Final 1)

1- Esquematizar y explicar los modelos OSI y TCP/IP.

2- Explicar control de flujo en TCP (ventana deslizable supongo).

3- Explicar cómo funciona la fragmentación de IP y para qué puede usarse bit DF.

4- Explicar y ejemplificar el uso de mensajes ICMP de redirección de ruta.

5- Un ejercicio de delegación de dominios DNS (estilo práctica).

1- Esquematizar y explicar los modelos OSI y TCP/IP.

Pila de protocolos: cada capa debe ofrecer **servicios** a sus capas superiores, mientras les oculta detalles relacionados con la forma en que se implementan los servicios ofrecidos

OSI: 7 capas. Son capas conceptuales, es un modelo más teórico

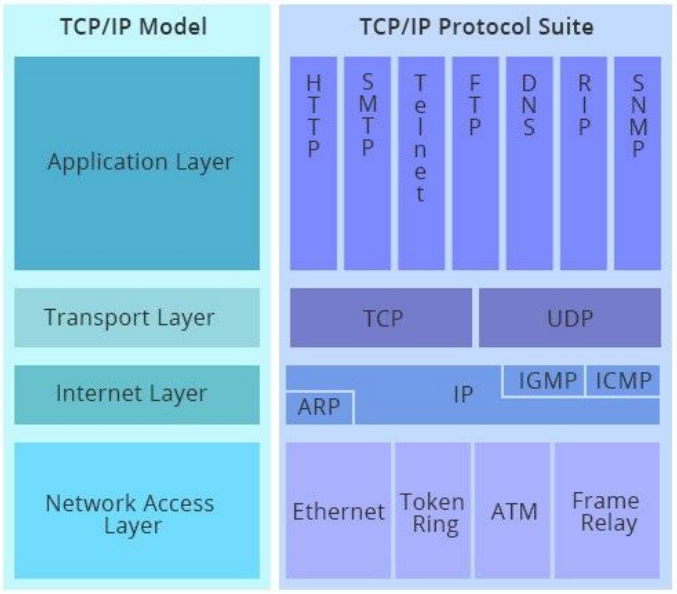
	APLICACIÓN	programas: ftp, http, ssh, mail, dns
	PRESENTACIÓN	se encarga de adaptar la información para que las máquinas puedan entenderse
	SESIÓN	permite establecer sesiones entre usuarios, control de diálogo entre aplicaciones
	TRANSPORTE	primera capa de comunicación extremo a extremo (punto a punto), realiza otra verificación de la transferencia
paq.	**RED**	ruteo de paquetes, control de congestionamiento
trama	**ENLACE (LLC + MAC)**	especifica la forma en que los datos viajan de un nodo a otro, formato y límite de tramas, control de errores
bit	**FÍSICA**	estándar para la interconexión física, incluye características de voltaje y corriente

Generalidades (OSI):

1) Servicio: indica qué hace cada capa, no la forma en que la entidad superior tiene acceso a ella o cómo funciona dicha capa
2) Interfaz: indica a los procesos que están sobre ella cómo accederla, cuáles son sus parámetros y qué resultados se esperan.
3) Protocolo: funcionamiento de la capa que garantiza que proporcione los servicios ofrecidos

