

# ***RESUMEN DE PRÁCTICA***

## ***HTTP***

¿Cómo saber si un mensaje es de requerimiento o de respuesta?

Si tiene método, URL y versión es un mensaje de requerimiento; mientras que si indica la versión HTTP, el código de estado y la frase de razón, corresponde a una respuesta.

¿Alcanza con realizar un único requerimiento para visualizar la página completa con imágenes en un navegador?

No, no alcanza con un único requerimiento. Cada recurso, como un archivo CSS, un archivo JavaScript o una imagen, se solicita por separado al servidor.

¿Cuántos requerimientos serían necesarios para obtener una página que tiene dos CSS, dos Javascript y tres imágenes?

Se necesitarán: 1 requerimiento para el archivo base HTML, 1 requerimiento para el favicon, 2 requerimientos para CSS, 2 requerimientos para Javascript, 3 requerimientos para las tres imágenes.

¿Cómo sabe el cliente que ya recibió todo el objeto solicitado de manera completa?

En HTTP/1.0 el cliente se da cuenta de que ha recibido todo el objeto solicitado cuando el servidor cierra la conexión después de enviar la respuesta.

En HTTP/1.1: Se introducen mejoras para manejar este problema. El encabezado "ContentLength" indica al cliente la longitud en bytes del objeto en la respuesta, permitiéndole saber cuántos datos esperar. El encabezado "Transfer-Encoding" con valor "chunked" divide la respuesta en trozos, y el cliente detecta el final de la respuesta cuando recibe un trozo de tamaño 0.

HTTP/1.0, HTTP/1.1 son protocolos basados en textos.

. Códigos de Retorno HTTP:

1XX: Códigos informativos.

2XX: Códigos de éxito.

3XX: Códigos de redirección.

4XX: Códigos de Error del Cliente.

5XX: Códigos de Error del Servidor.

## ***DNS***

El servicio DNS (Domain Name System) funciona como un sistema distribuido de forma jerárquica, a través de dominios, sub-dominios y nombres finales, con un conjunto de servidores a lo largo del mundo. Cada servidor tiene la responsabilidad de mantener una parte dentro de la jerarquía de nombres.

Su objetivo principal es el de traducir nombres de dominio a direcciones IP para lograr una abstracción de las direcciones de red utilizadas internamente por los protocolos, permitiendo así ubicar a un dispositivo por su nombre sin importar cuál es su dirección IP actual, haciendo que no sea necesario para las personas recordar la dirección IP.

. *Respuestas Autoritativas y No Autoritativas:*

Una respuesta autoritativa es aquella dada por el servidor que tiene la autoridad sobre el nombre que se está consultando. Este responde directamente desde su base de datos de nombres, sin subdelegaciones ni cacheo de direcciones. Caso contrario, si se realiza esto último, se trata de una Respuesta NO Autoritativa.

El flag aa indica que la respuesta fue autoritativa.

. *Consultas Recursivas e Iterativas:*

En una consulta recursiva, el servidor DNS al que se le solicita la resolución de un dominio se encarga de hacer todo el trabajo. Si no conoce la respuesta, contacta con otros servidores DNS en nombre del cliente, obteniendo la respuesta final. El cliente solo recibe la respuesta completa y no interactúa con otros servidores DNS.

Los flags rd y ra indican recursividad en las consultas y respuestas.

Mientras que cuando se realiza una consulta iterativa, se consulta por IPs para ir acercándose a la respuesta final.

El cliente DNS interactúa con múltiples servidores DNS, haciendo varias solicitudes hasta obtener la respuesta final. El primer servidor consultado no resuelve completamente la consulta, sino que da una pista de a dónde debe ir el cliente para continuar la búsqueda (es decir, le proporciona la dirección de un servidor DNS más específico).

. *Resolver:*

El Resolver se lo podría considerar como un agente encargado de resolver los nombres a solicitud del cliente. Se puede tener un Stub/Dumb Resolver que no realiza ninguna forma de caching y deja que el encargado de esto sea el Servidor Local o un resolver activo, llamado Smart Resolver, que funciona en cada equipo como si fuese un Servidor Local, realizando caching u ofreciendo funcionalidades extras. Este suele hacer consultas recursivas.

. *Registros DNS:*

**A:** mapean un nombre de dominio a una dirección IPv4. Pueden existir varios registros (A) con el mismo nombre

**NS:** indican los servidores de nombre autoritativos para una sub-dominio. A partir de esto, se puede lograr una delegación de sub-dominios. No hay prioridad, todos los servidores tienen la misma precedencia.

Imagina que tienes un dominio principal: **ejemplo.com**, y decides delegar su gestión a los servidores de nombres **ns1.proveedor.com** y **ns2.proveedor.com**. Los registros NS en la configuración de **ejemplo.com** serían los siguientes:

ejemplo.com. IN NS ns1.proveedor.com.

ejemplo.com. IN NS ns2.proveedor.com.

**MX:** indican para un nombre de dominio cuáles son los servidores de mail SMTP encargados de recibir los mensajes para ese dominio. El servidor de mail SMTP que envía el mensaje deberá consultar, vía el servicio de DNS, cuáles son los servidores SMTP receptores para el dominio dado. Se asignan prioridades para servidores del mismo dominio.

Si hay más de un correo registrado, cuando se envíe un correo se intentará enviar al que tiene mayor prioridad, si no se puede se le entregará al que sigue en el orden de prioridades.

**CNAME:** mapean un nombre de dominio a otros nombres. Hacen el mapeo del alias de un dominio su nombre canónico (vendría a ser el nombre original).

Supongamos que tienes un dominio principal: **ejemplo.com**, y quieres que el subdominio [www.ejemplo.com](http://www.ejemplo.com) apunte al dominio principal.

El registro CNAME sería:

www.ejemplo.com. IN CNAME ejemplo.com.

**PTR:** mapean direcciones IP a nombres de dominio. Son el inverso de los registros (A).

Trabajan en el dominio especial in-addr.arpa.

**SOA:** se utilizan para proporcionar información autoritaria sobre una zona de dominio, lo que incluye información sobre la administración y configuración de esa zona. Solo se admite un registro SOA por zona. Permite que servidores autoritarios de la misma zona se puedan sincronizar.

**AAAA:** mapean un nombre de dominio a una dirección IPv6.

**TXT:** Son registros que mapean de un nombre de dominio a información extra asociada con el equipo que tiene dicho nombre, por ejemplo pueden indicar finalidad, usuarios, etc. No son utilizados habitualmente. Se los puede ver en uso asociando una clave pública, utilizando por ejemplo IPsec con un esquema de Opportunistic Encryption.

**SRV:** se utilizan para asociar servicios o recursos a nombres de dominio.

#### *. Servidores Primarios y Secundarios:*

Un dominio cuenta con más de un servidor, uno de ellos es el primario (o maestro) y todos los demás son secundarios (o esclavos) para simplificar la configuración de los servidores autoritarios, evitando configurar a cada servidor de un mismo dominio de forma independiente. En lugar de esto, se configura aquel que es primario y el resto de los servidores se sincronizan con este. Esto a su vez garantiza la consistencia de los datos DNS.

