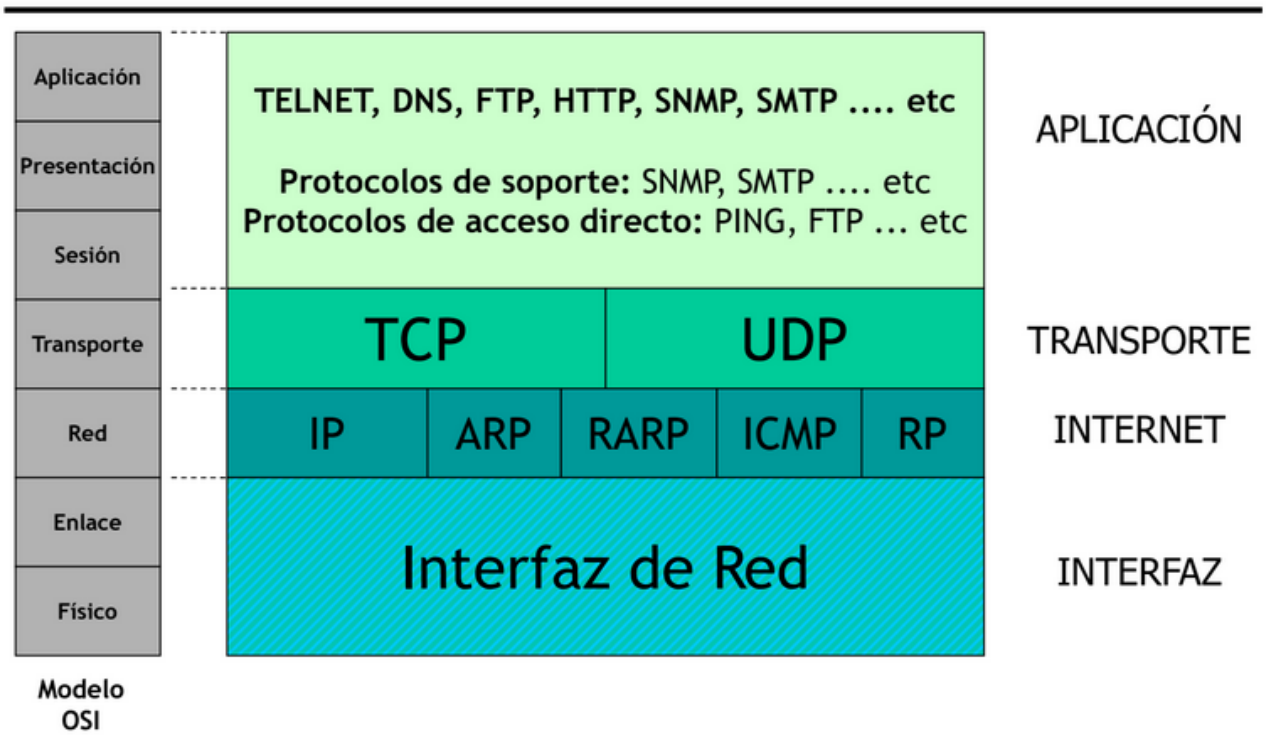
SUITE TCP/IP

Suite transmission control protocol / internet protocol.

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

Arquitectura TCP/IP



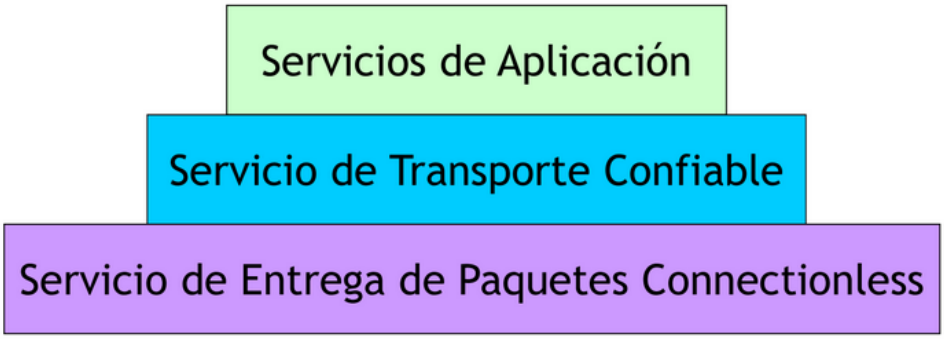
TCP/IP no especifica la interfaz de red, es decir no especifica sobre que protocolo de enlace debe ser encapsulado sino que esta abierto a ser encapsulado en cualquier protocolo de capa dos.