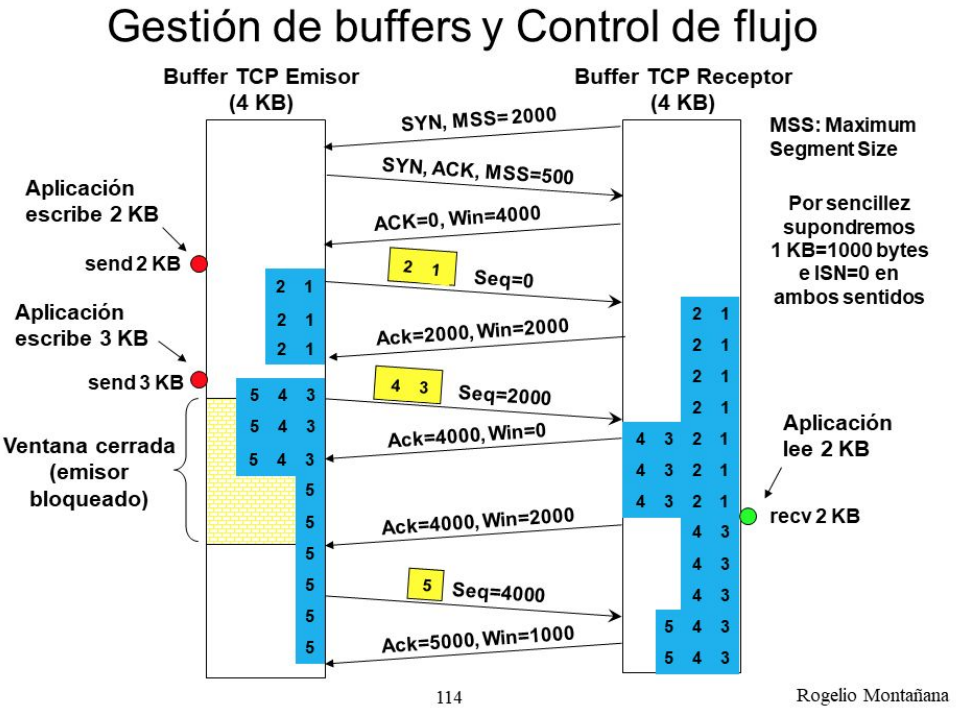
TCP/IP: modelo más aplicado en la práctica, no distingue entre los tres conceptos fundamentales de OSI (pero sí tiene protocolos concretos establecidos)

	APLICACIÓN	programas: ftp, http, ssh, mail
datagrama (UDP) segmento (TCP)	**TRANSPORTE**	primera capa de comunicación punto a punto, regula el flujo, **proporciona transporte confiable**, control de errores (acuses de recibo)
datagramas IP	**RED**	ruteo de paquetes, control de congestionamiento
tramas de red	**INTERFAZ DE RED (ENLACE)**	especifica la forma en que los datos viajan de un nodo a otro, formato y límite de tramas, control de errores



2- Explicar control de flujo en TCP (ventana deslizable supongo).

El control de flujo es algo directamente relacionado con el nivel de transporte. Significa cuidar que no se pierdan paquetes en el host de destino por desbordamiento en una conexión. Gracias al campo “tamaño de ventana” en la cabecera TCP, los extremos se comunican entre ellos anunciándose el tamaño de buffer que le queda libre para **esa** conexión (dado que el campo es de 16 bits, el tamaño máximo de ventana es 64KB - los tamaños se expresan en bytes). Cuando un TCP recibe datos los coloca en el buffer y en cada ACK que envía le anuncia al otro el espacio remanente. Si el buffer se llena anuncia ventana 0 y deja bloqueado al emisor (“ventana cerrada”).



VENTANA DESLIZABLE

3- Explicar cómo funciona la fragmentación de IP y para qué puede usarse bit DF.

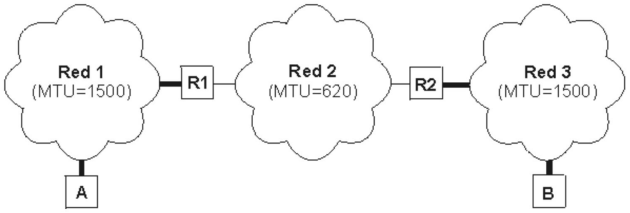
Cada red tiene su MTU (indica el tamaño máximo de trama soportado). Si se desea enviar un datagrama que supera ese tamaño, es necesario fragmentarlo.
La cabecera IP contiene campos útiles para la fragmentación:

	total length	
id	flags	offset

id: identifica el datagrama
total length: tamaño del fragmento
offset: posición del primer byte del segmento
flags: DF / MF

Cuando un router recibe un paquete de tamaño mayor que el MTU de la red del próximo salto, puede optar por fragmentar el paquete o descartarlo y enviar un mensaje ICMP al emisor (“Destination unreachable, fragmentation needed but DF activated”) para que lo fragmente en origen. Esta decisión depende del flag “DF” (“Don’t Fragment”). En IPv4, tanto host como routers pueden realizar el proceso de fragmentación. Un paquete puede ser fragmentado más de una vez durante la ruta hacia destino.