La transferencia de zona es la copia de la base de datos de nombres de un servidor primario a uno secundario. Esto permite mantener la consistencia entre los servidores de una zona de dominio.

#### *. Delegación de Dominios:*

Ej: *Imagine que usted es el administrador del dominio de DNS de la UNLP (unlp.edu.ar). A su vez, cada facultad de la UNLP cuenta con un administrador que gestiona su propio dominio (por ejemplo, en el caso de la Facultad de Informática se trata de info.unlp.edu.ar). Suponga que se crea una nueva facultad, Facultad de Redes, cuyo dominio será redes.unlp.edu.ar, y el administrador le indica que quiere poder manejar su propio dominio. ¿Qué debe hacer usted para que el administrador de la Facultad de Redes pueda gestionar el dominio de forma independiente?*

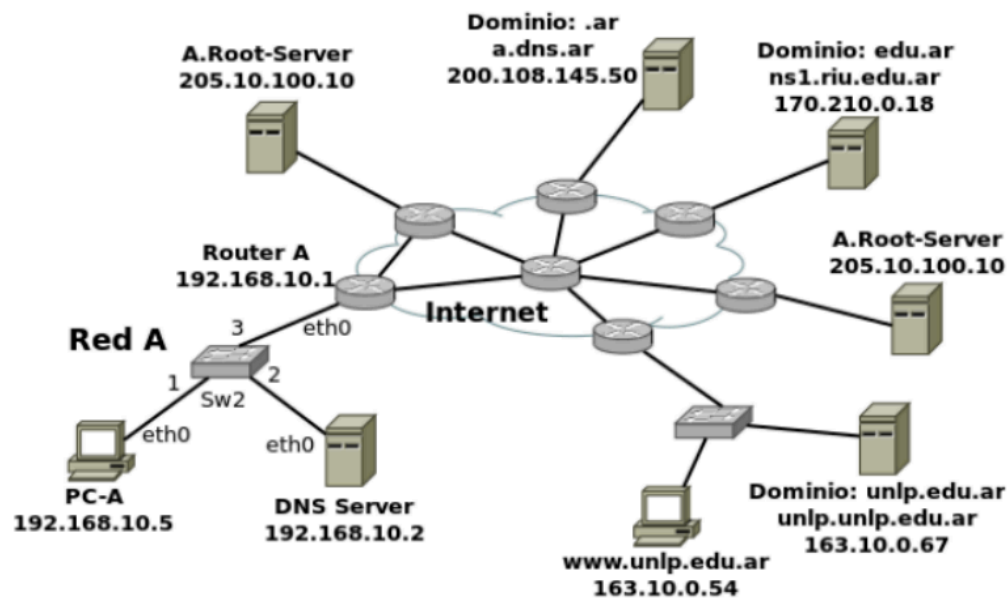
Debo delegarle la administración del dominio. Para ello transfiero la autoridad del subdominio "redes.unlp.edu.ar" al administrador de la Facultad de Redes. Hay que configurar registros NS que apunten a los servidores DNS de la Facultad de Redes como servidores autoritarios para ese subdominio. Para hacerlo, el administrador de la Facultad de Redes debe proporcionarle los nombres y direcciones IP de los servidores DNS que desea utilizar.

#### *. Obtener una IP:*

1. Consultar de forma recursiva al resolver privado. sobre la IP del host deseado.
2. Si el resolver tiene la IP cacheada la devuelve, sino delega al DNS Server.

3. Si el DNS Server no puede obtenerla de su caché, consultará de forma iterativa hasta llegar a la IP deseada. Cada respuesta que reciba tendrá los NS del dominio consultado.
4. Una vez obtenida la respuesta final, cacheará la misma y le responderá al resolver privado del cliente con la IP buscada, quien también la cacheará.

Ej:



Si la PC-A, que usa como servidor de DNS a "DNS Server", desea obtener la IP de [www.unlp.edu.ar](http://www.unlp.edu.ar), cuáles serían, y en qué orden, los pasos que se ejecutarán para obtener la respuesta.

1. PC-A (192.168.10.5) consulta primero a su resolver privado sobre la IP del host [www.unlp.edu.ar](http://www.unlp.edu.ar).
2. Si no puede obtener una respuesta (no está cacheada) el resolver primario delega al DNS Server (192.168.10.2).
3. Si este no puede obtenerla de su caché entonces el DNS Server consultará de forma iterativa al A.Root-Server (205.10.100.10) más cercano. Este le responderá (también de forma iterativa) con el NS (y el IP) de .ar, a.dns.ar (200.108.145.50).
4. DNS Server consultará (de forma iterativa) a a.dns.ar.
5. Este le responderá con los NS de .edu.ar, ns1.riu.edu.ar (170.210.0.18)
6. DNS Server consultará a ns1.riu.edu.ar
7. Este le responderá con el NS del servidor autoritativo del dominio unlp.edu.ar, unlp.unlp.edu.ar (163.10.0.67)
8. DNS Server consultará a unlp.unlp.edu.ar que le responderá con la IP de [www.unlp.edu.ar](http://www.unlp.edu.ar) (163.10.0.54)
9. El DNS Server cacheará la respuesta y le responderá al resolver de la PC-A con la IP de [www.unlp.edu.ar](http://www.unlp.edu.ar) (el resolver también la cacheará)

## MAIL

### . SMTP:

Protocolo utilizado para el envío de mails entre el cliente y su servidor de correo y entre servidores de correo.

### . POP3:

Protocolo utilizado para la recepción de mails.

Ofrece:

- **Simplicidad:** POP3 es un protocolo de acceso a correo extremadamente simple. Su simplicidad lo hace fácil de implementar y utilizar.
- **Descarga y Borrado:** POP3 generalmente se configura para descargar los correos electrónicos desde el servidor a la máquina local del usuario. En este modo, los correos se eliminan del servidor después de la descarga (aunque se pueden configurar para mantener una copia en el servidor).
- **No Mantiene Estado:** POP3 no mantiene información de estado entre sesiones. Esto significa que no guarda información sobre carpetas, mensajes marcados o cualquier otra información relacionada con el estado de la cuenta del usuario en el servidor.
- **Limitado para Usuarios Nómadas:** Para usuarios que desean acceder a sus correos electrónicos desde múltiples dispositivos, POP3 puede ser limitante ya que no ofrece una forma sencilla de sincronizar carpetas y correos entre dispositivos.

### . IMAP:

Protocolo utilizado para la recepción de mails.

