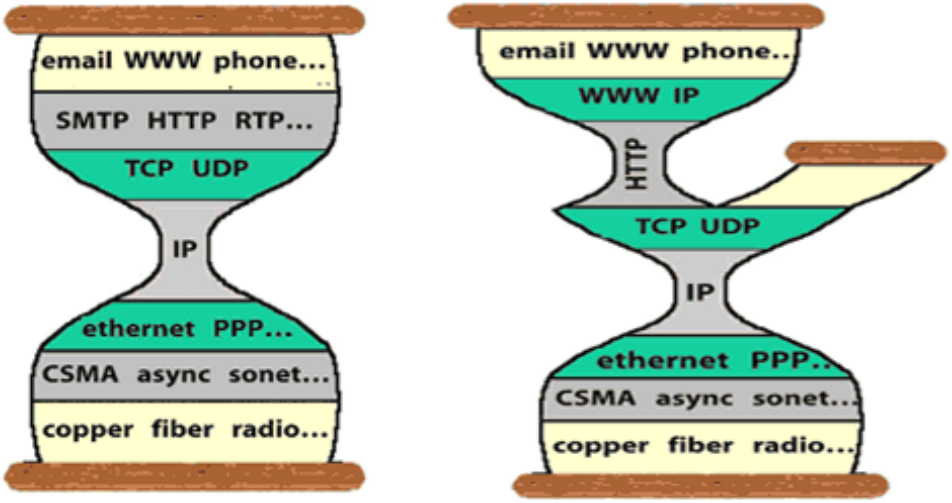
Resumen Capa de Red

Internet

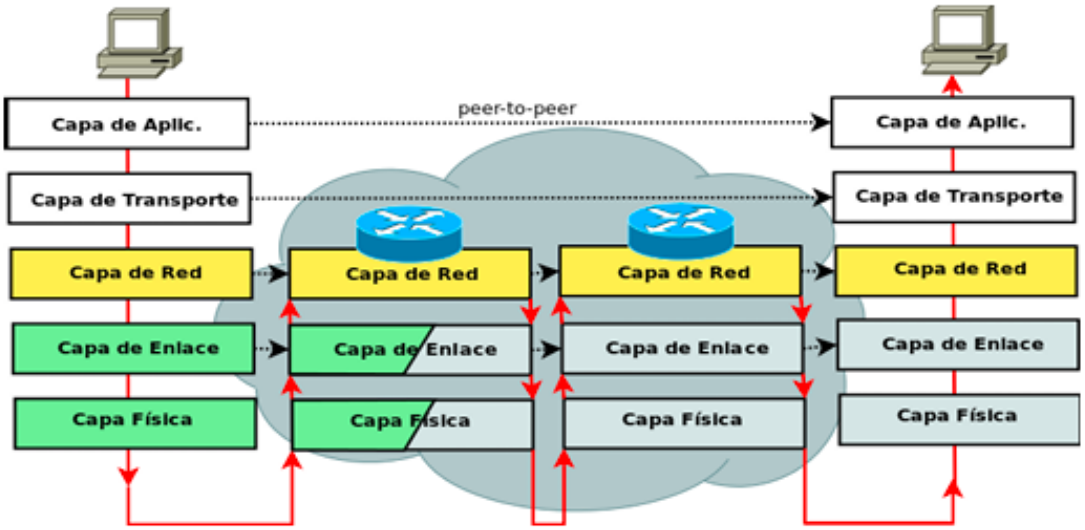
- Internet se ve como un conjunto de redes interconectadas y agregadas de forma “jerárquica”, la capa de red la podemos encontrar presente en todos los protocolos de Internet.



Modelo de Internet actualmente (derecha) debido a la concentración del uso de HTTP

Capa de Red

- Es una de las capas más complejas y se puede descomponer en 2 partes que interaccionan mutuamente, el **plano de datos** y el **plano de control.**
- Utiliza como **PDU** a los **Paquetes/Datagramas IP** (no confundir con datagrama de usuario UDP).
- El dispositivo principal de esta capa es el **router**.
- Es **End-to-End** y tiene que estar implementada en **todos los dispositivos intermedios.**



Funciones de la Capa de Red

- La función principal de la capa de red es engañosamente simple: **transporta paquetes desde un host emisor a un host receptor.** En la realización de esta tarea podemos identificar dos importantes funciones de la capa de red:

Reenvío (forwarding)

- Cuando un paquete llega al enlace de entrada de un router, este tiene que pasar el paquete al enlace de salida apropiado.
- El reenvío es solo una de las funciones implementadas en el **plano de datos.**
- A la hora de hacer el reenvío puede prohibirse a un paquete que salga de un router, o bien el paquete puede duplicarse y enviarse a través de múltiples enlaces de salida.
- Hace referencia a la acción local que realiza un router al transferir un paquete desde una interfaz de un enlace de entrada a una interfaz del enlace de salida apropiado. Tiene lugar en escalas de tiempo muy cortas y se implementa normalmente en hardware.

Enrutamiento (routing)

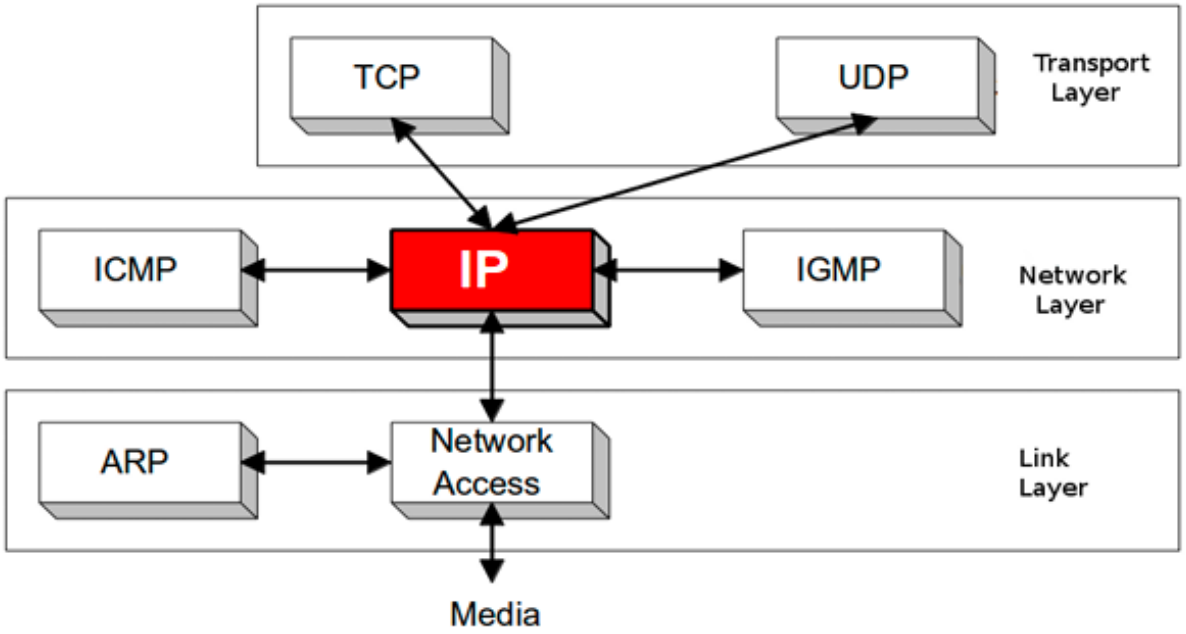
- La **capa de red** tiene que determinar la ruta o camino que deben seguir los paquetes a medida que fluyen de un emisor a un receptor. Los algoritmos que calculan estas rutas se conocen como **algoritmos de enrutamiento.**
- El enrutamiento se implementa en el **plano de control** de la capa de red.
- Hace referencia al proceso que realiza la red en conjunto para determinar las rutas extremo a extremo que los paquetes siguen desde el origen al destino. Tiene lugar con escalas de tiempo mucho más largas y suele implementarse en software.

Protocolos de IP actuales

- Brindan servicios a la capa de transporte y usa servicios de la capa de enlace.
- **Los 2 protocolos IP actuales son IPv4 e IPv6.**
    - No son versiones de un mismo protocolo, **no son compatibles**.

IPv4 (Internet Protocol version 4)

- Es un **protocolo de red no orientado a conexión y de best-effort, no confiable.**
    - Hace todo lo posible para entregar los datos, pero no garantiza que todos los paquetes llegarán al destino, ni de que lo harán en el orden correcto.
- **Funcionalidades de este protocolo:**
    - Direccionamiento (Identificación de los hosts).
    - Ruteo/Forwarding/Switching L3.
    - Mux/Demux de protocolos superiores.
    - Accesorias (Solucionar deficiencias del protocolo).
        - Fragmentación.
        - Otras: como evitar loops (TTL), detección de errores.
- Es el **núcleo** de la **Internet** y **requiere de protocolos "Helpers" (ICMP e IGMP).**



Esquema de IP en TCP/IP

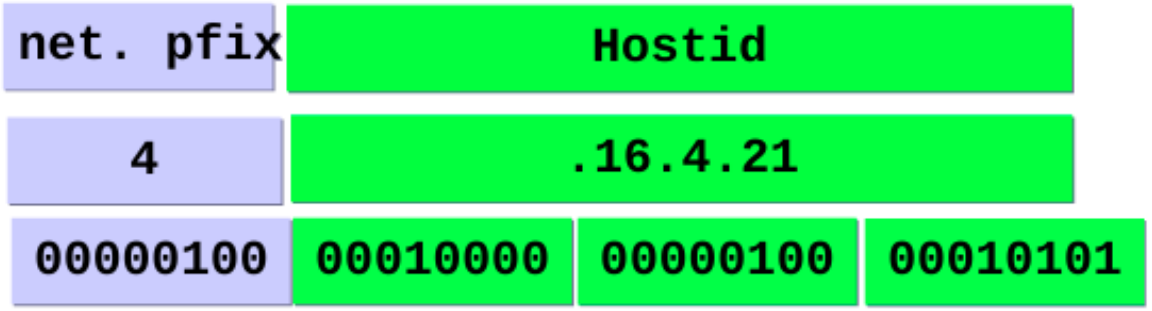
Direccionamiento IP

- La **dirección IP** identifica unívocamente un punto de acceso (interfaz) a la red.
    - Identifica red y luego host dentro de ella. Las redes se conectan a través de routers.
- Un router o un host multi-homed tienen varias IPs. Cada interfaz tiene un valor único.
- **Tienen un significado global en la Internet o privado (local):**

- Las **globales** son asignadas por una autoridad central (actualmente IANA).