Ejemplo: tamaño paquete = 1400 bytes



encabezado datagrama	DATA 1 600 octetos	DATA 2 600 octetos	DATA 3 200 oct

encabezado datagrama	DATA 1 600 octetos	offset=0 MF=1

encabezado datagrama	DATA 2 600 octetos	offset=600 MF=1

encabezado datagrama	DATA 3 200 oct	offset=200 MF=0

El flag DF también puede usarse para hacer "Path MTU Discovery". Proceso por el cual el host que desea enviar un paquete primero descubre cuál es el mínimo MTU de todo el trayecto hacia destino, para luego realizar él mismo la fragmentación, solo una vez en origen (no deja librado a los router que la hagan durante el recorrido).

PMTUD

1- El host envía paquetes con el bit DF activado
2- Cuando el paquete se topa con un enlace con un MTU menor al tamaño de paquete, este envía un mensaje ICMP a origen con el error (destino inaccesible) y con el tamaño de su MTU
3- Luego que el host recibe el mensaje con el nuevo MTU, envía paquetes con este tamaño (más el flag DF activado)
4- El proceso se repite hasta que el paquete llega a destino y así se concluyó cuál es el menor MTU del trayecto.

4- Explicar y ejemplificar el uso de mensajes ICMP de redirección de ruta.

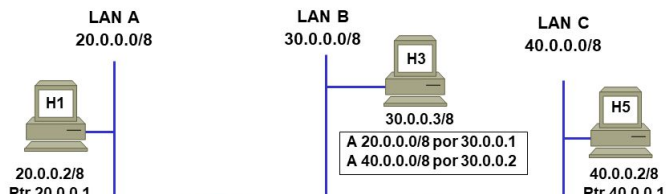
Estos mensajes los envía un router al host que envía paquetes para que cambie de ruta. Se utiliza para optimizar las rutas y que los paquetes lleguen más directos a destino en las ocasiones que tiene varios caminos para llegar al mismo destino.

Condiciones para que el router envíe el mensaje:

1- la interfaz de entrada y salida del paquete debe ser la misma
2- la IP de origen del paquete debe estar en la misma red que la IP de enrutamiento (la nueva ruta a destino)

Ejemplo:

3 redes locales
2 routers con 2 interfaces



Si **H4** quiere enviar un paquete a **H6**, y siendo **X** su default gateway, el paquete va a recibirlo **X**.
Como **X** tiene en su tabla de ruteo que para llegar a **H6** el próximo salto es una IP de la red **30**, y el paquete proviene de una IP **30**, entiende que hay algo mal (le están pidiendo que enrute un paquete por donde entró). Por eso envía un ICMP Redirect a **H4** con la IP del enlace del router más directo (**Y**)

Final 2)

1- Se quiere usar cable UTP o fibra óptica para tender una red Ethernet. Comente y explique qué conviene.

2- ¿Puede ser una dirección de difusión un número par? En caso afirmativo dé un ejemplo.

~~3- Explique el proceso de fragmentación. Dé ventajas y desventajas de reensamblar un datagrama cuando el MTU de la red a la cual será ruteado el datagrama lo permite.~~
Ver respuesta amplia

4- Se quiere conectar dos redes Ethernet con dos puentes en paralelo. ¿Qué problemas aparecen? Anda?

5- ¿Es posible que 2 máquinas mantengan una conexión TCP con un servidor y estas conexiones estén en un mismo puerto? Explique.

1- Se quiere usar cable UTP o fibra óptica para tender una red Ethernet. Comente y explique qué conviene.

Una red Ethernet se puede tender tanto con cable de par trenzado (UTP) como con fibra óptica. Generalmente la decisión depende de la distancia y velocidad de transmisión:

- Los cables de par trenzado no soportan distancias mayores a 100 metros, y las velocidades varían entre 10 Mb/s (10BaseT) y 1 Gb/s (10GBaseT, categoria 7UTP)
- Los cables de fibra óptica pueden alcanzar distancias mucho mayores, del orden de los kilómetros sin atenuar la señal, y también velocidades superiores (pueden llegar hasta los 50 Tb/s!)

