“redirect for service in network”* (code 2) y *“redirect for host”* (code 3).

**ICMP (Tipo 8 Código 0 y Tipo 0 Código 0) – PING**

**HAY EJEMPLOS DE TODOS LOS MENSAJES EN EL WORD DE TEORIA 9.**

Protocolos de Encaminamiento de Borde Interior (IGP).

La frase “borde interior” se refiere al establecimiento de rutas automáticas en el interior de un borde delimitado por un dominio administrativo. Ello significa que la configuración de los protocolos de encaminamiento dinámicos son responsabilidad de una autoridad centralizada (el administrador de red).

Los protocolos de borde interior (Internal Gateway Protocol) se basan en dos algoritmos:

1. Vector distancia (Bellman–Ford algorithm)

2. Estado de enlaces

**Protocolos basados en Vector Distancia**

Existen algunos protocolos basados en este algoritmo. Los hay de fuentes abiertas (código fuente) y lo hay de código cerrado. En protocolos de encaminamiento dinámicos, basados en el algoritmo vector distancia, cada uno de los router´s en una topología dada, solo conoce el costo en alcanzar a sus router´s vecinos “directamente conectados”.

El propósito del algoritmo es encontrar el camino de “menor costo” entre cualquiera de las subredes conectadas a los router´s.

Cuando se expresa “menor costo”; se puede referenciar a:

* Trayectoria de menor número de saltos (hops) en lo referido router´s
* Trayectoria de menor retardo
* Trayectoria de mayor velocidad en bits/s
* Trayectoria de menor número de pérdidas

El algoritmo es asincrónico y una vez en estado operativo, los router´s difundirán en distintos intervalos de tiempo a sus vecinos (solo a sus vecinos) su tabla de rutas.

**RIP (Routing Information Protocol)**

Existen dos versiones del protocolo RIP. RIPv1 fue actualizada a RIPv2 por problemas que presentaba la primera versión del protocolo. A continuación, el principal problema de RIPv1:

* “split-horizon” (horizonte dividido).
* RIPv1 no soporta VLSM (subredes de máscara variable) y los router´s intercambian información de encaminamiento en la dirección de difusión “no guiada” de IPv4

**Protocolos basados en Estado de Enlace**

OSPF (Open\_Shortest\_Path\_First) es un protocolo de código abierto que funciona según el algoritmo de estado de enlace; o algoritmo de Dijkstra. La implementación de OSPF respondió a las limitaciones del protocolo RIP (protocolo de lenta convergencia, ruta de menor costo en saltos de router´s, y limitaciones en la escalabilidad: aplicable a infraestructuras de red que no superen los 15 saltos).

**Algoritmo de Dijkstra**

El algoritmo de Dijkstra, opera intercambiando información con todos los nodos en un grafo (nodos es sinónimo de router´s en una red bajo un único dominio administrativo).

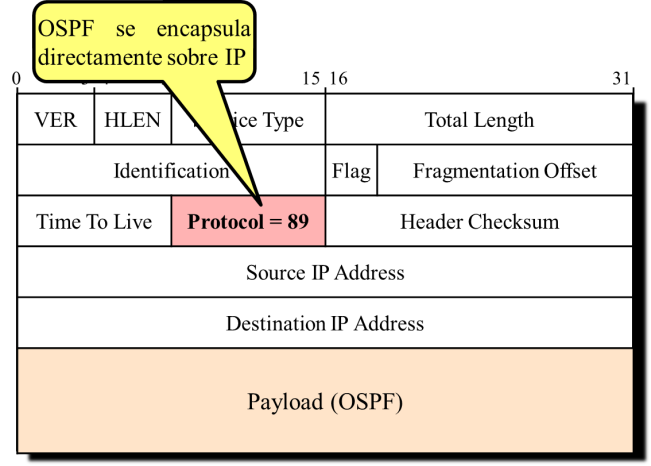
Desde el punto de vista de la información que intercambian los router´s, el algoritmo tratado se puede clasificar como de alcance global (la información genera un nodo “i” se difunde en el grafo). En cambio, el algoritmo vector distancia necesita de sucesivas iteraciones entre router´s vecinos para converger a la tabla de rutas de menor costo.

El concepto básico del enrutamiento de estado de enlace es que cada nodo construye un mapa de la conectividad a la red. Un nodo construye el grafo completo de la topología y los costos asociados a cada enlace en la red. Posteriormente, cada nodo calcula de forma independiente la ruta de menor costo a cada destino posible. La colección de las rutas de menor costo se traducirá en la tabla de enrutamiento del nodo.

Note que en los protocolos basados en el algoritmo “vector de distancia” la tabla de rutas de menor costo en un router Ri se construye en base a "rumores" de los vecinos directamente conectados (adyacentes). En otras palabras, los nodos no tienen una imagen completa de la red como en el caso de protocolos basados en el algoritmo de estado de enlace. Los protocolos basados en el algoritmo de estado de enlace, no publican solo rutas, sino que anuncian información de la topología de red.

**OSPF (Open Shortest Path First)**

OSPF se clasifica como un protocolo de enrutamiento de estado de enlace. OSPF fue especialmente diseñado para encaminamiento intra – dominio en una red IPv4. A diferencia de RIP, que utiliza UDP como protocolo de transporte para el intercambio de mensajes entre router´s; el protocolo OSPF se encapsula directamente sobre IPv4.