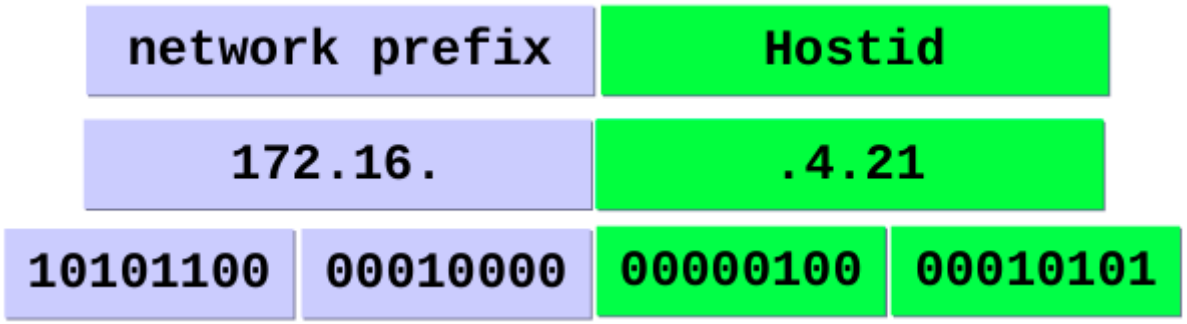
Direcciones IP

- Las **direcciones IP** son números de **32 bits**, expresados en **notación decimal delimitada por puntos byte a byte.**
  - **Son 2^32 (4G) de direcciones puras**, que organizadas en forma jerárquica se reducen
- Para facilidad de los usuarios, mapping con nombres de dominio (DNS).
- **Necesarias para rutear la información por la Internet.**
- Son **direcciones lógicas.**
- **Están codificadas en 2 partes:**
  - Red (Net).
  - Anfitrión (Host).



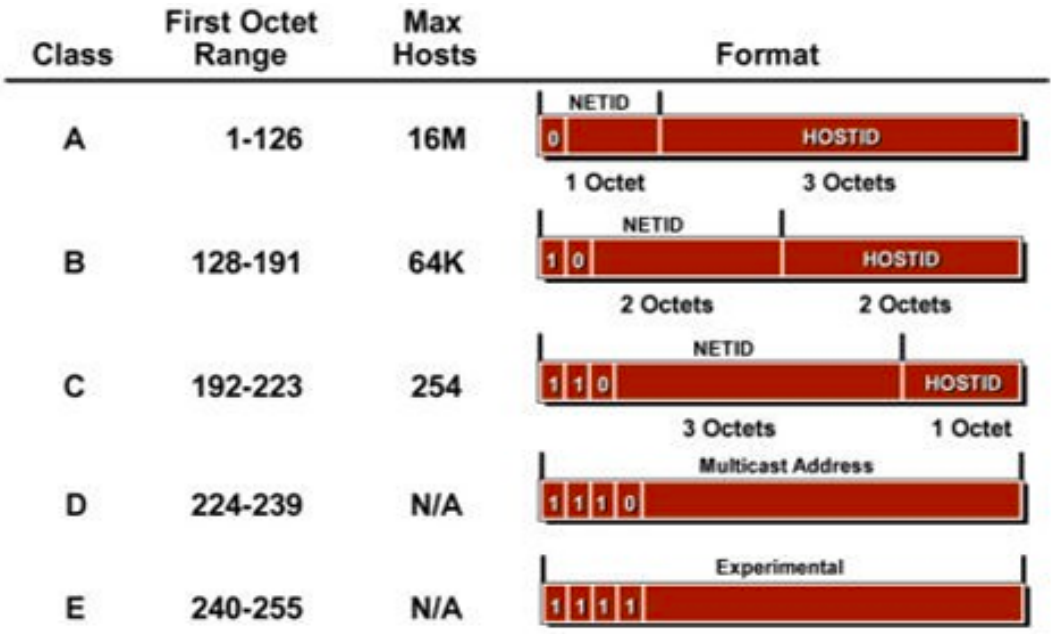
net. pfix → identifica la Red
Hostid → identifica a un host dentro de la red net. pfix

- Hasta 1981 sólo había redes con muchos hosts disponibles. Sin clases. Redes 8 bits. Luego se definen las clases. **Ahora existen clases para diferentes tipos de redes:**
  - **Clases A**: Pocas redes, muy grandes.
  - **Clases B:** Más redes, medianas.
  - **Clases C:** Muchas redes, chicas.



Clase	Primer octeto	Rango	Objetivo	Cant. redes	Cant. hosts
A	0xxxxxxx	0.0.0.0 127.255.255.255	Organizaciones con grandes cantidades de hosts	2^7	$2^{24} - 2$
B	10xxxxxx	128.0.0.0 191.255.255.255	Organizaciones de tamaño mediano y grande	2^{14}	$2^{16} - 2$
C	110xxxxx	192.0.0.0 223.255.255.255	Pequeñas redes	2^{21}	$2^8 - 2$
D	1110xxxx	224.0.0.0 239.255.255.255	Direcciones de multicast	-	-
E	1111xxxx	240.0.0.0 255.255.255.255	Direcciones reservadas (para investigación y otros fines)	-	-

Clases para los diferentes tipos de red

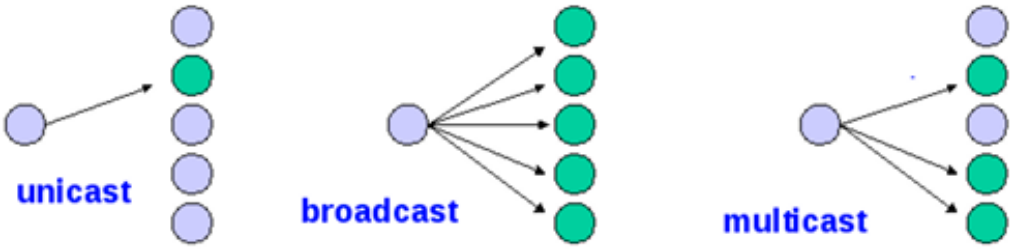


Clases para los diferentes tipos de red + Formato de los identificadores de redes y de hots

- Luego se agrega el **concepto de subredes** que **requieren una máscara.**

Tipos de Direcciones IP

- **Unicast:** Destino a un host/interfaz en particular, son las más comunes.
- **Broadcast:** Destino a todos los hosts en una red.
- **Multicast:** Destinada a un grupo de hosts en una red o varias redes. Clase D.
- **Anycast:** Destinada al primero que resuelva. IPv4 no hay casos especiales.



Tipos de Direcciones IP

Direcciones IP especiales

- Existen ciertas direcciones especiales que tienen un significado específico y existen siempre.
- **Loopback:**
  - Unicast, red clase A.
  - **Empiezan con 127 y representan direcciones que son asignadas a nuestro host local (localhost)**, es decir, no pueden salir de nuestra máquina datagramas con esa dirección.
  - **La más utilizada: 127.0.0.1, localhost.** Aunque podría ser cualquier otra que empiece con 127.
- **Dirección de red:**
  - La primera (zero).
  - Es la dirección que identifica al grupo (net id) que tiene la parte del host toda en 0.
  - Ejm. 172.16.0.0, 192.168.1.0.
- **Dirección de broadcast:**
  - Directed Broadcast: la última (ones).
    - Es la dirección que tiene todos 1 en la parte del host.
    - Ejm. 172.16.255.255, 192.168.1.255.
  - Limited Broadcast: (all ones).
    - Es la dirección que tiene todos 1 en la parte del net id y del host.
    - 255.255.255.255.
- **"Este host", cuando aún no tiene asignada una dirección:**
  - 0.0.0.0 (Utilizada en BOOTP/DHCP).

Direcciones Privadas

- **No tienen significado global, no son únicas.**
- Se utilizan en **Intranets**:
  - Redes autónomas sin conexión a Internet.
- Para **conectarse a Internet** requieren un **proceso de transformación (NAT).**
- **No deberían pasar a la Internet**, son filtradas por routers de borde.
  - 10.0.0.0 – 10.255.255.255 → todas las que empiezan con 10, **1 Clase A.**
  - 172.16.0.0 – 172.31.255.255 → todas las comprendidas entre el rango 172.16 a 172.31, **16 Clases B.**

    - 192.168.0.0 – 192.168.255.255 → todas las que empiezan con 192.168, **256 Clases C.**

Direccionamiento Fijo

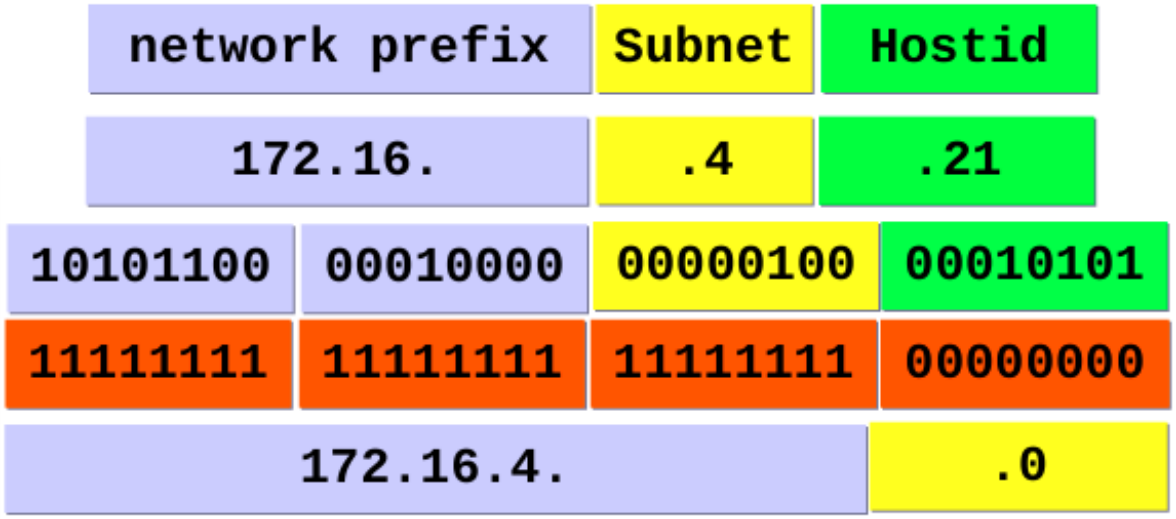
- Tipo de asignación de direcciones IP en una red donde se le otorga a un dispositivo una dirección IP específica y permanente. Esta dirección no cambia cada vez que el dispositivo se conecta a la red, a diferencia de una IP dinámica (asignada automáticamente por un servidor DHCP).
- Es por clase, uso los hosts que tengo según la clase.