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

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

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

Filosofía



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

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

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

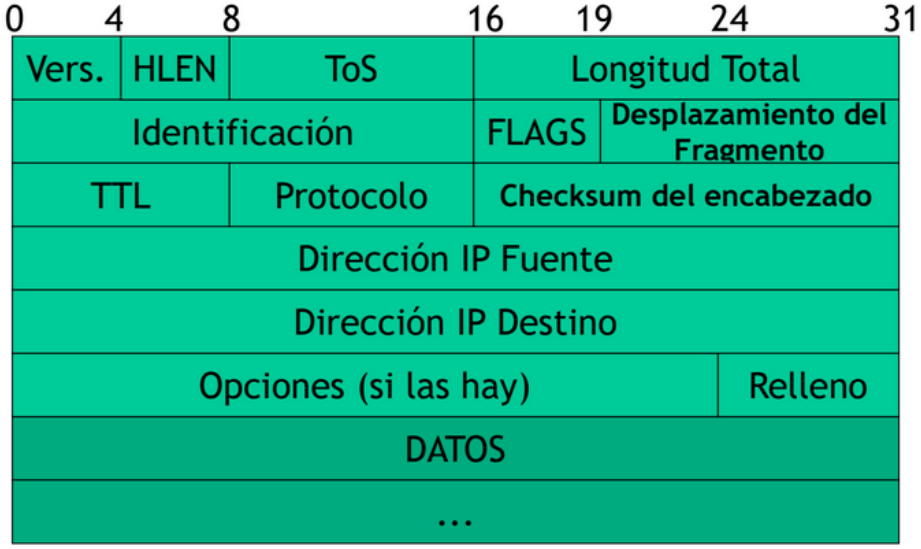
IP = Servicio connectionless

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

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

Datagrama IP

Datagrama **es la cabecera del protocolo mas los datos.**



Versión (4bits)

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

Tamaño de cabecera (HLEN. 4Bits)

Longitud de la cabecera, en **palabras de 32 bits**. Se utiliza debido a que **los últimos campos de la cabecera son de longitud variable**, por lo que el origen debe poner la longitud total de la cabecera. Su **valor mínimo y común es de 5 palabras** ( ) **para una cabecera correcta, y el máximo de 15 palabras** (**15x32** = 480 bits, 60 bytes).

Tipo de servicio (ToS, 8bits)

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

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

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

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

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

Valores recomendados para el ToS según RFC1349:

APLICACION	Minimize Delay	Maximize throughput	Maximize Reliability	Minimize Cost
Telnet / Rlogin	1	0	0	0
FTP - Control	1	0	0	0
DNS Query (UDP)	1	0	0	0
FTP - Data	0	1	0	0
ICMP	0	0	0	0

Longitud total (16bits)

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

Identificación (16bits)

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

Flags (3bits)

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

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

Desplazamiento del fragmento (13bits)

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

Tiempo de vida (TTL, 8bits)

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

Protocolo (8bits)

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

Suma de Control de Cabecera (16bits)

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

Dirección IP de origen (32bits)

Debe ser dada en formato de red.

Dirección IP de destino (32bits)

Debe ser dada en formato de red.

Opciones (bits variables)

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

Relleno (bits variables)

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

Datos (bits variables)

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

Fragmentación y reensamblado

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

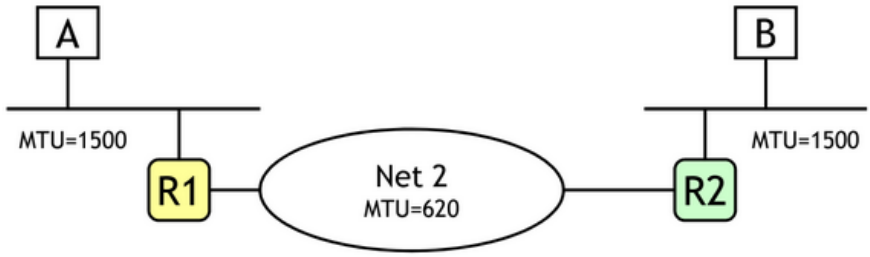
MTU: Cada tecnología de conmutación de paquetes, fija un **límite máximo para la cantidad de datos que pueden transmitirse en una única trama.**

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

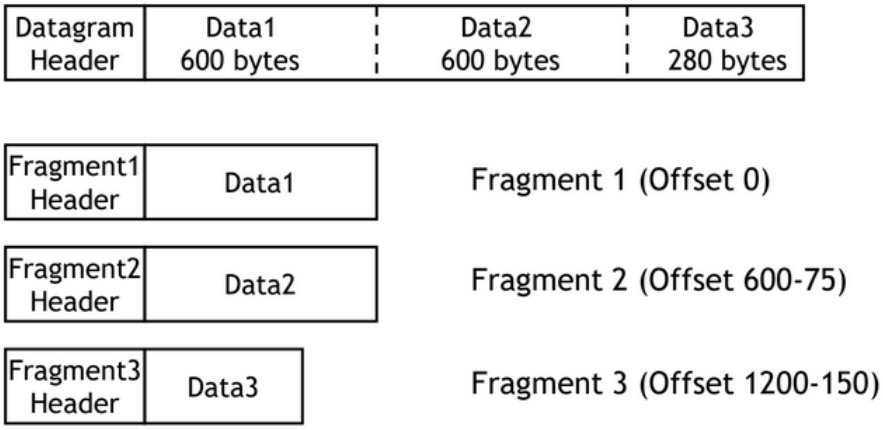
Cuanto **mas bytes puede enviar por mensaje, mas eficiente es la transmisión** (bits datos/ bits totales).