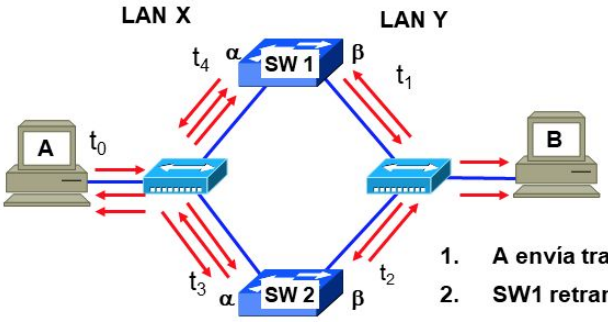
Se debe considerar el aspecto económico. Armar una red de fibra óptica es mucho más costoso, especialmente, por el hardware necesario en las interfaces. Además se requiere de personal especializado en el manejo de cables de fibra.
Es común que la fibra óptica se utilice en la transmisión de larga distancia en redes troncales de servicio de internet (red WAN). En cambio, los cables UTP suelen utilizarse en redes LAN.

2- ¿Puede ser una dirección de difusión un número par? En caso afirmativo dé un ejemplo.

No es posible. Las direcciones de broadcast tienen todos los bits de la parte de host en 1 (binario), lo cual es imposible que una dirección sea un número par si el último bit del número es 1.

4- Se quiere conectar dos redes Ethernet con dos puentes en paralelo. ¿Qué problemas aparecen? Anda?

Puede ocurrir el problema del bucle entre puentes:



1. A envía trama t_0 a LAN X
2. SW1 retransmite t_0 en LAN Y como t_1
3. SW2 retransmite t_0 en LAN Y como t_2
4. SW2 retransmite t_1 en LAN X como t_3
5. SW1 retransmite t_2 en LAN X como t_4
6. ... y así sucesivamente.

Enviando una sola trama la red se satura

Esto provoca que la red quede inutilizable.
Una forma de resolverlo es habilitando un mecanismo de detección para que desactive las interfaces necesarias para que ya no haya bucles. El protocolo "Spanning Tree" es un protocolo de comunicación entre puentes (a través de mensajes/tramas **BPDU** - "Bridge Protocol Data Units") que permite a los dispositivos activar o desactivar automáticamente los enlaces de conexión de forma que garantice la eliminación de bucles. El protocolo establece identificadores por puente y elige el que tiene prioridad más alta, el cual establecerá el camino de menor coste para todas las redes.

5- ¿Es posible que 2 máquinas mantengan una conexión TCP con un servidor y estas conexiones estén en un mismo puerto? Explique.

El TCP permite que varios programas de aplicación se comuniquen de manera concurrente y realiza el demultiplexado del tráfico TCP entrante. Los números de puerto identifican el destino final dentro de una máquina. El TCP se diseñó según la abstracción de conexión (circuito virtual) entre dos programas de aplicación.
Como TCP *identifica una conexión por medio de un par de puntos extremos [IP:puerto origen y destino]* (asocia los mensajes entrantes con una conexión en lugar de hacerlo con un puerto de protocolo, utiliza ambos puntos extremos para identificar la conexión apropiada), varias conexiones en la misma máquina pueden compartir un número de puerto
Por ejemplo: un servidor de mail, el cual permite que varias computadoras le envíen correo electrónico de manera concurrente

Final 3)

1- Explique las funciones de un bridge, hub y switch explicando sus diferencias.

~~2- Igual que el 3 anterior pero SOLO explicar las desventajas.~~

3- Explique el algoritmo por el cual un host decide si debe hacer una entrega directa de un datagrama o si debe rutearlo

~~4- Igual que el 4 anterior~~.

5- Explique los algoritmos por los cuales se calcula el tiempo de retransmisión en el protocolo TCP.

1- Explique las funciones de un bridge, hub y switch explicando sus diferencias.