Problemas del Direccionamiento Fijo

- Al hacer uso de prefijos de longitud fija por clase, provoca un **uso ineficiente en el espacio de direcciones.**
- **Con muchos equipos se empieza a producir una escasez de direcciones** ya que son fijas.
- El crecimiento acelerado de la Internet, evidencia la **falta de escalabilidad del esquema. Crecimiento de tablas de ruteo en el núcleo de la red.**
- **Codificar la red en la dirección IP implica que si un host cambia de red, cambiará su dirección** (IP Mobility). Problema atacado en IPv4, mejor resuelto en IPv6.
- **Soluciones IPv4:** subnetting, CIDR, NAT, DHCP.
- **Todo definitivamente solucionado en IPv6.**

Subnetting IP

- **Básicamente es dividir un grupo de red en varios subgrupos de red.**
- Se toma una parte del **hostid** para generar **redes dentro de la red.**
    - No podemos tomar parte del net pfix.
- Se agrega una **“máscara” de bits**.
- Para saber la **subred** se aplica un **“AND” lógico** entre la **dirección IP de una interfaz/host** y la **máscara.**



Dirección IP con Subnetting

- **Agrega un nivel más a la estructura:**
  - Red.
  - Subred.
  - Host.
- La división en subredes plantea que si una red de clase desperdicia muchas direcciones IP entonces la misma sea dividida en N subredes más pequeñas que aprovechen mejor el espacio de direccionamiento.
- Las **máscaras** se escriben en **notación decimal o hexadecimal** y **son de 32 bits** como la dirección IP.
  - Se utilizan para saber en una dirección IP **qué bits son de red y subred, y qué bits son de host.**
  - **Tiene 1 en la parte de red y subred.**
  - También pueden escribirse como **longitud de prefijo**: /24.
    - El prefijo nos indica la cantidad de 1 de izquierda a derecha que tiene la máscara.
  - **Ejemplos:**
    - 255.255.255.0.
    - 0xff ff ff 00.
    - 255.255.255.192 /26.
    - 255.224.0.0 /11.
    - 255.255.255.252 /30.
- **Las máscaras defaults son:**
  - **Clase A:** 255.0.0.0.
  - **Clase B:** 255.255.0.0.
  - **Clase C:** 255.255.255.0.
- **Cálculo de cantidad de subredes:**
  - Para calcular la cantidad de subredes que se pueden crear, **utilizamos la fórmula** $\rightarrow Cantidad\ de\ subredes\ = 2^n$**.**
  - Donde "**n**" es la cantidad de **bits "prestados"** del espacio de host para formar la subred.

    - **Ejemplo:**
        - Si tenemos una **red clase C (máscara default /24)** y la dividimos usando **/26**, tomamos **2 bits adicionales de los 8 bits originales del host** dejándonos con $\rightarrow 2^2 = 4\ subredes.$
- **Cálculo de cantidad de hosts:**
    - Se calcula de la **misma forma** que la **cantidad de subredes.**
- **Cálculo de hosts por subred/hosts útiles:**
    - La cantidad de hosts que se pueden asignar en cada subred **se calcula como** $\rightarrow Cantidad\ de\ hosts = 2^h - 2.$
    - Donde **"h"** es la **cantidad de bits de host en la subred**. **Restamos 2** porque una dirección es para la **dirección de red** y otra para la **dirección de broadcast.**
    - **Ejemplo:**
        - Si tenemos una **subred /26**, tenemos **6 bits para los host** $\rightarrow 2^6 - 2 = 62\ host\ por\ subred.$
- **Cálculo de la cantidad de subredes útiles:**
    - Se calcula de la **misma forma** que la **cantidad de hosts por subred/hosts útiles.**
    - **Actualmente** no se resta 2 por las direcciones de red y broadcast ya que si se pueden utilizar, **esto deja la fórmula igual a la del cálculo de cantidad de subredes.**

Subnetting Fijo (FLSM: Fixed-Length Subnet Masking)

- En el **subnetting fijo**, todas las subredes tienen la **misma máscara de subred**, lo que significa que cada subred tiene la misma cantidad de direcciones IP y, por lo tanto, el mismo número de hosts.
- Mantiene un desperdicio de direcciones aunque no muy grande.
- **Ejemplo:**
    - **Tenemos esta red:** 192.168.1.0/24.
    - **División en subredes:** 255.255.255.128 (/25)
        - **Subred 1:** 192.168.1.0 - 192.168.1.127 (126 host útiles ya que restamos la dirección de red y la de broadcast).
        - **Subred 2:** 192.168.1.128 - 192.168.1.255 (126 host útiles ya que restamos la dirección de red y la de broadcast).
    - Todas las subredes tienen 128 direcciones (126 utilizables), independientemente de cuántos hosts necesite cada subred.

Subnetting Variable (VLSM: Variable-Length Subnet Masking)

- En el **subnetting variable**, se utilizan máscaras de subred de **longitud variable** para crear subredes de diferentes tamaños, adaptadas a la cantidad de hosts en cada segmento de la red. Esto permite optimizar el uso del espacio de direcciones IP.
- Evita que haya un desperdicio de direcciones.
- **Mecanismo general de VLSM:**

1. Calcular la máscara para la/s subred/es de mayor cantidad de hosts.
2. De las subredes que obtenemos, asignamos todas las que se puedan con el menor desperdicio posible.
3. Si quedan segmentos de red sin una subred asignada volver a hacer el paso 1.

- **Ejemplo:**
  - **Tenemos esta red:** 192.168.1.0/24.
  - Se necesita dividir esa red en 4 subredes con los siguientes requisitos de host:
    - **Subred 1:** 50 hosts.
    - **Subred 2:** 20 hosts.
    - **Subred 3:** 10 hosts.
    - **Subred 4:** 5 hosts.
  - **Pasos:**
    1. **Calculamos la máscara necesaria para cada subred:**
       - **Subred 1 (50 host):**
         - $2^h - 2 >= 50 \rightarrow h >= log2(52) \rightarrow h >= 6$
         - **Máscara:** $24 + (8 - 6) = 26 \rightarrow 255.255.255.192$
         - **Rango:** 192.168.1.0 - 192.168.1.63 (62 hosts útiles).
       - **Subred 2 (20 host):**
         - $2^h - 2 >= 20 \rightarrow h >= log2(22) \rightarrow h >= 5$
         - **Máscara:** $24 + (8 - 5) = 27 \rightarrow 255.255.255.224$
         - **Rango:** 192.168.1.64 - 192.168.1.95 (30 hosts útiles).
       - **Subred 3 (10 host):**
         - $2^h - 2 >= 10 \rightarrow h >= log2(12) \rightarrow h >= 4$
         - **Máscara:** $24 + (8 - 4) = 28 \rightarrow 255.255.255.240$
         - **Rango:** 192.168.1.96 - 192.168.1.111 (14 hosts útiles).
       - **Subred 4 (5 host):**
         - $2^h - 2 >= 5 \rightarrow h >= log2(7) \rightarrow h >= 3$
         - **Máscara:** $24 + (8 - 3) = 29 \rightarrow 255.255.255.248$
         - **Rango:** 192.168.1.112 - 192.168.1.119 (6 hosts útiles).
    2. **Asignar las subredes:**
       - **Subred 1:**
         - **Dirección de red:** 192.168.1.0
         - **Dirección de broadcast:** 192.168.1.63
       - **Subred 2:**
         - **Dirección de red:** 192.168.1.64
         - **Dirección de broadcast:** 192.168.1.95
       - **Subred 3:**
         - **Dirección de red:** 192.168.1.96
         - **Dirección de broadcast:** 192.168.1.111
       - **Subred 4:**
         - **Dirección de red:** 192.168.1.112
         - **Dirección de broadcast:** 192.168.1.119

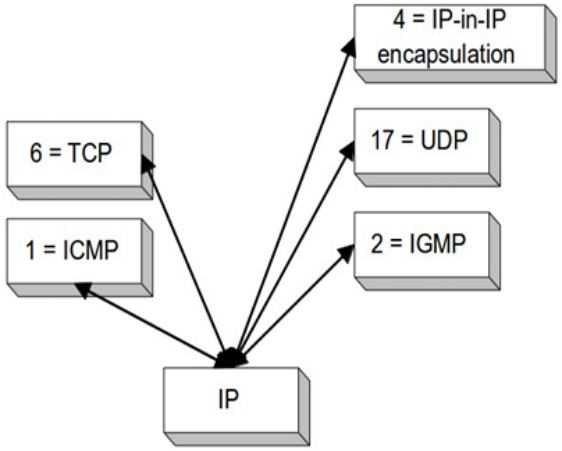
CIDR (Classless Inter Domain Routing) - Supernetting

- **CIDR (Classless Inter-Domain Routing)** es una metodología de direccionamiento IP que permite una asignación más eficiente de direcciones IP, superando las limitaciones del sistema de clases tradicional (Clase A, B y C). A su vez, este mecanismo es una estrategia para **frenar ciertos problemas** que se manifestaron con el crecimiento de Internet, estos son:
    - Las clases A y B el 50% asignadas, clases C solo el 2%.
    - Las clases C: 2^21 redes aumentarían las tablas de ruteo notablemente.
    - Crecimiento de 1988 a 2000 de tablas de ruteo.
- **Optimiza la gestión del espacio de direcciones en Internet y reduce el tamaño de las tablas de enrutamiento mediante el agrupamiento.**
- **Elimina las clases** para no limitar las redes a tamaños específicos. En su lugar, utiliza una notación de prefijo (por ejemplo, /24) para indicar la máscara de subred.
- Consiste básicamente en permitir máscaras de subred de longitud variable (VLSM) para optimizar la asignación de direcciones IP y utilizar resumen de rutas para disminuir el tamaño de las tablas de enrutamiento.

Datagrama IPv4

- Un datagrama IPv4 tiene los siguientes campos:
    - **Número de versión**: Estos **4 bits** especifican la versión del protocolo IP del datagrama. A partir del número de versión, el router puede determinar cómo interpretar el resto del datagrama IP.
    - **Longitud de la cabecera**: Puesto que un datagrama IPv4 puede contener un número variable de opciones, estos **4 bits** son necesarios para determinar dónde comienza realmente la carga útil (por ejemplo, el segmento de la capa de transporte encapsulado en este datagrama) del datagrama IP. La mayoría de los datagramas IP no contienen opciones, por lo que **el datagrama IP típico tiene una cabecera de 20 bytes.**
    - **Tipo de servicio:** Los bits del tipo de servicio (TOS, Type Of Service) se incluyeron en la cabecera de IPv4 con el fin de poder diferenciar entre los distintos tipos de datagramas IP.
    - **Longitud del datagrama:** Es la longitud total del datagrama IP (la cabecera más los datos) en bytes. Puesto que este campo tiene una longitud de 16 bits, el tamaño máximo teórico del datagrama IP es de 65.535 bytes. Sin embargo, los datagramas rara vez tienen una longitud mayor de 1.500 bytes, lo que permite que los datagramas IP quepan en el campo de carga útil de una trama Ethernet de tamaño máximo.
    - **Identificador, indicadores, desplazamiento de fragmentación:** Estos tres campos tienen que ver con lo que se denomina fragmentación IP. Los **indicadores** son 3 bits:
        - El primero es 0.
        - DF bit (Do not fragment).
        - MF bit (More fragments).