Ofrece:

- **Funcionalidad Avanzada:** IMAP es más avanzado que POP3 y ofrece una amplia gama de funcionalidades. Permite a los usuarios organizar correos electrónicos en carpetas remotas, buscar mensajes, mover mensajes entre carpetas y realizar otras acciones avanzadas.
- **Mantiene Estado:** IMAP mantiene información de estado en el servidor. Esto significa que las carpetas, los mensajes marcados como leídos/no leídos, y otras acciones realizadas en un dispositivo se reflejan en todos los dispositivos conectados, lo que lo hace ideal para usuarios nómadas.
- **Acceso a Partes Componentes de los Mensajes:** IMAP permite a los usuarios acceder a partes específicas de los mensajes, como la cabecera o partes de un mensaje MIME. Esto es útil cuando se necesita descargar solo partes específicas de un mensaje para ahorrar ancho de banda.
- **Complejidad Adicional:** Debido a su mayor funcionalidad, IMAP puede ser más complejo de implementar tanto en el lado del cliente como en el lado del servidor en comparación con POP3.

### . Registro de Dominio:

Ejemplo sobre lo siguiente: *Suponga que registró bajo su propiedad el dominio redes2024.com.ar y dispone de 4 servidores:*

- Un servidor DNS instalado configurado como primario de la zona redes2024.com.ar. (hostname: ns1 - IP: 203.0.113.65).
- Un servidor DNS instalado configurado como secundario de la zona redes2024.com.ar. (hostname: ns2 - IP: 203.0.113.66).
- Un servidor de correo electrónico (hostname: mail - IP: 203.0.113.111). Permitirá a los usuarios enviar y recibir correos a cualquier dominio de Internet.

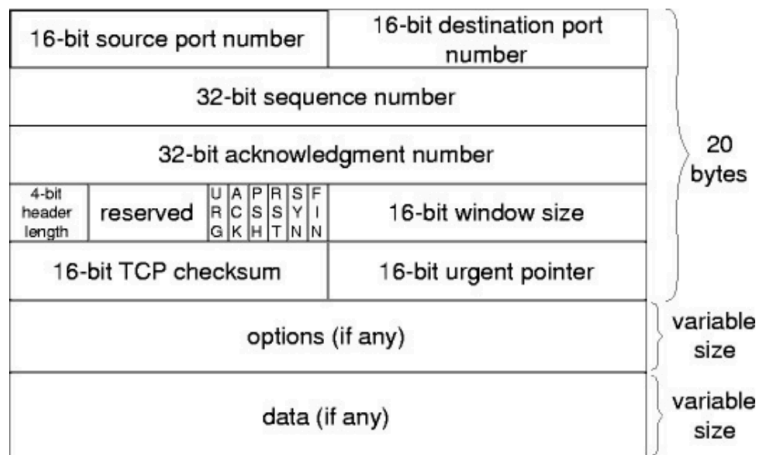
- *Un servidor WEB para el acceso a un webmail (hostname: correo - IP: 203.0.113.8). Permitirá a los usuarios gestionar vía web sus correos electrónicos a través de la URL <https://webmail.redes2024.com.ar>*

1. Informar al servidor autoritativo el NS de los servidores autoritativos del dominio a registrar y el A de dichos servidores autoritativos. Ej:
  - . El NS de los servidores autoritativos del dominio redes2024.com.ar (ns1 y ns2).
  - . El A de ambos servidores autoritativos.
2. Configurar registros en el servidor de nombres, indicando toda la información necesaria del archivo de zona (Nombre del registro, Tipo de Registro, Prioridad, TTL, Valor del Registro). Ej:
  - . redes2024.com.ar 86400 IN NS ns1.redes2024.com.ar.
  - . redes2024.com.ar 86400 IN NS ns2.redes2024.com.ar.
  - . redes2024.com.ar 86400 IN MX 5 mail.redes2024.com.ar.
  - . ns1.redes2024.com.ar 86400 IN A 203.0.113.65.
  - . ns2.redes2024.com.ar 86400 IN A 203.0.113.66.
  - . mail.redes2024.com.ar 86400 IN A 203.0.113.111.
  - . correo.redes2024.com.ar 86400 IN A 203.0.113.8.
  - . webmail.redes2024.com.ar 86400 IN CNAME correo.redes2024.com.ar.
  - . redes2024.com.ar 86400 IN SOA ns1.redes2024.com.ar
  - root.redes2024.com.ar 2023091300 604800 86400 2419200 86400
3. ¿El servidor de DNS debe aceptar consultas recursivas?  
Se debe pensar en si el servidor necesitaria resolver alguna petición consultando a otros servidores o no. Si es autoritativo, es decir, dueño de los dominios, no es necesario.
4. Configurar protocolos en los distintos servidores.  
DNS en los servidores DNS, SMTP e IMAP o POP3 en los de correo, y HTTPS en el servidor web.
5. Indicar puertos abiertos a internet (*servidor, protocolo de transporte y número de puerto*). Ej:
  - ns1/ns2 – UDP o TCP (en caso de que la respuesta exceda 512 bytes) – 53
  - correo – TCP – 80 (http) o 443 (https)
  - mail – TCP – 25 (SMTP), 110 (POP3) y 143 (IMAP).

## **CAPA DE TRANSPORTE**

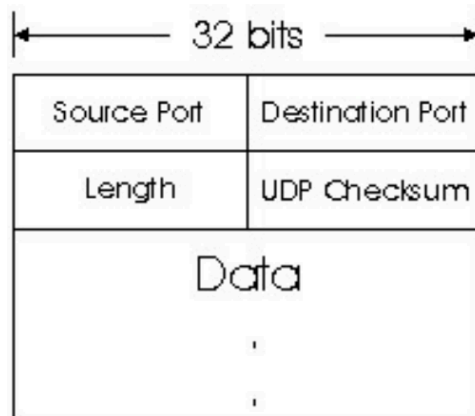
La función de la capa de transporte es gestionar la transmisión de mensajes o datos a través de una red, asegurándose de que lleguen de manera correcta y en el orden adecuado. Es decir, garantiza la entrega confiable de datos entre sistemas finales.

. *Segmento TCP:*



Si llega un segmento TCP con el flag SYN activo a un host que no tiene ningún proceso esperando en el puerto destino de dicho segmento (es decir, el puerto destino no está en estado LISTEN), el host enviará un segmento con la flag RST (reset) activado para indicar que la conexión no se puede establecer en dicho puerto. De esta manera informará al remitente que no hay no se puede establecer la conexión en ese momento para que no siga enviando segmentos.

. *Segmento UDP:*



Si llega un datagrama UDP a un host que no tiene ningún proceso esperando en el puerto destino de dicho datagrama (es decir, que dicho puerto no está en estado LISTEN), se responderá con un ICMP "Destination Unreachable" (esto es otro protocolo de IP). Este mensaje ICMP indica que el puerto o el host destino no están disponibles. Sin embargo, este paquete ICMP se puede perder y UDP no generará ninguna notificación al remitente ya que se trata de un protocolo sin conexión y no confiable.

Si detecta un error usando el checksum UDP no entrega el datagrama a la aplicación.

Para determinar si una conexión UDP fue exitosa o no se puede determinar con los mensajes ICMP.

En UDP el puerto origen puede ser 0 si no necesita una respuesta, puede ser simplemente un envío.