La función de estos dispositivos es la interconexión, desplazar tramas y paquetes de una computadora a otra, la diferencia está en que trabajan a distinto nivel del modelo OSI.
Un hub (concentrador) permite estructurar el cableado de redes, interconectando los dispositivos que se enchufan en cada interfaz, A NIVEL FÍSICO. No cumple ninguna función adicional. Las tramas que llegan a una interfaz se distribuyen a todas las demás.
Los bridges y switches trabajan a nivel de enlace. Funciones:

- separan redes a nivel de MAC (usando tablas CAM)
- mejoran el rendimiento separando el tráfico local (sólo pasa lo que va de una red a otra - direccionado, NO difusión por inundación)
- permiten la interoperabilidad entre distintos tipos de redes y velocidades (ethernet - wifi)
- aumentan la seguridad (los sniffers ya no capturan todo el tráfico),
- aumentan la fiabilidad (actúan como puertas cortafuegos, un problema ya no afecta a toda la red)
- mejor alcance
- permite un mayor número de estaciones

Cada interfaz en el switch es un dominio de colisión distinto.
La diferencia entre un bridge y un switch es la cantidad de interfaces: un bridge tiene de 2 a 6, un switch puede tener hasta 500.

3- Explique el algoritmo por el cual un host decide si debe hacer una entrega directa de un datagrama o si debe rutearlo

La transmisión de un datagrama dentro de una misma red no involucra routers (no hay enrutamiento). Para saber si el destino está en la misma red, el transmisor compara los prefijos de la direcciones IP (origen y de destino). En el caso que tengan el mismo prefijo, están en la misma red, entonces sabe que tiene que hacer una **entrega directa**. Si no coinciden los prefijos de red, entonces debe enrutarlo a través del router.
Los routers emplean una tabla de ruteo en las que se almacenan información de posibles destinos y cómo alcanzarlos. Por lo tanto, cuando llega un paquete al router se fijará en la tabla de ruteo si tiene la dirección de destino en uno de sus

enlaces (para hacer entrega directa) o qué enlace debe atravesar el paquete para alcanzarla.

5- Explique los algoritmos por los cuales se calcula el tiempo de retransmisión en el protocolo TCP.

Cada vez que el emisor envía un mensaje, comienza un timer esperando la respuesta del receptor (ACK). El escenario en el que se dá la necesidad de retransmisión es cuando se terminó el tiempo del timer y el emisor entiende que el paquete se perdió. Entonces hay que retransmitir.
Si el tiempo de retransmisión es demasiado grande, el emisor espera inútilmente - baja el rendimiento. Si es demasiado bajo, se hacen reenvíos innecesarios - baja el rendimiento. Por eso se utiliza un algoritmo **autoadaptativo** a partir del registro de los tiempos de respuesta del receptor (ACKs). Conforme cambia el desempeño de una conexión, se ajusta el tiempo de retransmisión.
Para recolectar los datos necesarios para un algoritmo adaptable, TCP guarda la hora en que se envían los segmentos y la hora de sus ACKs. Considerando las dos horas, se computa el tiempo transcurrido, conocido como “tiempo de viaje redondo”. Siempre que obtiene una nueva muestra de viaje redondo, el TCP ajusta su noción del tiempo de viaje redondo **promedio** para la conexión. Por lo general, el software TCP. almacena el tiempo estimado de viaje redondo, RTT (“round trip time”), como promedio calculado y utiliza nuevos ejemplos de viaje redondo para cambiar lentamente dicho promedio.

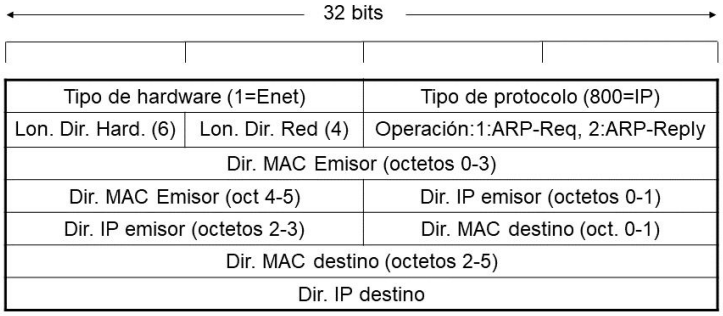
Algoritmo de Karn: se descartan los ACKs de segmentos retransmitidos del cálculo del promedio de RTT
Se dice que los ACKs son ambiguos porque un emisor nunca puede saber si corresponden a un reenvío o no. Si llega un acuse de recibo después de una o más retransmisiones, el TCP medirá la muestra de viaje redondo de la transmisión original y computará un RTT nuevo utilizando la muestra excesivamente larga. Por lo tanto, el RTT crecerá ligeramente.
Si tanto la transmisión original como la más reciente fallan en proporcionar tiempos de viaje redondo. Se concluye entonces que el TCP **no** debe actualizar la estimación de viaje redondo para los segmentos retransmitidos.