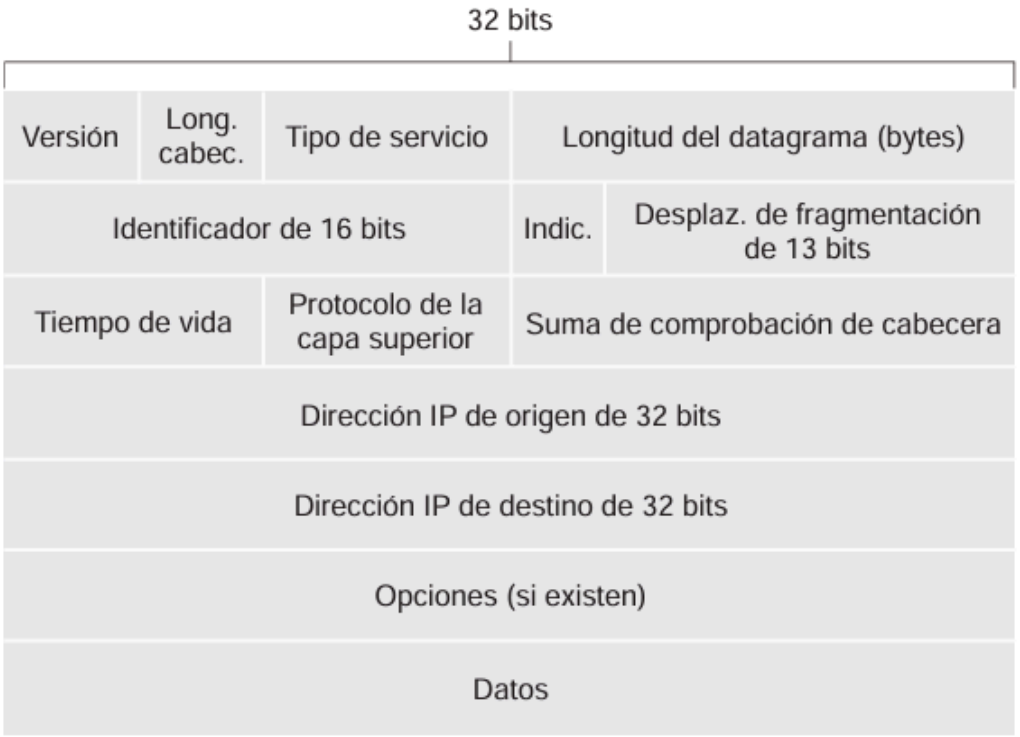
- **Tiempo de vida TTL:** Se incluye con el fin de garantizar que los datagramas no estarán eternamente en circulación a través de la red (debido, por ejemplo, a un bucle de enrutamiento de larga duración). Este campo se decrementa en una unidad cada vez que un router procesa un datagrama. Si el campo TTL alcanza el valor 0, el datagrama tiene que ser descartado por el router.
- **Protocolo:** Este campo solo se suele emplear cuando un datagrama IP alcanza su destino final. El valor de este campo indica el protocolo específico de la capa de transporte al que se pasarán los datos contenidos en ese datagrama IP y es utilizado para mux/demux. Por ejemplo, un valor de 6 indica que los datos se pasan a TCP, mientras que un valor igual a 17 indica que los datos se pasan a UDP.



- **Suma de comprobación de cabecera:** La suma de comprobación de cabecera ayuda a los routers a detectar errores de bit en un datagrama IP recibido. Esta suma de comprobación se calcula tratando cada pareja de 2 bytes de la cabecera como un número y sumando dichos números utilizando aritmética de complemento a 1. Un router calcula la suma de comprobación de cabecera para cada datagrama IP recibido y detecta una condición de error si la suma de comprobación incluida en la cabecera del datagrama no coincide con la suma de comprobación calculada. Normalmente, los routers descartan los datagramas en los que se ha detectado que existe un error.
- **Direcciones IP de origen y de destino:** Cuando un origen crea un datagrama, inserta su dirección IP en el campo de dirección IP de origen e inserta la dirección del destino final en el campo de dirección IP de destino.
- **Opciones:** El campo de opciones permite ampliar una cabecera IP. La idea original era que las opciones de cabecera rara vez se emplearan: de ahí la decisión de ahorrar recursos no incluyendo la información de los campos opcionales en la cabecera de todos los datagramas. Sin embargo, la mera existencia de opciones complica las cosas, ya que las cabeceras de datagrama pueden tener una longitud variable, por lo que no puede determinarse a priori dónde comenzará el campo de datos. Además, dado que algunos datagramas pueden requerir el procesamiento de opciones y otros no, la cantidad de tiempo necesario para procesar un datagrama IP en

un router puede variar enormemente. Estas consideraciones cobran una particular importancia en el procesamiento IP realizado en los hosts y routers de altas prestaciones. Por estas razones y otras, las opciones IP fueron eliminadas en la cabecera de IPv6.

- **Datos (carga útil):** En la mayoría de las circunstancias, el campo de datos del datagrama IP contiene el segmento de la capa de transporte (TCP o UDP) que va a entregarse al destino. Sin embargo, el campo de datos puede transportar otros tipos de datos, como por ejemplo mensajes ICMP.



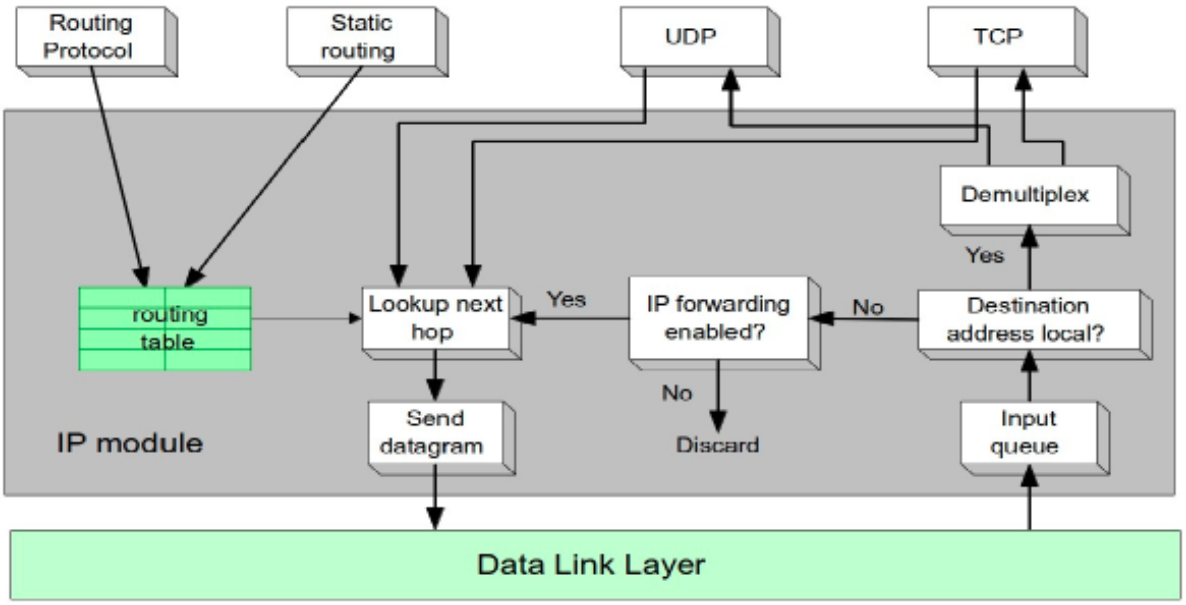
Formato del datagrama IPv4

Ruteo

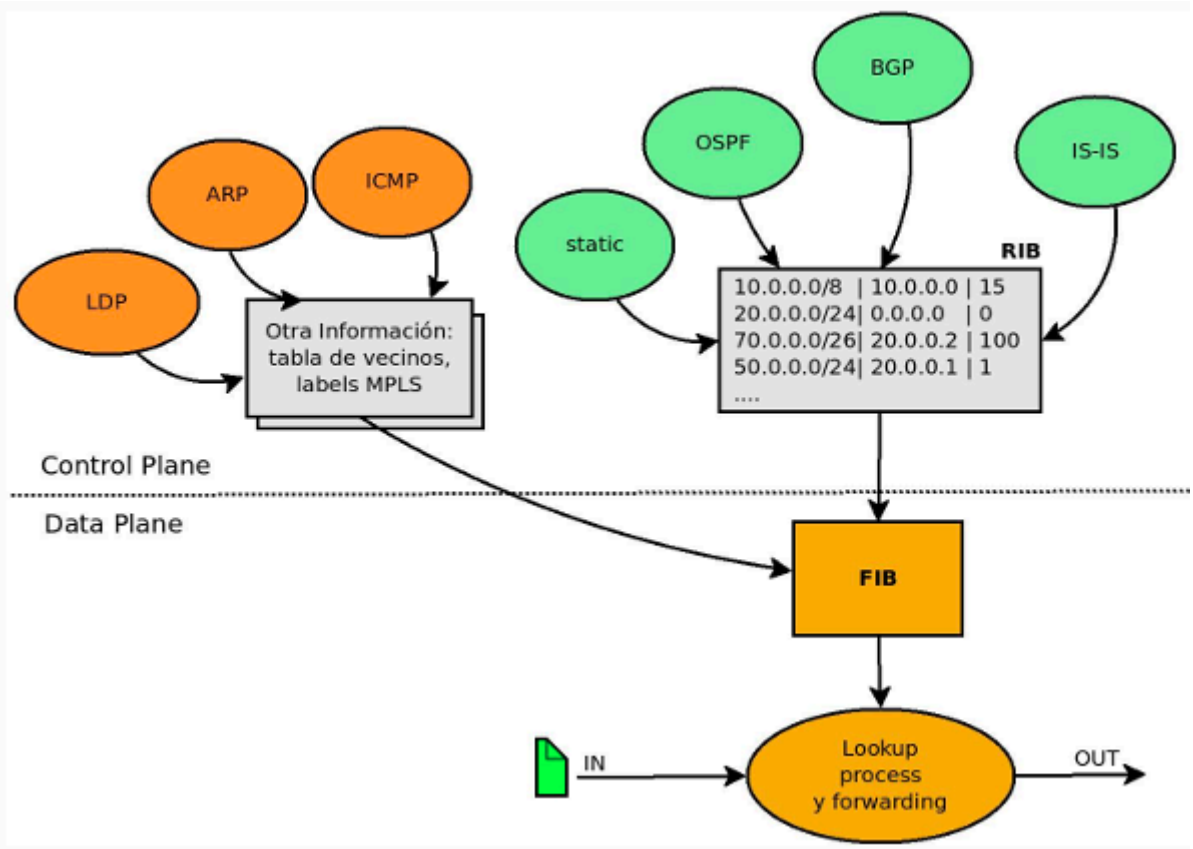
- **Tabla de ruteo/Tabla de enrutamiento:**
  - Estructura en hosts y routers (gateways) que indica cómo despachar un mensaje (cuál es el camino que tiene que seguir ese paquete cuando va a un destino dado). Perspectiva del vecino, siguiente salto.
    - No brinda todo el camino a recorrer, sólo indica cuál es el mejor salto que puede dar el paquete desde donde está parado.
- **Host:**
  - No despacha mensajes que recibe que no son para él. Despacha sólo sus mensajes mirando su tabla de ruteo.
  - Pueden participar de forma pasiva en el routing.
- **Router:**
  - Nodos intermedios, más de una interfaz, despacha mensajes mirando su tabla de ruteo, desde cualquier interfaz.