. *Utilización de puertos:*

Se utilizan para distinguir las aplicaciones (y, por lo tanto, protocolos) que están enviando/recibiendo datos. Los puertos actúan como puntos finales en una comunicación y permiten que múltiples aplicaciones en una misma computadora o dispositivo se comuniquen simultáneamente a través de la red.

. *Conexiones:*

Ante una salida del comando SS:

ESTAB: conexiones establecidas.

LISTEN: conexiones en estado listen o de espera.

IPS: corresponden a cada extremo de una conexión, si son iguales la conexión es local.

CLIENTES: se encuentran en estado ESTAB y utilizan puertos efímeros con números grandes, como por ejemplo 52123, 41220, etc (usualmente entre 32768 y 60999).

SERVIDORES: los que se encuentran en estado LISTEN y/o en puertos del 0 al 1023.

Además utilizan puertos conocidos o predefinidos como por ejemplo HTTP en el puerto 80, HTTPS en el 443, MySQL en el 3306.

TIME-WAIT: indica que el cierre de la conexión fue iniciado por el host local.

CLOSE-WAIT: indica que el cierre de la conexión fue iniciado por el host remoto.

SYN-SENT: conexiones pendientes por establecerse.

Si se intenta enviar un paquete mediante UDP a un puerto que no está escuchando para UDP se devolverá una ICMP Port Unreachable.

Si se intenta enviar un paquete mediante TCP a un puerto donde no hay ningún proceso en estado LISTEN se devolverá un flag RST/ACK.

. *Multicast:*

El multicast es una técnica que permite enviar un mensaje a un grupo de destinatarios de forma simultánea. A diferencia del broadcast, que envía un mensaje a todos los dispositivos de una red, el multicast solo envía el mensaje a los dispositivos que están interesados en recibirlo.

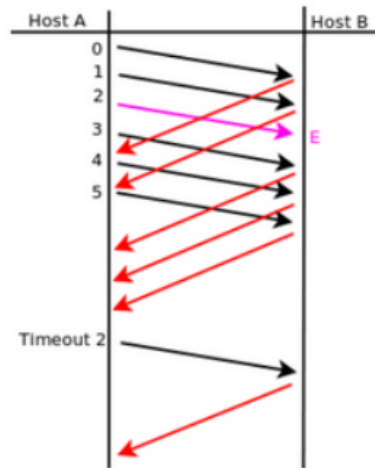
La técnica del multicast funciona sobre UDP, ya que no necesita establecer una conexión y se podría usar un mismo socket (un proceso tiene asociado un socket) para recibir datos de varios procesos que se quieren comunicar con un proceso a la vez.

Teóricamente podría intentarse adaptar multicast sobre TCP, pero sería demasiado complejo e iría en contra de la naturaleza del modelo ya que TCP establece una conexión punto a punto entre un único emisor y receptor.

. *Selective Repeat:*

Suponiendo Selective Repeat, tamaño de ventana 4 y sabiendo que E indica que el mensaje llegó con errores:





Host B envía los ACK 0, 1, 3, 4 y 5, ya que el 2 llega con error. Como no se pueden dejar huecos, no se corre la ventana hasta que 2 se retransmita, y se vuelve a enviar en el timeout 2.

El tamaño de la ventana no debe exceder la mitad del tamaño total del espacio de números de secuencia.

#### . Capturas TCP:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.20.1.1	172.20.1.100	TCP	74	41749 > vce [11111] Seq= Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=270132 TSecr=0
2	0.001264	172.20.1.100	172.20.1.1	TCP	74	vce > 41749 [SYN, ACK] Seq=1047471501 Ack=3933822138 Win=5792 Len=0 MSS=1460 SACK_PERM=1
3	0.001341			TCP	66	vce > [11111] Seq= Ack= Win=5888 Len=0 TSval=270132 TSecr=1877442

Internet Protocol Version 4, Src: 172.20.1.100 (172.20.1.100), Dst: 172.20.1.1 (172.20.1.1)

Transmission Control Protocol, Src Port: vce (11111), Dst Port: 41749 (41749), Seq: 1047471501, Ack: 3933822138, Len: 0

Source port: vce (11111)  
Destination port: 41749 (41749)  
[Stream index: 0]  
Sequence number: 1047471501  
Acknowledgement number: 3933822138  
Header length: 40 bytes

Flags: 0x012 (SYN, ACK)

000. .... = Reserved: Not set  
...0 .... = Nonce: Not set  
.... 0... = Congestion Window Reduced (CWR): Not set  
.... .0.. = ECN-Echo: Not set  
.... ..0. = Urgent: Not set  
.... ...1 = Acknowledgement: Set  
.... ....0 = Push: Not set  
.... .....0 = Reset: Not set  
... ..1. = Syn: Set  
.... ....0 = Fin: Not set  
Window size value: 5792  
[Calculated window size: 5792]  
Checksum: 0x9803 [validation disabled]

SYN → Comienzo de 3WH

3933822137 → Se que es ese porque el receptor (línea 2) me indica que espera (ACK) que se le envíe el segmento 3933822138, por lo tanto el que le envíe en 1 es 3833822138 – 1

172.20.1.1 → Source

172.20.1.100 → Destination

41749 → Puerto Destino

vce → Puerto Origen

ACK → Fin de 3WH

3933822138 → En el segmento anterior a este se indicó que se esperaba 3933822138

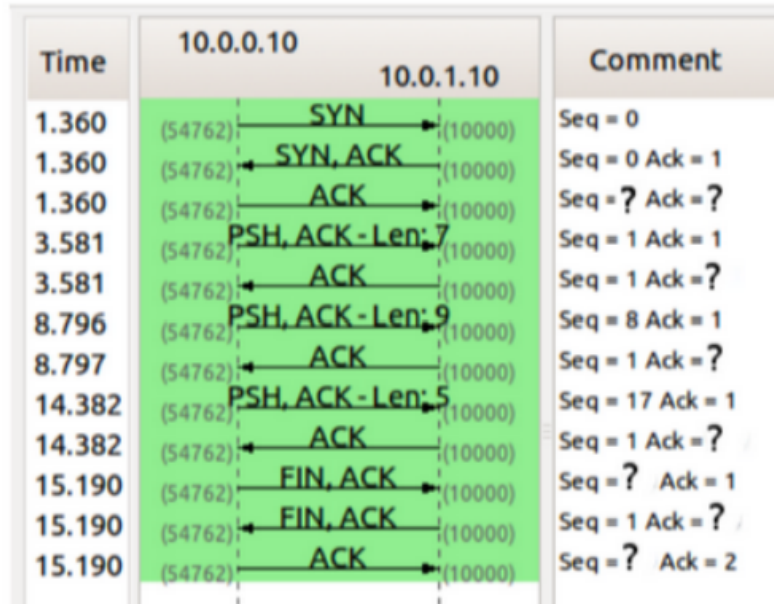
1047471502 → Se recibió 1047471501, por lo que se espera recibir 1047471502

Se debe hacer un SYN, SYN-ACK, ACK para el 3 Way Handshake.