Final 4)

1- ¿ARP puede funcionar sin modificaciones con otros protocolos de capa de red que no sean IP?

ARP funciona para cualquier tipo de red (nivel enlace: ethernet, token ring, etc) y de protocolo de red. Esto es posible gracias a que los campos de su cabecera están diseñados para que su uso sea flexible. Se cambiarían los campos, por ejemplo, tipo de hardware, tipo de protocolo, y, como consecuencia, longitud de dirección de hardware y longitud de dirección de red.

Formato de ARP para IPv4 y Ethernet



Tipo de hardware (1=Enet)		Tipo de protocolo (800=IP)
Lon. Dir. Hard. (6)	Lon. Dir. Red (4)	Operación:1:ARP-Req, 2:ARP-Reply
Dir. MAC Emisor (octetos 0-3)		
Dir. MAC Emisor (oct 4-5)		Dir. IP emisor (octetos 0-1)
Dir. IP emisor (octetos 2-3)		Dir. MAC destino (oct. 0-1)
Dir. MAC destino (octetos 2-5)		
Dir. IP destino		

En el ARP-Request la MAC de destino está a ceros

2- "A" es un cliente de correo. "S" es el servidor. A se conecta a S. Termina la transmisión y envía el mensaje de FIN. Hay un cambio de ruta y ese mensaje no llega a S. Entonces, A retransmite el mensaje, llega, y se cierra la conexión normalmente. Segundos después, A vuelve a conectarse con S, y luego llega a S el FIN perdido que había enviado A en la sesión anterior. ¿Este mensaje corta la nueva conexión de A con S?

No, porque cada conexión está asociada a algún identificador, y S notaría que ese mensaje de "FIN" era de un número usado anteriormente, pero que no esté en uso ahora. Dijo Kohan, que se usa una especie de regla, para que los números que se usan para identificar una conexión no se usen más por un periodo razonable de tiempo, para no confundir las conexiones entre sí.

3- ¿Cómo colabora el campo TTL para que el protocolo IP funcione correctamente?

Evita que queden mensajes vagando por la red indefinidamente, evitando así el colapso de la misma.

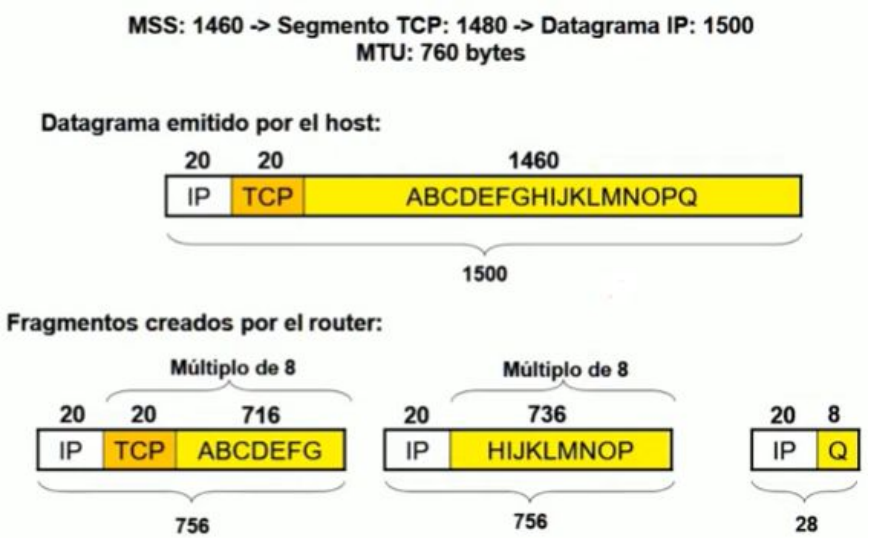
4- Explicar el proceso de fragmentación de segmentos TCP <-Cap. 13

Explicas el proceso, haciendo referencia a los campos que intervienen. A algunos chicos les pidió que dibujaran el encabezado TCP, pero no es eso lo que él quiere (se ve que esos chicos no explicaron bien el proceso). Yo no dibuje nada, pero a medida que iba contando la historieta de cómo se producía la fragmentación y reensamblado, iba nombrando los campos que intervenían, y cómo se seteaban en ese momento, en que se fijaba el transmisor, o el receptor, etc.