    - Pueden participar de forma activa en el routing: reciben, generan y propagan información.
- **Host multihome:**
    - Tiene varias interfaces, no rutea.
- **Ruteo:**
    - El ruteo es el proceso de determinar el camino o la ruta que debe seguir un paquete de datos desde su origen hasta su destino a través de una red.
    - Su función principal es seleccionar la interfaz de salida y el próximo salto. Lo implementan los Routers y los Hosts.
    - Es de control, es decir, no maneja directamente el tráfico de datos, sino que toma decisiones sobre cómo se debería enrutar el tráfico.
    - Ocurre en routers y, en menor medida, en hosts.
    - Utiliza la **RIB (Routing Information Base)** o tabla de enrutamiento para almacenar los resultados del proceso de ruteo, esta tabla es alimentada a partir del uso de **protocolos de enrutamiento** (RIP, OSPF, BGP, etc) que construyen y actualizan la RIB.
    - **Tipos de Ruteo:**
        - **Ruteo Estático:**
            - Las rutas se configuran manualmente por el administrador.
            - **Ventajas:**
                - Fácil de implementar en redes pequeñas.
                - No consume recursos adicionales en el router.
                - Ofrece mayor control y seguridad.
            - **Desventajas:**
                - No se adapta automáticamente a cambios en la red.
                - No es escalable ni tolerante a fallos.
        - **Ruteo Dinámico:**
            - Utiliza protocolos de enrutamiento para aprender y actualizar rutas automáticamente aunque requiere una configuración inicial por el administrador.
            - **Ventajas:**
                - Se adapta automáticamente a cambios en la topología.
                - Escalable y tolerante a fallos.
                - Facilita la gestión en redes grandes o complejas.
            - **Desventajas:**
                - Requiere más recursos de procesamiento.
                - La configuración inicial es más compleja.
- **Protocolos de Enrutamiento:**
    - **Clasificación según el ámbito de operación:**
        - **IGP (Interior Gateway Protocols):** Dentro de un sistema autónomo (AS). Ejemplos: OSPF, EIGRP, RIP.
        - **EGP (Exterior Gateway Protocols):** Entre diferentes sistemas autónomos. Ejemplo: BGP.
    - **Clasificación según el método de operación:**
        - **Vector de Distancia (DV):** Calculan la ruta basada en la distancia hacia la red destino. Ejemplo: RIP.

        - **Estado de Enlace (Link State):** Cada router tiene una vista completa de la topología de la red. Ejemplo: OSPF.
        - **Vector de Camino (PV):** Similar a DV, pero incluye información del camino. Ejemplo: BGP.
        - **Híbridos:** Combinan características de DV y Link State. Ejemplo: EIGRP.
- **Routing Domain:**
    - Conjunto de routers que comparten el mismo protocolo de ruteo. Un **AS** puede contener uno o más **Routing Domains**.
- **Sistema Autónomo (AS- Autonomous System):**
    - Conjunto de redes bajo una misma administración y política de ruteo.
    - Cada AS tiene un número único llamado **ASN (Autonomous System Number)**.
- **Forwarding/Despacho:**
    - El forwarding es el proceso de mover un paquete de datos desde la interfaz de entrada de un router hasta la interfaz de salida, según la información contenida en la tabla de enrutamiento.
    - Su función principal es pasar el paquete desde una interfaz de entrada hacia una interfaz de salida. Ocurre solo en routers.
    - Es más intensivo.
    - Es de datos, envía protocolos enrutados (routed).
    - Los routers tienen el forwarding habilitado, los hosts no.
    - Utiliza la información de la RIB para construir una versión optimizada llamada **FIB (Forwarding Information Base / Forwarding Table)** que sirve para tomar decisiones rápidas y reenviar paquetes de forma eficiente.



Función de Ruteo



Routing y Forwarding

Tabla de Ruteo

- **Estructura:**
  - Red Destino (Net/Mask).
  - Next Hop (Próximo salto).
  - Interfaz de salida.
- En un **Host** está estructura es más simple.

```
andres@h1:~$ netstat -nr
Destination     Gateway         Genmask          Metric Iface
193.168.4.224   0.0.0.0         255.255.255.224  0      e0
193.168.4.128   193.168.4.225   255.255.255.192  2      e0
0.0.0.0         193.168.4.225   0.0.0.0          -      e0
```

Estructura de la tabla de ruteo

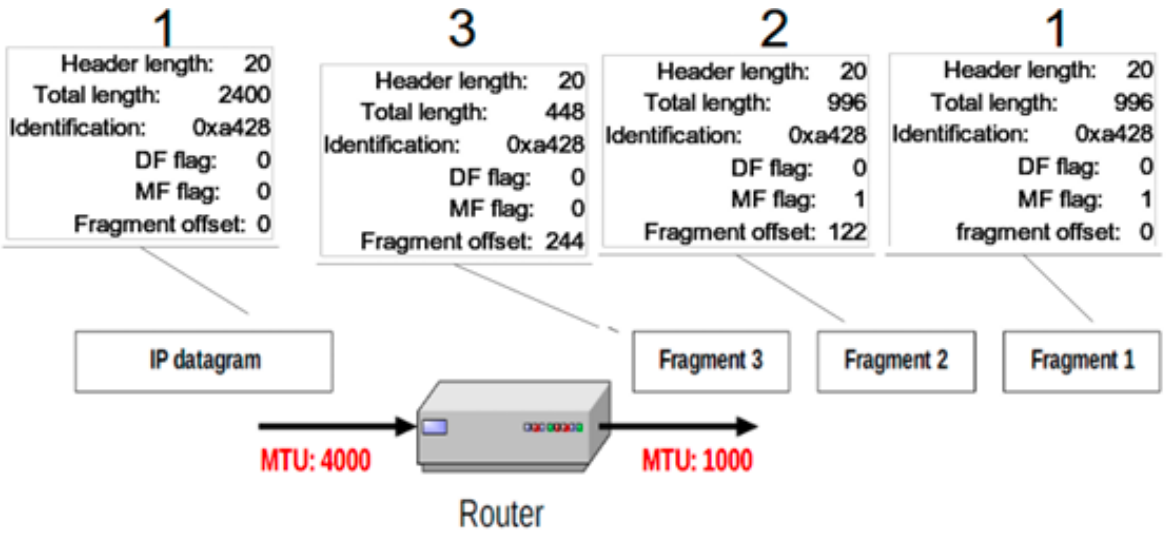
```
andres@r1:~$ netstat -nr
Destination     Gateway         Genmask          Metric Iface
193.168.4.224   0.0.0.0         255.255.255.224   0      e1
193.168.4.192   0.0.0.0         255.255.255.224   0      e0
200.3.4.0       0.0.0.0         255.255.255.252   0    ppp0
193.168.4.0     193.168.4.194   255.255.255.0     0      e0
0.0.0.0         200.3.4.1       0.0.0.0           -    ppp0
```

Alternativa de una tabla con uso de CIDR

- **Destination**
    - Red destino.
    - El Destination 0.0.0.0 representa la ruta por defecto. A esta ruta se dirigen los paquetes que no coincidan con ninguna red destino dentro de la tabla.
- **Gateway**
    - Representa la IP del próximo salto.
    - Si el Gateway es 0.0.0.0 eso quiere decir que se trata de una red directamente conectada.
- **Genmask**
    - Máscara.
- **Metric**
    - Se utiliza para decidir cuál es la mejor ruta, puede ser por ejemplo la cantidad de saltos que se tienen que hacer para llegar al destino, la capacidad de enlace, el ancho de banda.
- **Iface**
    - Interfaz de salida.

Tareas de Ruteo

- **Esto es lo que pasa cuando un paquete llega a un Router:**
    - Validación de datagrama: IP header.
    - Calcula checksum (solo header).
    - Leer IP destino.
    - Buscar en tabla de ruteo, seleccionar prefijo más largo (“best match”).
    - Decrementar TTL.
    - Fragmentar (alternativo).
        - Debido a que hay diferentes capas de enlaces con diferentes MTUs.
        - **Si la capacidad del enlace es menor que el tamaño del paquete, hay que fragmentar.**
        - Fragmentos múltiplos de 8 bytes.
            - Offset en unidades de 8 bytes.
        - Se deben agregar los headers necesarios al datagrama IP.



Ejemplo de Fragmentación

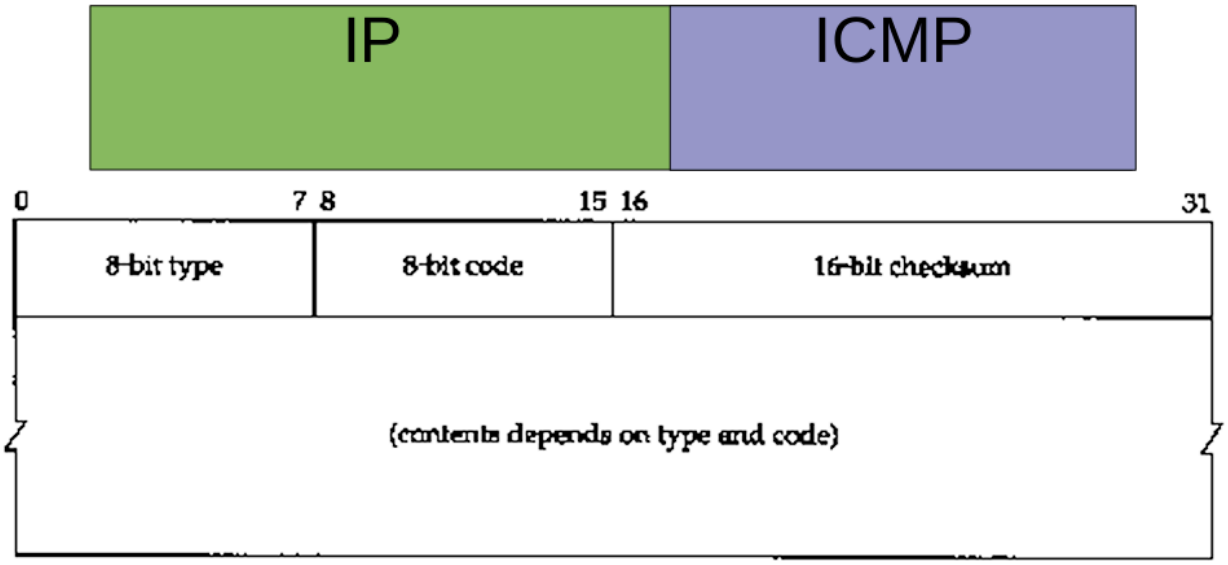
- Transmitir o Descartar.
- Generar ICMP (alternativo).

ICMP (Internet Control Message Protocol)

- Protocolo auxiliar (Helper) que funciona en conjunto con el Protocolo de Internet (IP) compensando la falta de mecanismos de control de errores en IP.
- No es un protocolo de transporte ya que no fue concebido para llevar datos de usuario, es de la capa 3 (Red).
- IP carece de control, el mismo es dado por un protocolo auxiliar.
- Se encapsula en IP.
- ICMP no agrega confiabilidad a IP. En cambio, proporciona una forma para que los hosts y enrutadores intercambien información vital sobre la red, principalmente informes de error y diagnósticos a modo de feedback.
- Podría ser prescindible en IPv4.
- **Funciones clave:**
  - **Informes de Error:** ICMP informa a los hosts sobre varios errores encontrados durante el procesamiento de datagramas IP.
  - Diagnóstico de Red: ICMP proporciona herramientas para evaluar la conectividad y el rendimiento de la red.
  -

Mensajes ICMP

- **Tienen la siguiente estructura:**
  - **Tipo:** Identifica el tipo de mensaje ICMP.
  - **Código:** Proporciona información más específica sobre el tipo de mensaje.
  - **Cabecera y Primeros 8 Bytes del Datagrama IP:** Esto permite al host de origen identificar el datagrama IP que causó el error.