En el SYN el 'seq=' es el número de secuencia que envió.

En el SYN-ACK 'seq=' es el número de secuencia que envió y 'ack=' el que espero recibir.

En el ACK 'seq=' es el número de secuencia que envió y 'ack=' el que espero recibir. Siempre 'seq=' es el número de secuencia que se envía de un punto al otro, y 'ack=' el próximo que se espera recibir. Puede haber un 'LEN' que indique la longitud del segmento enviado, y esto provocará que el próximo número de secuencia pedido sea el actual + LEN. Ej:



seq = 1 Ack = 1

Ack = 8

Ack = 17

Ack = 22

seq = 22

Ack = 23

seq = 23

Ej 2:

tcp.flags.syn eq 1 and tcp.flags.ack eq 0					
No.	Time	Source	Destination	Protocol	Length Info
961 82.420045	10.0.2.10	10.0.4.10	TCP	74 46907 → 5001	[SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=120632 TSecr=0 WS=16
963 83.540758	10.0.2.10	10.0.4.10	TCP	74 45670 → 7002	[SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=141236 TSecr=0 WS=16
967 97.960958	10.0.2.10	10.0.4.10	TCP	74 45671 → 7002	[SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=141517 TSecr=0 WS=16
961 135.753952	10.0.2.10	10.0.4.10	TCP	74 46910 → 5001	[SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=145124 TSecr=0 WS=16
1106 149.807117	10.0.2.10	10.0.4.10	TCP	74 54425 → 9000	[SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=158083 TSecr=0 WS=16

6 intentos de conexiones TCP ya que son todos [SYN]

Fuente	Destino
10.0.2.10:46907	10.0.4.10:5001
10.0.2.10:45670	10.0.4.10:7002
10.0.2.10:45671	10.0.4.10:7002
10.0.2.10:46910	10.0.4.10:5001
10.0.2.10:54424	10.0.4.10:9000
10.0.2.10:54425	10.0.4.10:9000

No.	Time	Source	Destination	Protocol	Length Info
4 0.000116	10.0.4.10	10.0.2.10	TCP	74 5001 → 46907	[SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=120650 TSecr=120632 WS=16
968 97.960923	10.0.4.10	10.0.2.10	TCP	74 5001 → 46910	[SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=145143 TSecr=145124 WS=16
982 135.754058	10.0.4.10	10.0.2.10	TCP	74 9000 → 54424	[SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=154588 TSecr=154569 WS=16
1107 149.807136	10.0.4.10	10.0.2.10	TCP	74 9000 → 54425	[SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=158102 TSecr=158083 WS=16

4 conexiones TCP exitosas, que se reconocen ya que tienen el flag SYN/ACK en 1, mientras que las fallidas tendrían los flags RST/ACK en 1.

Para saber que host es el que envió más cantidad de datos se puede observar cual es el que más incrementó su número de secuencia.

Para saber la cantidad de bytes que fueron confirmados se puede observar cual es el próximo número de byte que espera.

Por ejemplo, si van por seq=1, y se envían 20 bytes, la confirmación indica que confirmó los 20 bytes indicando que espera el byte nro 21.

Quien inicia el cierre de la conexión envía el flag FIN en 1.

#### *. RTT:*

El RTT (Round-Trip Time) es el tiempo que tarda un paquete en viajar desde un origen hasta un destino y de vuelta al origen. Se mide desde el momento en que se envía un paquete de datos hasta que el remitente recibe una respuesta. RTT incluye tanto el tiempo de transmisión (de ida y vuelta) como los tiempos de procesamiento en los dispositivos intermedios, como routers o servidores.

La opción de marcas de tiempo en TCP permite a los endpoints mantener una medición más precisa del tiempo de ida y vuelta (RTT) de la red entre ellos. Este valor ayuda a cada pila TCP a configurar y ajustar su temporizador de retransmisión. Hay otros beneficios, pero la medición RTT es la principal.

Para ello se incluye un Timestamp Value TSval en cada segmento que se envía. Los valores TSval se repiten en el lado opuesto de la conexión en el campo Timestamp Echo Reply TSecr. Entonces, cuando se confirma un segmento, el remitente de ese segmento puede simplemente restar su marca de tiempo actual del valor TSecr para calcular una medición precisa del tiempo de ida y vuelta (RTT).

$RTT = TSecr - TSval$

#### *. Control de Flujo:*

El control de flujo lo activa el receptor enviando ventanas más chicas. Esto deja en evidencia que el receptor tiene poco espacio (o no tiene más lugar) para seguir recibiendo datos. Esto se realiza a través del campo de tamaño de ventana en los encabezados de los segmentos TCP.

Resuelve el problema de la posible saturación o congestión de los buffers en los endpoints. Al indicar al emisor que reduzca la cantidad de datos que está enviando, evita que el receptor se sobrecargue.

Cuanto tiempo dura activo depende del receptor (más que nada la velocidad en que lee la aplicación). El control de flujo está activo mientras el receptor envíe ventanas más pequeñas (indicando capacidad limitada). Durará activo hasta que el receptor envíe ventanas más grandes, lo que indica que tiene más capacidad para recibir datos.

En todo momento ambos extremos están actualizando su propia ventana.

#### *. Control de Congestión:*

El control de congestión lo activa el emisor. El emisor limita la velocidad de transmisión de tráfico a través de su conexión en función de la congestión de red percibida. Este proceso es dinámico y adaptativo, y el emisor ajusta su velocidad de transmisión en respuesta a las condiciones cambiantes de la red.

Los posibles disparadores son:

- Fin de Temporización: La expiración del temporizador asociado con el envío de un segmento TCP puede ser interpretada como una señal de pérdida, indicando posiblemente congestión en la ruta.
- Recepción de TRES ACK Duplicados: La recepción de paquetes ACK duplicados procedentes del receptor también se interpreta como un suceso de pérdida. Este evento puede sugerir la pérdida de un paquete en la red debido a la congestión.

El objetivo es que no se desborde la propia red. Esto ocurre cuando hay más tráfico de red del que la red puede manejar eficientemente, lo que puede resultar en la pérdida de paquetes, retrasos elevados y un rendimiento de red deficiente. El control de congestión busca evitar que la red se sobrecargue ajustando la tasa de transmisión de datos del emisor para que sea compatible con la capacidad de la red.

## CAPA DE RED

. Clases de Redes:

CLASE	REDES	HOSTS	RANGO	MÁSCARA POR DEFECTO
A	128	16777214	0.0.0.0 a 127.255.255.255	/8
B	16384	655534	128.0.0.0 a 191.255.255.255	/16
C	2097152	254	192.0.0.0 a 223.255.255.255	/24

. *Importancia de las Máscaras de Subred:* Los prefijos de longitud fija por clase provocan un uso ineficiente en el espacio de direcciones y muchos equipos, produce escasez de direcciones.