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

Funcionamiento



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



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

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

El **fragment header** contiene:

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

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

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

Desventajas de la fragmentación

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

Direccionamiento

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

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

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

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

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

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

Subredes

Por que surgen

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

Definición

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

Cantidad de subredes

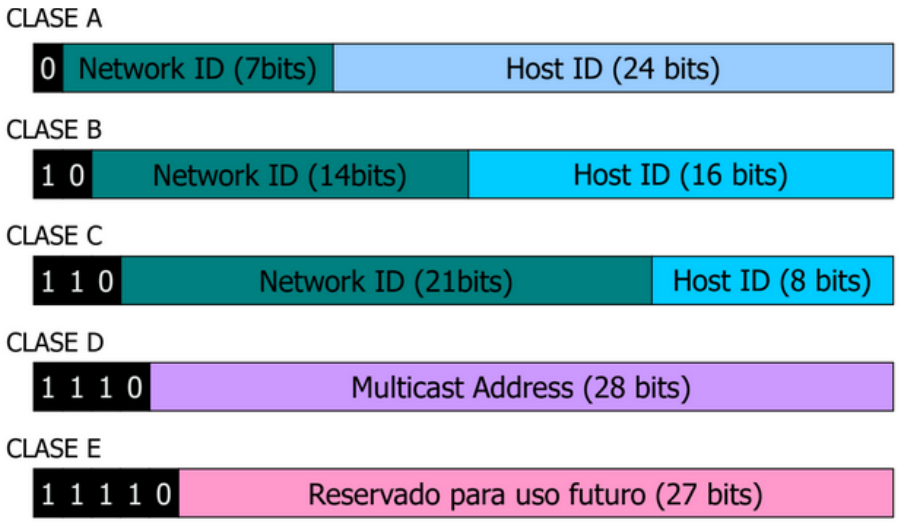
Cuando hacemos una partición la **mínima cantidad** de bits que puedo agarrar para formar subredes es **1bit**, formando **dos subredes** pero, por **estándar** (creo que igual ya no se le da bola al estándar), la **mínima cantidad** de bits del ultimo octeto que podemos agarrar para formar subredes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

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

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

Clases de direcciones

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



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

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

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

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

RFC1819 classfull

Dirección privada

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

Esos bloques son:

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

Dirección publica

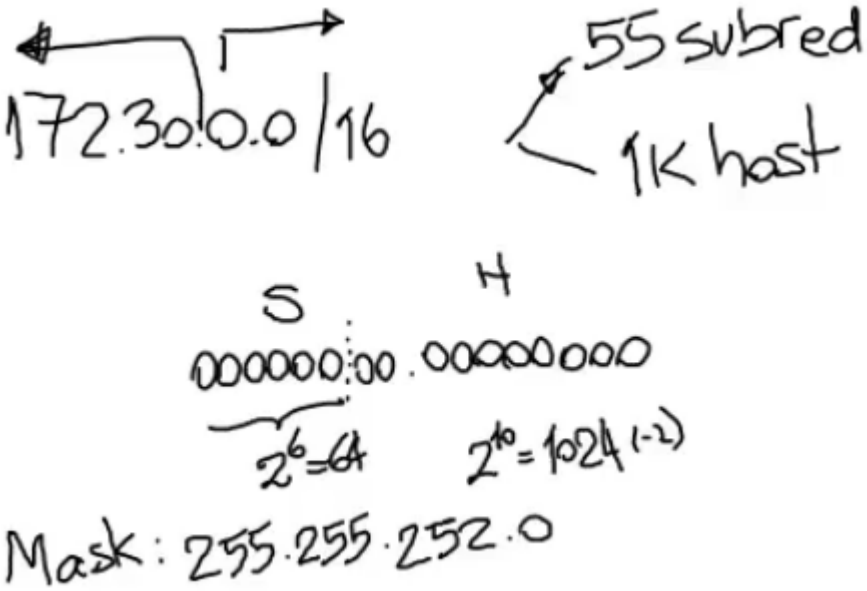
A los routers se le asigna una dirección publica, es decir con posibilidad de acceso a internet debido a que no existe la duplicidad. Esta dirección del router nos la asigna (presta) nuestro proveedor de internet que a su vez pidió a la IANA un bloque para sus clientes.

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

Dirección loop

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

Ejemplo de ejercicio subred



Classless interdomain routing (CIDR) y VLSM

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

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

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