Formato

Tipos de Mensajes ICMP

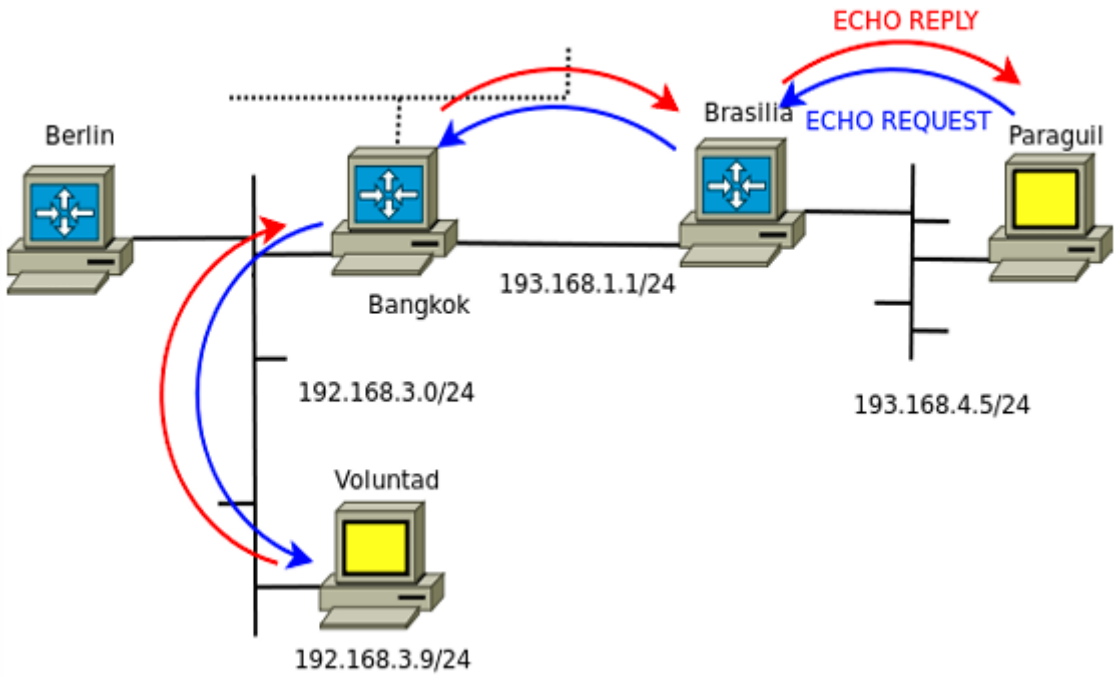
Tipo ICMP	Código	Descripción
0	0	respuesta de eco (para ping)
3	0	red de destino inalcanzable
3	1	host de destino inalcanzable
3	2	protocolo de destino inalcanzable
3	3	puerto de destino inalcanzable
3	6	red de destino desconocida
3	7	host de destino desconocido
4	0	regulación del origen (control de congestión)
8	0	solicitud de eco
9	0	anuncio de router
10	0	descubrimiento de router
11	0	TTL caducado
12	0	Cabecera IP errónea

Tipos de mensajes ICMP

Echo Request/Echo Reply (PING)

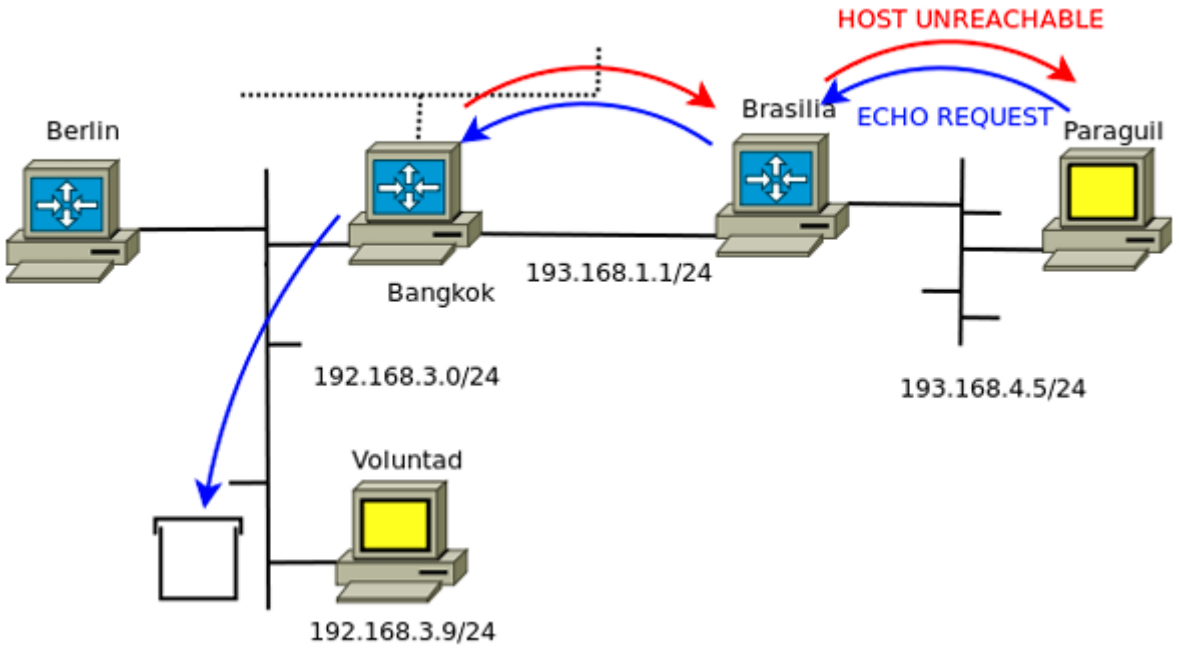
- Pensado para probar conectividad IP entre dos hosts.
- Sirve para medir el RTT min/avg/max/dev y loss, de esta forma poder diagnosticar problemas.

- El comando **ping** utiliza estos mensajes para medir la latencia y verificar la conectividad.
- **Funcionamiento:**
    - **Echo Request:** El host emisor envía un paquete ICMP de tipo 8 al destino con código 0. Este paquete contiene un mensaje de solicitud de eco. El código 0 especifica que es una solicitud estándar sin código específico.
    - **Echo Reply:** El host destino responde con un paquete ICMP de tipo 0 al emisor con código 0. El código 0 especifica que es una respuesta estándar sin código específico.
- Si un nodo recibe un ICMP **Echo Request**, debe responder copiando el contenido con un **Echo Reply (PONG).**