Esto supone la aparición de las subredes, que básicamente permite que haya subgrupos en las redes, se utiliza para generar redes dentro de la red. Para ello toma una parte del hostid. 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 utilizan para saber en una dirección IP qué bits son de red y qué bits son de host.

¿Cómo saber cuál es la dirección de subred?

Para obtener la subred de la dirección IP **172.16.58.223/26**, necesitamos analizar la máscara de subred **/26**:

**Convertir la máscara de subred:** **/26** significa que los primeros 26 bits son 1 y los restantes 6 bits son 0. En formato decimal, esto corresponde a:

**255.255.255.192**

1. Esto significa que cada subred tiene 64 direcciones (de 0 a 63 en el último octeto).
2. **Identificar la dirección base de la subred:**

- La dirección IP 172.16.58.223 en binario es  
10101100.00010000.00111010.11011111.
- Aplicando la máscara 255.255.255.192 (o  
11111111.11111111.11111111.11000000 en binario), obtenemos la  
dirección base de la subred:  
172.16.58.192

172.16.58.223/26: 10101100.00010000.00111010.11011111 AND  
11111111.11111111.11111111.11000000 = 172.16.58.192

163.10.5.49/27: 10100011.00001010.00000101.00110001 AND  
11111111.11111111.11111111.11100000 = 163.10.5.32

128.10.1.0/23: 10000000.00001010.00000001.00000000 AND  
11111111.11111111.11111110.00000000 = 128.10.0.0

10.1.0.0/24: 00001010.00000001.00000000.00000000 AND  
11111111.11111111.11111111.00000000 = 10.1.0.0

8.40.11.179/12: 00001000.00101000.00001011.10110011 AND  
11111111.11110000.00000000.00000000 = 8.32.0.0

¿Cantidad máxima de hosts?

#### 172.16.58.223/26

- **Máscara:** /26 tiene 6 bits para hosts.
- **Cantidad de hosts:**  $2^6 - 2 = 64 - 2 = 62$  hosts.

163.10.5.49/27: 30

128.10.1.0/23: 510

10.1.0.0/24: 254

8.40.11.179/12: 1,048,574

¿Rango de IP válidas dentro de una subred?

#### Subred 172.16.58.223/26

- **Dirección de red:** 172.16.58.192
- **Broadcast:** 172.16.58.255
- **Rango de IP válidas:** 172.16.58.193 a 172.16.58.254

¿Es una dirección de host o de red?

Ej: 128.50.10.0

Es de host puesto que todos sus bits de hosts no están en 0.

Se trata de una dirección de clase B cuya máscara por defecto es 255.255.0.0.

128.50.10.0 en binario es 10000000 00110010 00001010 00000000 La máscara de red es 11111111 11111111 00000000 00000000

Podemos ver como los bits que corresponden a la parte de host no están en 0, por lo que se trata de una dirección de host.

#### . CIDR:

CIDR es una estrategia para frenar algunos problemas que se habían comenzado a manifestar con el crecimiento de Internet. Los mismos son:

- Agotamiento del espacio de direcciones de clase B.
- Crecimiento de las tablas de enrutamiento más allá de la capacidad del software y hardware disponibles.
- Eventual agotamiento de las direcciones IP en general.

CIDR 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.

Para publicar estas redes usando CIDR (Classless Inter-Domain Routing), se pueden agrupar en el bloque más grande posible para reducir la cantidad de rutas que el router necesita anunciar.

Ej:

- a. 198.10.1.0/24
- b. 198.10.0.0/24
- c. 198.10.3.0/24
- d. 198.10.2.0/24

Al observar los bits en común en cada dirección, podemos ver que los primeros 22 bits de todas las direcciones son iguales: **198.10.0.0/22**.

11111111 11111111 11111100 00000000

198.10.0.0/22

Esta red contiene todas las direcciones mencionadas.

#### . VLSM:

VLSM es realizar divisiones en subredes con máscaras de longitud variable. Sugiere hacer varios niveles de división en redes para lograr máscaras óptimas para las subredes. Se utiliza para evitar el agotamiento y desperdicio de direcciones IP.

Pasos:

- 1) Subnetear para la red con mayor cantidad de hosts.
- 2) De las subredes obtenidas, asignar todas las que se puedan con el menor desperdicio posible.
- 3) Si quedan segmentos de red sin una subred asignada volver al paso 1.

#### . ICMP:

Internet Control Message Protocol es un protocolo “helper” de IP. Brinda un “feedback” para poder resolver problemas en la red. Este protocolo se encapsula en IP.

Tipo	Código	Descripción	Uso Común
0	0	Echo Reply	Respuesta a una solicitud de ping.
3	0	Destination Network Unreachable	Red de destino inaccesible.
	1	Destination Host Unreachable	Host de destino inaccesible.
	2	Protocol Unreachable	Protocolo no soportado en el destino.
	3	Port Unreachable	Puerto no accesible en el destino.
	6	Network Unknown	La red de destino no es conocida.
	7	Host Unknown	El host de destino no es conocido.
4	0	Source Quench (deprecated)	Indica congestión en la red (obsoleto en IPv4 moderno).
5	0	Redirect Message (Redirect Datagram for Network)	Solicita al emisor cambiar la ruta de los paquetes.
8	0	Echo Request	Solicitud de ping.
9	0	Router Advertisement	Anuncio de routers disponibles.
10	0	Router Solicitation	Solicitud para encontrar routers cercanos.
11	0	Time to Live Exceeded in Transit	TTL agotado en tránsito.
	1	Fragment Reassembly Time Exceeded	Tiempo agotado al reconstruir fragmentos.
12	0	Parameter Problem: Pointer Indicates the Error	Problema con los parámetros del paquete.
13	0	Timestamp Request	Solicita información de tiempo al host.
14	0	Timestamp Reply	Responde a una solicitud de marca de tiempo.
15	0	Information Request (deprecated)	Solicita información sobre la red (obsoleto).
16	0	Information Reply (deprecated)	Respuesta a una solicitud de información (obsoleto).
17	0	Address Mask Request	Solicita la máscara de subred al host.
18	0	Address Mask Reply	Responde con la máscara de subred.

## RUTEO

El ruteo consiste en seleccionar la interfaz de salida y el próximo salto. Involucra a los routers y hosts. Es necesario para que un paquete vaya de un extremo a otro.

Cualquier máquina, esté o no conectada a internet, mantiene una tabla de ruteo.

Las tablas de ruteo se componen de:

RED DESTINO	MASK	NEXT-HOP	INTERFACE
Red alcanzable	Máscara de la red alcanzable	Es el destino previo, si es necesario, para llegar a la red destino.	Interface utilizada: eth0, eth1, eth2, ...

		(Sucedee cuando no hay un camino directo).	
--	--	--	--

Si se ingresa un next hop que no existe, al tomarlo, se entrara en un loop infinito hasta que finalice el TTL del paquete IP y se descarte.