Los grandes segmentos TCP resultan en grandes datagramas IP. Cuando estos datagramas viajan a través de una red con MTU pequeña, IP los fragmenta. Para la capa de TCP esto es transparente. A diferencia de un segmento TCP completo/entero, un fragmento no se puede confirmar o retransmitir en forma independiente. Por lo que TCP intenta evitar la fragmentación para optimizar el rendimiento.
La idea es que se consiga el tamaño de paquete máximo posible sin fragmentar (un MTU que sea válido para todo el trayecto): PMTUD **"Path MTU Discovery".**

MSS (maximum segment size): cualquier TCP cuando se conecta (SYN) anuncia un MSS que va a tolerar (esta dado en función de las posibilidades de la interfaz física y del

buffer que el SO le asignó). Esto no asegura que en el trayecto del paquete no atraviese redes que tengan un MTU menor y entonces IP deba fragmentar. El MSS de cada extremo solo es información local, cercana a su interfaz, no conoce nada sobre el trayecto



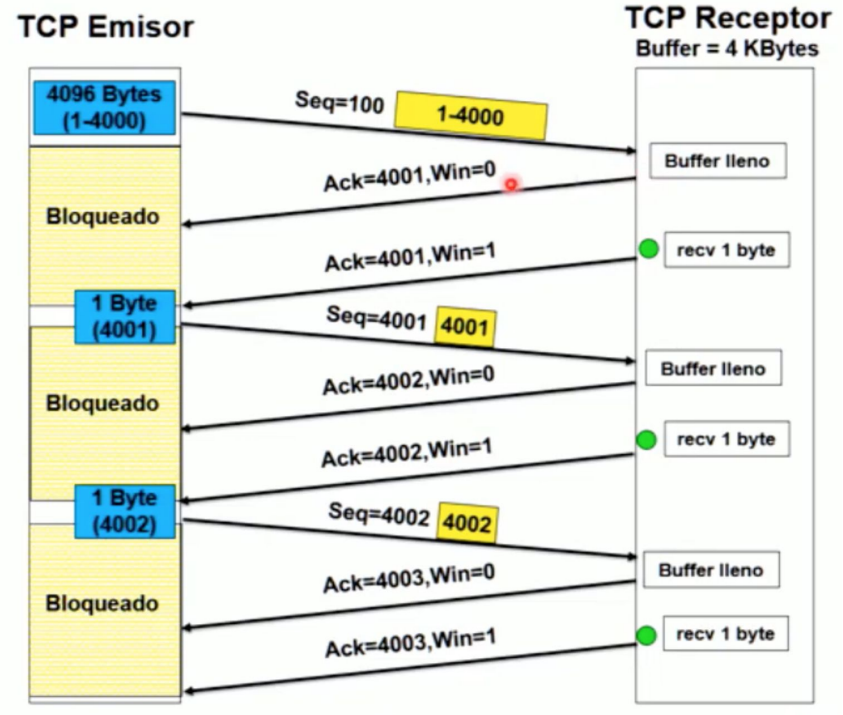
Notar que:
1- el MSS es 1460 (no incluye cabeceras IP ni TCP)
2- si fuera el caso, como en este, que IP necesita fragmentar, la cabecera IP viaja en cada fragmento, pero la de TCP no, ya que se considera como playload del paquete IP

5- Explicar el síndrome de la ventana tonta, y las soluciones (ambos casos).

Acá también, está todo en el libro. También tenés que ir nombrando los campos de la cabecera que intervienen... contar la historia, digamos.