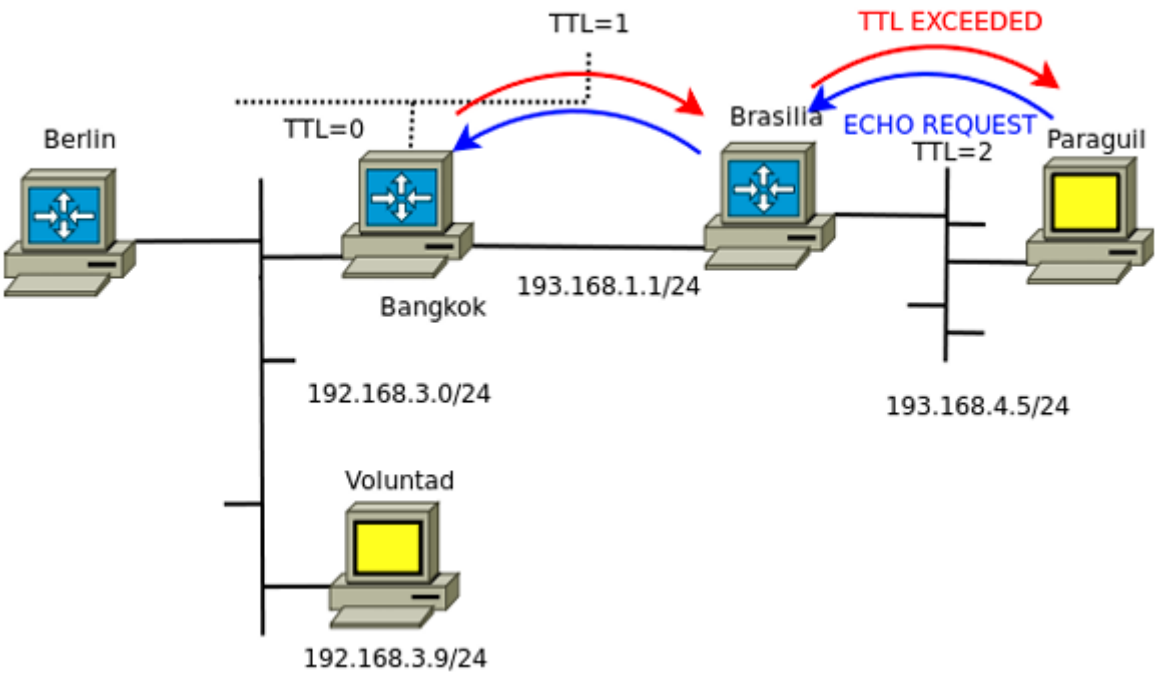
ICMP Destino Inalcanzable

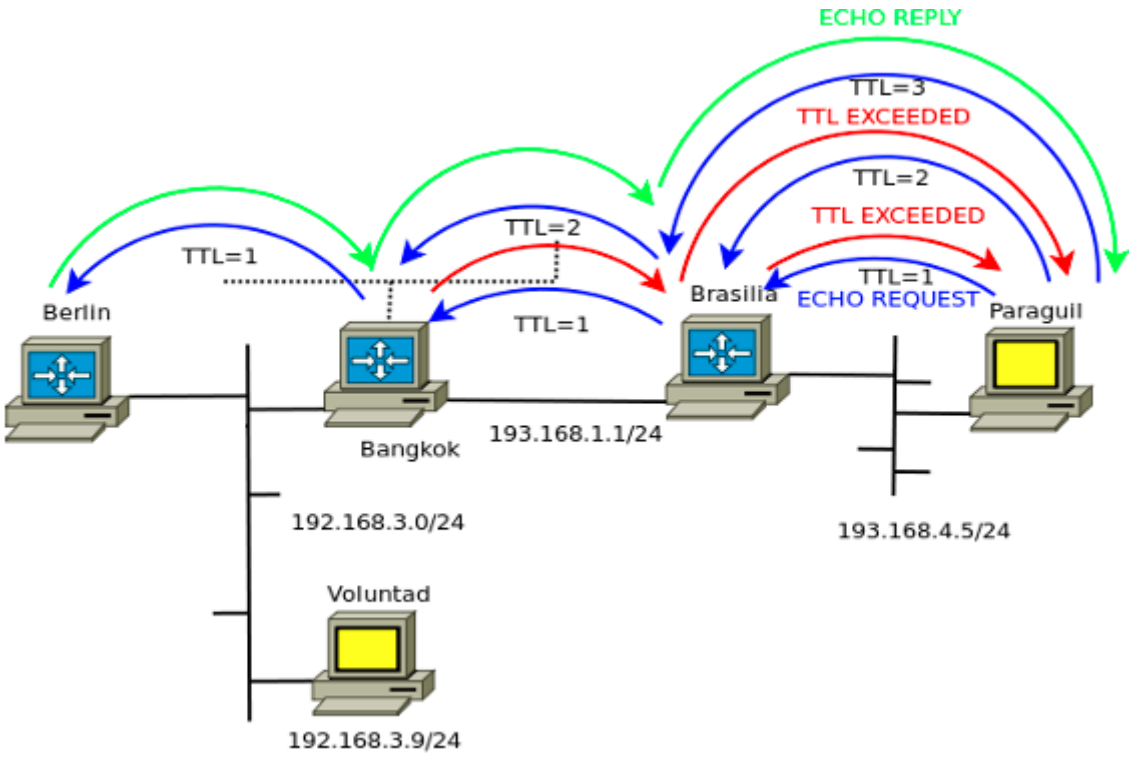
- Se genera cuando un paquete no puede ser entregado a su destino. Este mensaje tiene varias subcategorías, dependiendo de la causa del fallo.
- **Tipos de error común dentro de "Destino Inalcanzable":**
    - **Red inalcanzable (Type 3, Code 0):** La red especificada no puede ser alcanzada, es decir, el router no tiene una ruta en la tabla de ruteo a esta red.
    - **Host inalcanzable (Type 3, Code 1):** El host de destino no es accesible.
    - **Protocolo inalcanzable (Type 3, Code 2):** El protocolo especificado no es compatible.
    - **Puerto inalcanzable (Type 3, Code 3):** No hay ningún proceso UDP escuchando en el puerto destino.
    - **Destino inalcanzable debido a filtrado de comunicación (Type 3, Code 13):** El tráfico está bloqueado por un firewall o ACL.



ICMP TTL Expirado

- Se envía cuando el **TTL (Time To Live)** de un paquete alcanza cero antes de llegar a su destino, es decir, el hop count con el cual salió el mensaje ha expirado. Esto evita que los paquetes circulen indefinidamente en la red.
- El TTL se puede exceder en viaje o en re-ensamblado.
- El TTL puede tener un valor máximo de 255.
- **Funcionamiento:**
    - Cada vez que un paquete pasa por un router, su TTL se reduce en 1.
    - Si el TTL llega a 0 antes de alcanzar el destino, el router descarta el paquete y envía un mensaje ICMP **Type 11** al emisor.
- Es utilizado por la herramienta **traceroute** para determinar la ruta que sigue un paquete a través de una red.





ICMP Route Redirect

- Se utiliza para informar a un host que debe enviar sus paquetes a través de una ruta diferente para llegar a una red específica.
- **Funcionamiento:**

    - Cuando un router recibe un paquete destinado a una red y encuentra que hay una ruta más directa, envía un mensaje ICMP **Type 5** al emisor con la ruta correcta.
    - El emisor actualiza su tabla de enrutamiento para futuros envíos.

ICMP Source Quench (Control de Congestión)

- Este mensaje fue diseñado para indicar a un host emisor que reduzca la tasa de envío de paquetes debido a congestión en la red.
- **Funcionamiento:**
    - El router o el destino envía un mensaje **Type 4** al emisor, solicitando que disminuya la velocidad de transmisión.
- Este tipo de mensaje está **obsoleto** en las redes modernas, ya que han sido reemplazados por mecanismos más eficientes como TCP control de congestión.

ICMP Address Mask

- Permite a un host solicitar la máscara de subred utilizada en una red local.
- **Funcionamiento:**
    - **Address Mask Request (Type 17):** Un host envía este mensaje para obtener la máscara de subred.
    - **Address Mask Reply (Type 18):** El router responde con la máscara de subred adecuada.

ICMP Timestamp

- Proporciona la hora del sistema del host emisor y puede sincronizar el tiempo entre dispositivos.
- **Funcionamiento:**
    - **Timestamp Request (Type 13):** Se envía para solicitar la hora actual del sistema.
    - **Timestamp Reply (Type 14):** El host receptor devuelve la hora actual.

DHCP (Dynamic Host Configuration Protocol)

- **Un host para conectarse a una red IP requiere 3 parámetros + 1**.
- **Para conectarse a una Red local:**
    - **Dirección IP:** Una dirección IP única que identifica al host en la red.
    - **Máscara de red:** Define la subred a la que pertenece el host y ayuda a determinar qué direcciones están en la red local y cuáles están en redes externas.
- **Para conectarse a otras redes:**
    - **Router por default (Default Gateway):** La dirección IP del router que el host usa para enviar tráfico a redes externas.
- **Para usar servicios:**
    - **Servidor(es) de DNS:** Las direcciones IP de los servidores DNS que el host utiliza para resolver nombres de dominio a direcciones IP.

- **Forma de obtener esos parámetros:**
    - **De forma estática:**
        - Configuración manual.
        - Difícil de mantener.
        - No escalable.
        - No sirve para movilidad
    - **De forma dinámica:**
        - RARP.
        - ICMP.
        - BOOTP.
        - DHCP .
- DHCP es un protocolo de red fundamental (Helper de Ip) que permite a los dispositivos en una red obtener automáticamente la información de configuración necesaria para comunicarse en la red.
- Al estar montado sobre UDP se lo suele considerar protocolo de nivel de aplicación.
- Sirve tanto para IPv4 como para IPv6.
- Cuando los hosts arrancan solo tienen acceso a su red local de forma broadcast.
- En la red local existe un o más servidor de autoconfiguración:
    - DHCP servers.
- Los hosts sin parámetros de red envían requerimiento.
- Los servidores los atienden asignando los valores que brindan conectividad.
- El parámetro se reserva por un tiempo.

DHCP Mensajes

- **Algunos Mensajes DHCP:**
    - Discover.
    - Offer.
    - Request.
    - ACK.
    - Release.
    - NAK.
- **Montado sobre UDP:**
    - Bootpc (client) 68.
    - Bootps (server) 67.

DHCP Mensajes Broadcast

- **Broadcast:**
    - Discover.
    - Request.
- **Unicast/Broadcast:**
    - Offer.
- En general se envía unicast, pero debido a que pueden existir equipos que no procesan mensajes unicast antes de tener configurada la dirección IP completa, se podrían enviar en forma broadcast.

NAT (Network Address Translation)

- Traslación de direcciones de un espacio privado (no “enrutable” en Internet) a un espacio público.
- Los procesos de traslación se realizan sobre redes stubs (solo una salida) y se deben mantener tablas de traslaciones.
- Existen varias formas de realizarlo:

NAT Básico

- Una forma de realizarlo es “one-to-one”.
- Se mapea una dirección IPv4 privada a una dirección IPv4 pública.
- Permite acceso en ambas direcciones.
- **Forma Estática:**
    - Requiere tantas direcciones públicas como privadas.
- **Forma Dinámica:**
    - No requiere tantas direcciones públicas como privadas, pero sí requiere un timer por cada entrada.
    - Limita el acceso simultáneo de acuerdo al pool pub.

NAPT (Network Address Port Translation)

- NAT no es implementable cuando se tiene un **pool chico** de direcciones o no se poseen direcciones públicas asignadas.
    - En ese caso se debe trabajar con campos de la capa de transporte o del payload.
- NAPT es conocido como **PAT (Port Address Translation): “one-to-many”.**
- Se utilizan los **puertos** de los protocolos u otros valores como **ICMP Identifier** para resolver el mapeo.
- Se pueden usar timers y sesión del protocolo.
- En la tabla de traslaciones se mantienen el **protocolo y los puertos origen y destino.**
- Se intenta conservar el puerto origen, pero si está “ocupado” se debe reemplazar por otro. Básicamente si hay otro que genera mismo puerto y destino y no hay mas direcciones, se cambia el puerto origen (por uno que no está ocupado)
- El dispositivo debe “violar” los límites impuestos por la división en capas.
- **Tiene 2 alternativas:**
    - **Dinámico sobre pool:** utiliza un pool y hace PAT sobre este (habitual en nuestras casas).
    - **Dinámico sobre dirección overload/masquerade:** utiliza la dirección IP externa y haciendo overloading/masquerading sobre esta.

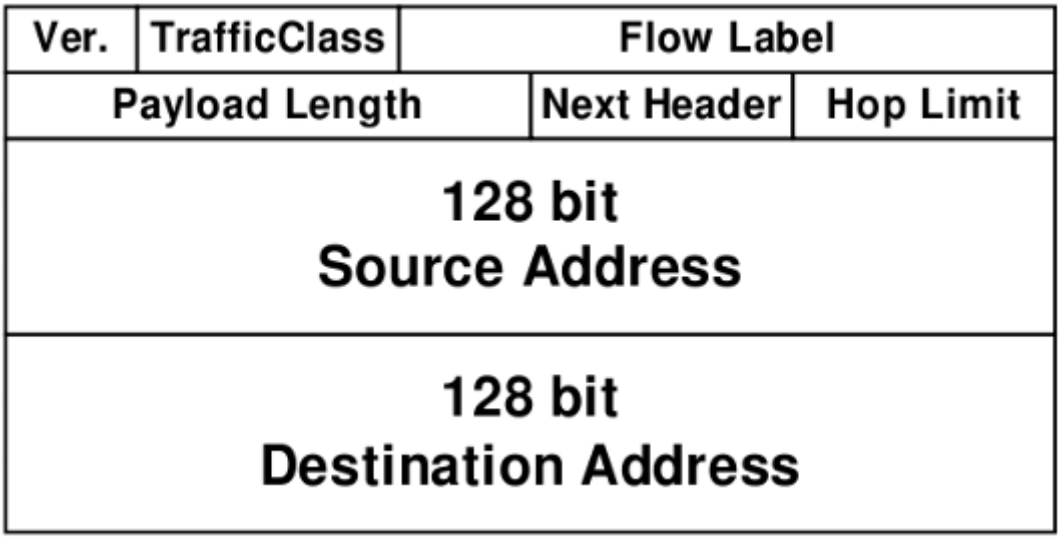
Port Forwarding

- Overloading/Masq no permiten acceso desde “afuera” hacia “adentro”.
- Solo se permite entrar tráfico de conexiones generadas internamente.

- Permite poder tener servicios en una red privada accesibles desde “afuera”.
- No se requiere NAT estático, se implementa con NAPT y mapeo reverso estático de puertos.
- Se configura a mano.