. *Redes Públicas:*

1.0.0.0 a 9.255.255.255

11.0.0.0 a 126.255.255.255

128.0.0.0 a 172.15.255.255

172.32.0.0 a 191.255.255.255

192.0.0.0 a 192.167.255.255 (exceptuando las reservadas)

193.0.0.0 a 223.255.255.255

. *Redes Privadas:*

**Clase A:** 10.0.0.0 a 10.255.255.255

**Clase B:** 172.16.0.0 a 172.31.255.255

**Clase C:** 192.168.0.0 a 192.168.255.255

. *Direcciones Reservadas o Especiales:*

**Dirección de loopback:** 127.0.0.0 a 127.255.255.255

**Dirección de enlace local (APIPA):** 169.254.0.0 a 169.254.255.255

**Direcciones reservadas para documentación:** 192.0.2.0/24, 198.51.100.0/24, 203.0.113.0/24

**Direcciones reservadas para multicast:** 224.0.0.0 a 239.255.255.255

**Direcciones reservadas para futuras implementaciones:** 240.0.0.0 a 255.255.255.254

. *Subnetting:*

Se utiliza para dividir una red en partes más pequeñas y evitar desperdiciar direcciones IP que no van a ser utilizadas.

Si la red a asignar es demasiado grande y sobran muchas direcciones IP, se debe reducir disminuyendo la máscara y asignando solo las IP necesarias.

Ej:

Tengo la red 192.168.10. 00000000 /24 para asignar.

A la Red A debo asignarle 100 hosts, por lo que necesito una máscara /25 (de 7 bits) que me proporcione 128 direcciones. Pero /24 me proporciona 256 direcciones, desperdicio la mitad.

Además, a la Red B debo asignarle 70 hosts, por lo que necesito una máscara /25 (de 7 bits) que me proporcione 128 direcciones.

Entonces, puedo dividir la red en 2, y asignar 192.168.10.0/25 a la Red A, y luego asignar 192. 168.10.128/25 a la Red B.

. *Asignación de IPs:*

Se asigna la IP inicial a la red, y luego se asigna una IP a cada dispositivo.

Ej:

Red A está compuesta por RouterA, PC-A, PC-B, PC-C, PC-D.



192.168.10.0/25 para Red A.  
192.168.10.1 para RouterA.  
192.168.10.2 para PC-A.  
192.168.10.3 para PC-B.  
192.168.10.4 para PC-C.  
192.168.10.5 para PC-D.

. *Sumarización:*

La sumarización (también llamada agregación de rutas) es una técnica que se utiliza para simplificar las tablas de enrutamiento al combinar múltiples rutas de red en una sola entrada. Esto reduce el tamaño de las tablas de enrutamiento y mejora la eficiencia del procesamiento en los dispositivos de red.

Se agrupan varias redes con prefijos similares en una sola ruta que puede representarlas todas.

Esto se logra calculando un prefijo común basado en la máscara de subred más amplia que abarque todas las redes individuales.

Se puede aplicar en tablas de ruteo siempre que 2 redes tengan la misma interfaz y el mismo salto.

Imagina que tienes las siguientes redes en tu tabla de enrutamiento:

- **192.168.1.0/24**
- **192.168.2.0/24**
- **192.168.3.0/24**
- **192.168.4.0/24**

Estas redes tienen prefijos similares, por lo que puedes resumirlas en una sola ruta:

- **192.168.0.0/22**

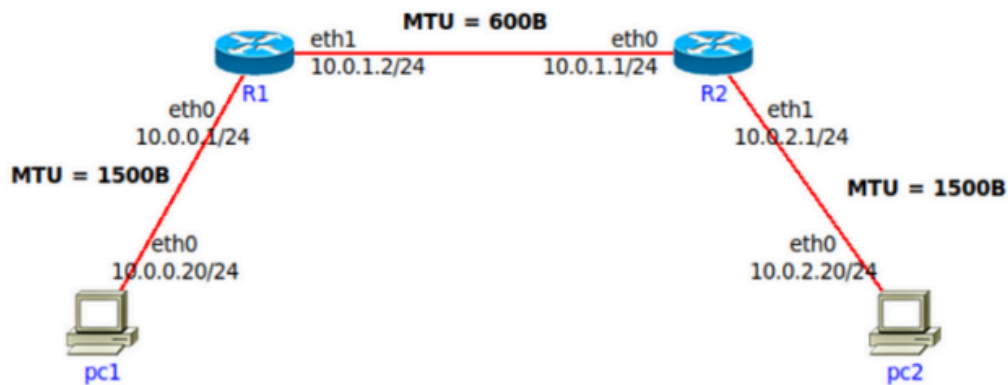
La máscara /22 abarca las cuatro redes, reduciendo la tabla de enrutamiento de 4 entradas a 1.

. *Fragmentación:*

Para fragmentar hay que tomar el valor máximo del MTU y restarle el valor del header (20), luego hay que encontrar el múltiplo de 8 más cercano a ese número.

Ej:

*Se tiene la siguiente red con los MTUs indicados en la misma. Si desde pc1 se envía un paquete IP a pc2 con un tamaño total de 1500 bytes (cabecera IP más payload) con el campo Identification = 20543, responder:*



- Indicar IPs origen y destino y campos correspondientes a la fragmentación cuando el paquete sale de pc1

IP origen: 10.0.0.20/24

IP destino: 10.0.2.20/24

Header: 20

Tamaño total: 1500

Identificación: 20543

DF Flag: 0

MF Flag: 0

Fragment Offset: 0

- ¿Qué sucede cuando el paquete debe ser reenviado por el router R1?

Como el enlace entre el router R1 y el R2 tiene un MTU de 600B, el paquete se debe fragmentar.

- Indicar cómo quedarían los paquetes fragmentados para ser enviados por el enlace entre R1 y R2.

Para fragmentar hay que tomar el valor máximo del MTU y restarle el valor del header (20), luego hay que encontrar el múltiplo de 8 más cercano a ese número.

1

Header: 20

Tamaño total: 596

Identificación: 20543

DF Flag: 0

MF Flag: 1

Fragment Offset: 0

2

Header: 20

Tamaño total: 596

Identificación: 20543

DF Flag: 0

MF Flag: 1

Fragment Offset: 72

3

Header: 20

Tamaño total: 348

Identificación: 20543

DF Flag: 0

MF Flag: 0

Fragment Offset: 144

Anotaciones:

- Al tamaño total le sumo el header. La suma de los totales de los fragmentos me debería dar el total del original + 20 \* (cantidad de fragmentos – 1)
- El offset se calcula como la suma del tamaño de datos (SIN HEADERS) de los fragmentos anteriores dividido por 8.
- El último fragmento tiene el MF Flag en 0.
- El primer fragmento tiene el offset en 0.

## **IPv6**

. Validación de Direcciones IPv6:

### **1. Verificar el formato general**

Una dirección IPv6 válida:

- Está formada por 8 grupos de hasta 4 dígitos hexadecimales cada uno.
- Los grupos están separados por el carácter de dos puntos (:).
- Cada dígito hexadecimal puede ser un número de 0-9 o una letra de a-f (mayúsculas o minúsculas).
  - Ejemplo válido: 2001:0db8:85a3:0000:0000:8a2e:0370:7334.