Términos que tienen especial significado para tratar con OSPF:

1. Sistema Autónomo: un grupo de router´s que intercambian información bajo un protocolo de encaminamiento común y bajo una autoridad administrativa.
2. Router´s vecinos: dos router´s que tienen interfaces a una red común. La relación entre vecinos es descubierta y mantenida en forma dinámica a través de mensajes OSPF “hello”.
3. Adyacencia: relación entre router´s vecinos, seleccionados para intercambiar información. No todos los pares de router´s vecinos son adyacentes.
4. Link State Advertisement (LSA): unidad de datos que describe el estado local de un router. Incluye el estado de las interfaces y las adyacencias del router. Cada anuncio de estado de enlace se inunda en todo el dominio de enrutamiento. Los anuncios de estado de enlace recopilados forman la base de datos de estado de enlace del protocolo o LSDB (Link State Data Base).
5. Flooding: la traducción a nuestro idioma es “inundación” y es la forma que OSPF utiliza para distribuir y sincronizar la LSDB entre los router´s de la red.

En el encabezado estándar de OSPF hay un campo de especial interés, el campo “type” que consta de 8 bits. De acuerdo a su valor se definen distintos tipos de mensajes OSPF, cada uno de los cuales agrega un encabezado extra al encabezado estándar. Un mensaje de especial relevancia es “hello” (campo type igual a 1). Un router envía mensajes “hello” para descubrir vecinos, por cada una de sus interfaces. Mensajes “hello” se envían regularmente de acuerdo a la configuración del protocolo. Durante cada intervalo “hello”, un router adquiere conocimiento acerca de relaciones de vecindad. La dirección IPv4 fuente será la de la interfaz por la que envía el mensaje. Respecto a la dirección destino del datagrama IPv4 (en redes Ethernet o enlaces punto a punto), se utiliza la dirección multicast 224.0.0.5 (denominada AllSPFRouters). Cuando un router recibe un paquete “hello” verifica múltiples parámetros en el encabezado del mensaje, en caso de coincidencia con los equivalentes a la interfaz receptora, el mensaje será válido, de otro modo el mensaje se descarta y la relación progresa. Cuando un router envía un mensaje “hello” incluye su identificación (router ID) y la de los vecinos desde los cuales ha recibido paquetes “hello”. Un nodo al recibir un paquete “hello” verifica que existe su propio router ID (entre otros) entonces reconoce que existe una comunicación bidireccional con ese vecino y está en condiciones de intentar una adyacencia completa, Una adyacencia completa permite intercambiar información sobre la topología de la red.

En la etapa de “adyacencia completa” cada router crea una base de datos detallando información sobre la red a partir de los anuncios de estado de enlace (LSA) generados por el, y por otros router´s en la red. La base de datos se denomina LSDB y contiene una lista de: router´s, los enlaces entre router´s, y las redes o subredes conectadas a los router´s. En base a esa información cada router ejecuta el algoritmo de estado de enlace (Dijkstra) para calcular el camino de menor costo para alcanzar cada una de las redes o subredes en la topología considerada.

Resulta evidente que todos los mensajes LSA recopilados por un router, forman la base de datos de estado de enlace y el proceso de “flooding” asegura que cada router en la red, tenga la misma colección de LSA´s. Como consecuencia luego de un período de convergencia todos los router´s en la red tendrán una LSDB idéntica. En otras palabras, cada router en tendrá la misma descripción de la topología de red.

El significado de direcciones multicast en IPv4 (la traducción a nuestra lengua es multidifusión) que posibilitan la transmisión de “uno a muchos”. Un router enviará un paquete IPv4 a una dirección multicast (campo IPv4 Destination Address), y en el campo Source Address simplemente escribirá su propia dirección unicast (dirección unicast hace referencia a la dirección “única” de una interfaz de red).

Hasta el momento se ha tratado con protocolos de pasarela interior o Interior Gateway Protocol (IGP), que resuelven el problema de ruteo en el interior de un sistema autónomo y reconfigura automáticamente las rutas ante fallas en enlaces o en interfaces de red. En este documento se tratará con BGP (Border Gateway Protocol). BGP es un protocolo de pasarela exterior (EGP) que se encarga de proporcionar información de enrutamiento entre sistemas autónomos.

**Protocolos de Encaminamiento de Borde Exterior (EGP)**

La principal diferencia entre EGP y IGP es el ámbito en que operan. Protocolos IGP lo hacen en el interior de un sistema autónomo, protocolos EGP operan entre sistemas autónomos.

**BGP (Border Gateway Protocol)**

La principal función de BGP es intercambiar información de alcanzabilidad de redes en la forma IP/prefijo (CIDR)con otros sistemas BGP. BGP basa su funcionamiento en un algoritmo “path vector” que es una aproximación del algoritmo vector distancia pero que evita bucles a través de la incorporación de etiquetas de camino (path tagging) que son simplemente números de sistemas autónomos o ASNumbers. BGP utiliza como métrica “el número de saltos entre sistemas autónomos”. El costo entre dos sistemas autónomos contiguos es un salto