IPv6 (Internet Protocol version 6)

- **Brinda un mayor espacio de direcciones (128 bits).**
- Menor overhead de procesamiento.
- Ordenar las tablas de enrutamiento
- Conectar todo, usar autoconfiguración de direcciones
- Arquitectura de red jerárquica para un ruteo eficiente.
- Seguridad a nivel IP (IPSec obligatorio).
- Jumbogramas, size(datagrama) > 64KB.
- Movilidad y más direcciones de multicast.
- No puedo desactivar ICMP.
- Datagramas de 40 bytes.
- **Se simplifica la cabecera:**
    - Se saca la fragmentación, se deja solo de extremo a extremo como opción.
    - Se saca el checksum de cabecera.
    - Header de tamaño fijo. No existen más las Opciones.
    - **Flow Label:** identificador de flujo (20 bits). Si hay mensajes que son de una misma conexión debo meterme a transporte, Flow Label se usaba para eso pero no se terminó usando.
    - Se renombran los campos: Traffic Class (TOS, permite tratar paquetes de forma diferenciada), Hop Limit (TTL), Next Header (Protocol)
    - **Cabeceras de extensión:**
        - Permite la extensibilidad del protocolo.
        - Se encuentran a continuación del header.
        - En general, son procesadas por los extremos.



Formato de cabecera IPv6

Comparación entre cabeceras de IPv4 e IPv6

Funcionalidades

- Direccionamiento.
- Ruteo/Forwarding.
- Generalidades de IPv6
- Mux/Demux de protocolos superiores.
- Otras: como evitar loops.
- Descubrimiento de Vecinos (NDP):
    - ND propiamente
    - Router discovery y autoconfiguración
- Manejo de Grupos de Multicast.

Direcciones IPv6

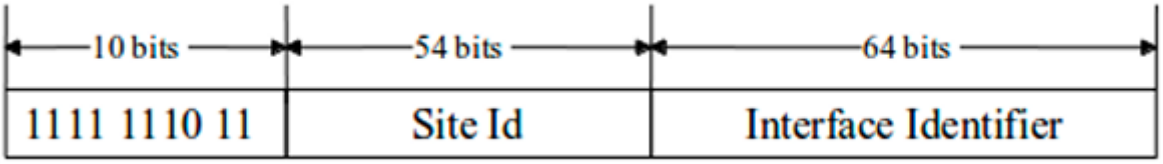
- Son de 128 bits.
- Se anotan en hexadecimal en grupos de 16 bits, separadas por ":".
- Los ceros al inicio de cada grupo se pueden obviar.
- Ceros contiguos se puede eliminar con "::". Sólo se puede utilizar una vez
- Se utilizan "[","]"para indicar port en URL: http://[2001:db8:1011:1:0:0:0:1]:8080
- No se usa máscara, solo prefix length.

Tipos de direcciones:

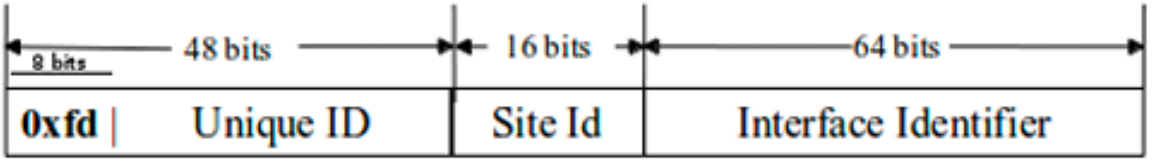
Unicast

- **Clasificadas por Alcance:**
  - **Locales (Link-local)**
    - **Alcance:** solo red directamente conectada
    - **Prefijo Asignado:** FE80::/10.
    - **Prefijo Utilizado:** FE80::/64 (len. en LAN /64)
    - IID se usan direcciones del hardware
      - De forma manual.
      - Se generan con el prefijo link-local y realiza DAD (Duplicate Address Detection).

  - **De sitio site-local (desaconsejadas)**
    - **Prefijo:** FEC0::/10
    - **Alcance:** sitio u organización. Similar a las redes privadas de IPv4.
    - Dificultad de establecer los límites.



  - **De Sitio Únicas**
    - **Prefijo:** FC00::/7, dividido en FC00::/8 y FD00::/8.
    - **Prefijo Utilizado:** FD00::/8, [xxxxxxxL] L bit = 1 (def. local).
    - **Alcance:** sitio u organización.
    - Reemplazan las direcciones de Site Local. Unique ID generado de forma pseudoaleatoria



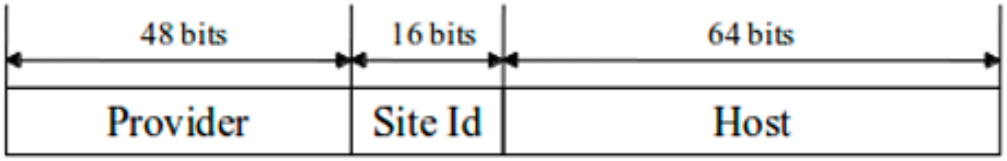
  - **Compatibilidad ipv4-compat (desaconsejadas)**
    - Usadas para la transición.
    - Asigna a un IPv4 global única una IPv6.

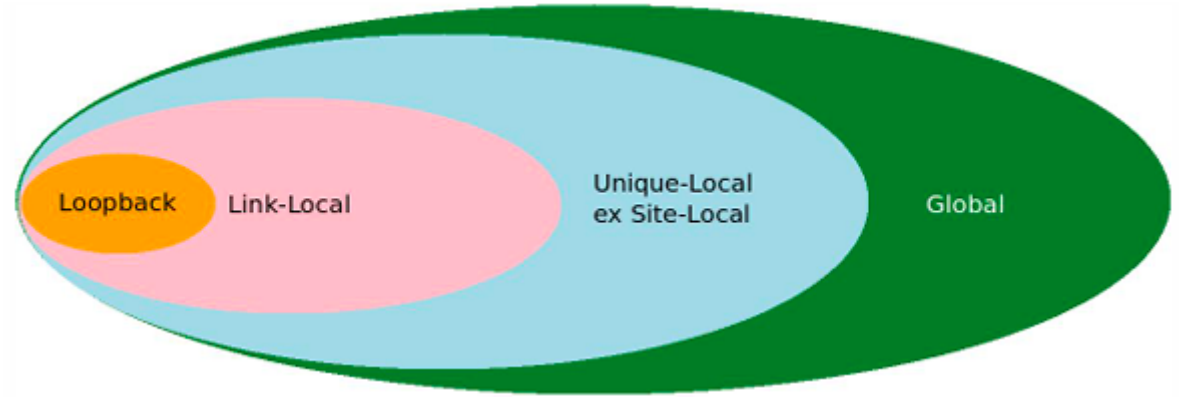
IPv4-Compatible Address = 0:0:0:0:0:0:192.168.30.1
= ::192.168.30.1
= ::C0A8:1E01

- **IPv4-mapped IPv6**

IPv4-Mapped Address = 0:0:0:0:0:FFFF:192.168.30.1

- **Globales**
  - **Prefijo:** cedidos por un provider.
  - **Alcance:** Internet. Similar a las direcciones públicas de IPv4.
  - La parte del host se puede generar como uno quiera pero debe ser única.
  - Siempre se deja 64 bits para el host.
  - El proveedor puede usar menos de 48 bits lo que significa que queda más espacio para subredes.

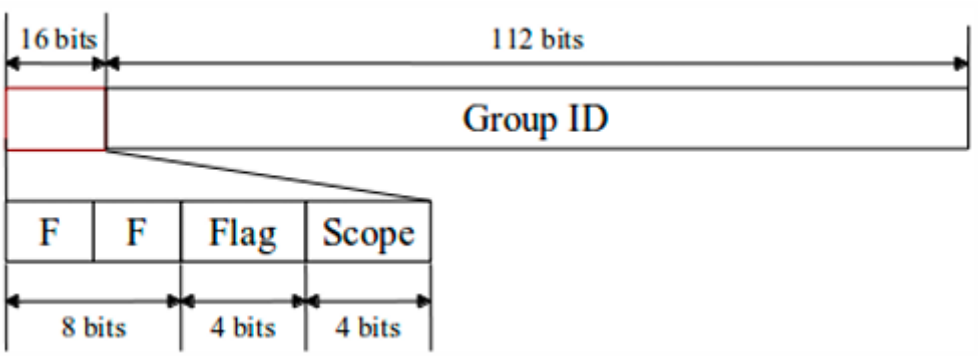




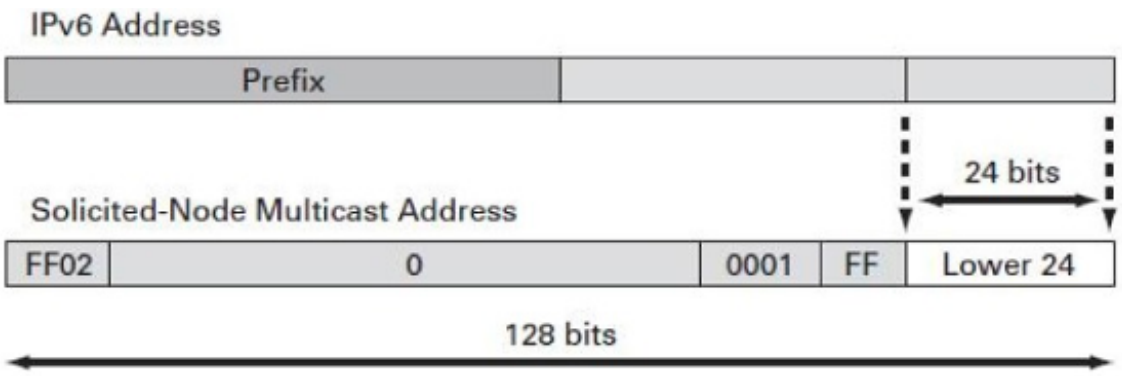
Direcciones IPv6 Unicast

Multicast

- **Prefijo:** FF00::/8
- **Flags:** permanente, temporaria. Otros reservados.
- **Alcance:** 1: nodo local, 2: link local, 5: site local, 8: org. local, E: global.
- **GID:** grupo de multicast.



- **Solicited Node Multicast Address (SD)**
  - Usada para ND (Neighbor Discovery) en lugar de flooding en la LAN.
  - Generada a partir de unicast/anycast.
  - Por cada unicast/anycast debe hacer join de la multicast.



Anycast (Tomadas del rango Unicast)

Direcciones Especiales

- **Any (sin especificar):**
  - ::0/0
- **Loopback/Localhost:**
  - ::1/128
- **Documentación:**
  - 2001:db8::/32
- **6Bone:**
  - 3FFE::/16, devueltas al IANA en 2006.

Ruteo

- Se hace uso de RIB

```
root@n7:/# ip -6 route show
2001:db8:1234:3::/64            dev eth0  proto kernel  metric 256
fe80::/64                       dev eth0  proto kernel  metric 256
default via 2001:db8:1234:3::1  dev eth0  metric 1024
default via fe80::200:ff:feaa:5 dev eth0  proto kernel ... expires 24sec
...

root@n7:/# netstat -nr -A inet6
Kernel IPv6 routing table
Destination           Next Hop             Flag   Met Ref   Use If
2001:db8:1234:3::/64  ::                   U      256 0     1   eth0
fe80::/64             ::                   U      256 0     0   eth0
::/0                  2001:db8:1234:3::1   UG    1024 0     0   eth0
::/0                  fe80::200:ff:feaa:5  UGDAe 1024 0     0   eth0
::1/128               ::                   Un       0 1     1   lo
...
```

- Ruteo Estático.
- RIP-ng.
- OSPFv3.
- IS-IS.
- MP-BGP.
- ...