### **2. Reglas de compresión válidas**

IPv6 permite ciertos atajos de escritura:

- Compresión de ceros consecutivos: Se pueden reemplazar múltiples grupos de ceros consecutivos con :: (solo puede haber una compresión en la dirección).
  - Ejemplo válido: 2001:db8::1 (equivale a 2001:0db8:0000:0000:0000:0000:0000:0001).
- Supresión de ceros a la izquierda: Los ceros iniciales en un grupo pueden omitirse.
  - Ejemplo válido: 2001:db8:0:1:1:1:1:1.

### **3. Longitud total máxima**

- Una dirección IPv6 no puede superar los 128 bits. Esto significa que debe haber:
  - Exactamente 8 grupos si no hay compresión (::).
  - Un número de grupos total que, con la compresión, no supere 128 bits cuando se expanden los ceros.

### **4. Evitar errores comunes**

- Demasiados grupos de dígitos: Una dirección con más de 8 grupos no es válida.

- Ejemplo inválido: 2001:0db8:85a3:0000:0000:8a2e:0370:7334:abcd.
- Uso incorrecto de :: La compresión :: solo puede aparecer una vez en la dirección.
  - Ejemplo inválido: 2001::85a3::7334.

## 5. Prefijos y sufijos adicionales

- Prefijos de subred: Las direcciones IPv6 pueden incluir un sufijo de longitud de prefijo (/n) que indica la máscara de subred. Esto no afecta su validez.
  - Ejemplo válido: 2001:db8::1/64.
- Dirección con puerto: Las direcciones IPv6 en URLs a menudo se escriben entre corchetes con un puerto, pero esto no es parte de la dirección como tal.
  - Ejemplo válido: [2001:db8::1]:8080.

## 6. Tipos de direcciones válidas


Verifica que la dirección pertenezca a un rango permitido en IPv6:

- Link-local: FE80::/10.
- Global unicast: Cualquier otra dirección que no comience con FF ni FE.
- Multicast: FF00::/8.
- Unspecified: :: (todos ceros, usada temporalmente en ciertos contextos).
- Loopback: ::1.

## 7. Herramientas y comandos para validar direcciones IPv6

- Ping/Tracert: Usar herramientas como ping o tracert en sistemas operativos para validar conectividad y formato.
- Librerías de validación:
  - En Python: `ipaddress.IPv6Address()`.
  - En Linux: `ip -6 address`.

### Ejemplo práctico:

- Entrada: 2001:db8:85a3::8a2e:370:7334.
- Validación:
  1. Tiene un formato válido (8 grupos máximo, con compresión permitida).
  2. Usa caracteres hexadecimales válidos.
  3. Cumple con las reglas de compresión y longitud.
- Resultado: Dirección válida. 

Si la dirección no cumple con estas reglas, es inválida.

## CAPA DE ENLACE

### . Dispositivos:

HUB: Actúa como un repetidor y simplemente repite las señales a todos los puertos. No divide dominios de colisión ni de broadcast.

Switch: Examina las direcciones MAC para enviar tramas solo al puerto específico donde se encuentra el destinatario. Divide dominios de colisión. No divide dominios de broadcast.

Bridge: Conecta dos segmentos de red, examina las direcciones MAC y aprende las ubicaciones de las direcciones MAC en ambos lados. Puede dividir dominios de colisión, no divide dominios de broadcast (es como el switch solo que tiene menos puertos).

. **Broadcast:**

Dirección FF:FF:FF:FF:FF:FF.

La función principal de la dirección de broadcast es enviar información a todos los dispositivos en la red sin la necesidad de conocer sus direcciones MAC individuales.

El Router divide dominios de broadcast.

. **Colisión:**

Dos dispositivos intentan transmitir datos simultáneamente en una red compartida, causando una interferencia y resultando en la pérdida de datos.

Router, Switch y Bridge dividen dominios de colisión.

. **Tablas CAM:**

Las tablas **CAM** (Content Addressable Memory, o Memoria Asociativa) son un tipo especial de memoria utilizado principalmente en dispositivos de red como switches y routers. A diferencia de las memorias tradicionales, donde los datos se recuperan proporcionando una dirección específica, las tablas CAM permiten buscar datos proporcionando directamente el contenido que se desea encontrar.

Funcionamiento:

1. El switch recibe un paquete con dirección MAC de destino **AA:BB:CC:DD:EE:FF**.
2. Busca esa dirección en su tabla CAM.
3. Si la encuentra, identifica el puerto asociado y reenvía el paquete.
4. Si no la encuentra, realiza un broadcast para determinar en qué puerto está el dispositivo con esa dirección y actualiza la tabla CAM.

Switch2	
MAC	PORT
MAC_PC-A_eth0	0
MAC_PC-C_eth0	7
MAC_PC-B_eth0	1

. **ARP:**

El protocolo ARP mapea direcciones IP a direcciones MAC. Una estación emite una solicitud ARP para conocer la dirección MAC asociada a una dirección IP y recibe una respuesta con la información correspondiente. La tabla ARP en caché almacena estas asociaciones para futuras referencias.

## TABLAS ARP

Las tablas ARP (Address Resolution Protocol) son estructuras utilizadas en redes para almacenar la relación entre las direcciones IP y las direcciones MAC de los dispositivos en una red local (LAN). Son fundamentales para que los dispositivos puedan comunicarse dentro de una red basada en Ethernet.

Dirección IP	Dirección MAC	Tiempo de Vida
192.168.1.1	00:1A:2B:3C:4D:5E	20 segundos
192.168.1.2	00:1A:2B:3C:4D:5F	15 segundos

Funcionamiento:

1. Cuando un dispositivo necesita enviar un paquete a una dirección IP dentro de la misma red local, revisa su tabla ARP para ver si ya tiene la dirección MAC correspondiente.
  - 2.1. Si la dirección MAC no está en la tabla ARP:
    - Se envía un ARP Request (una consulta en broadcast) a toda la red para preguntar: "¿Quién tiene la dirección IP X.X.X.X?"
    - El dispositivo que posee esa dirección IP responde con un ARP Reply, proporcionando su dirección MAC.
    - La respuesta se guarda en la tabla ARP para futuros usos.
  - 2.2. Si la dirección está en la tabla, el dispositivo utiliza la entrada almacenada para enviar el paquete.

191.8.0.0/20

31 + primer direccion + ultima direccion = 33

5 bits = 32 direccion

6 bits = /26

191.8.0.0 necesito /26

191.8.0.0 - 191.8.0.127 = 128 direcciones

191.8.0.128 - 191.8.0.191 = 64 direcciones

191.8.0.192/26 - 191.8.0.255 ----> 191.8.1.0

Red D = 191.8.0.192/26

R-D = 191.8.0.193

PC = .194