El problema se presenta cuando los extremos trabajan a velocidades diferentes. Si el receptor no puede procesar los segmentos al ritmo que son enviados, solicita al emisor que reduzca el flujo de datos y achica el tamaño de la ventana. Si el receptor sigue sin poder procesar, va achicando su ventana cada vez más, al punto que los datos transmitidos son más chicos que el encabezado del paquete. Esto genera una transmisión ineficiente.

Otra explicación:
Ocurre cuando, estando lleno el buffer del TCP receptor, la aplicación lee los datos en pequeñas dosis. TCP anunciará cada vez que se produzca un hueco (pequeño) en el buffer y el emisor se verá obligado a enviar segmentos pequeños.



La solución de Clark:
El TCP receptor sólo debe notificar una nueva ventana cuando tenga una cantidad razonable de espacio libre: un MSS o la mitad del espacio asignado en el buffer a esa conexión.

6- Solución al congestionamiento (en TCP). <- Cap. 13 FALTA/VER MÁS

Mismo sistema que en las anteriores (contar el proceso e ir nombrando los campos del encabezado que intervienen).

Para los puntos extremos el congestionamiento significa un mayor retraso, respondiendo con la terminación de tiempo y retransmisiones, las cuales empeoran el congestionamiento. La capa de red detecta la congestión cuando las colas crecen demasiado en los enrutadores y trata de lidiar con este problema, aunque lo único que haga sea descartar paquetes.
Los protocolos de transporte pueden ayudar a evitar el congestionamiento al **reducir automáticamente la velocidad de transmisión** siempre que ocurra un retraso.

TCP utiliza dos algoritmos: **ARRANQUE LENTO** Y **DISMINUCIÓN MULTIPLICATIVA**

Para el control de congestión el TCP (emisor) mantiene un campo de información llamado “ventana de congestionamiento”. En un caso normal de conexión sin congestión, esta ventana es igual al tamaño de ventana del receptor. La reducción de la ventana de congestión reduce el tráfico que TCP inyecta en la conexión.

Prevención del congestionamiento por **Disminución Multiplicativa**: Cuando se pierda un segmento, reducirá a la **mitad** la ventana de congestionamiento (hasta un mínimo de un segmento). Para los segmentos que permanezcan en la ventana permitida, anular exponencialmente el temporizador para la retransmisión.

Recuperación de **arranque lento**: siempre que se arranque el tráfico en una **nueva conexión** o se **aumente el tráfico después de un periodo de congestionamiento**, activar

la ventana de congestionamiento con el tamaño de un solo segmento y aumentarla un segmento cada vez que llegue un acuse de recibo.
(Los arranques lentos evitan saturar la red con tráfico adicional, justamente después de que se libere un congestionamiento o cuando comienzan repentinamente nuevas conexiones)

Juntos, el incremento de arranque lento, la disminución multiplicativa, la prevención de congestionamiento, la medida de variación y la anulación exponencial del temporizador mejoran notablemente el desempeño del TCP, sin agregar ningún trabajo computacional significativo del software de protocolo.

Final 5)

~~1- Síndrome de la ventana tonta.~~

2- MTU óptima en una red.

3- Ruido y fórmulas para los diferentes tipos de medios físicos.

4- DNS: delegación inversa de subredes.

5- Datos fuera de banda (URG).

2- MTU óptima en una red.

(VER: por óptima se puede entender como “la mejor para cada situación”, y puede haber casos en que se necesite que sea grande)
(VER: MTU de red dice, no habla del trayecto entre varias redes. Puede estar mal la respuesta anterior)
(VER: tal vez sea el MSS, no MTU)

Cada tecnología a nivel de enlace tiene un tamaño máximo de trama que puede transmitir. Cualquier interfaz (en un router, host, etc) tiene una MTU característica que depende de su tipo. Por ejemplo: Ethernet, MTU de 1500 bytes.
El nivel de red es el encargado de preparar las tramas para que el nivel de enlace sea capaz de transmitirlas.
Hay casos en que si las tramas son demasiado grandes, pueden tardar demasiado en salir por la interfaz, generando un retardo en el envío. Por eso se prefiere, en algunos casos, que el nivel de enlace maneje tramas más chicas.
Pros y Contras de un MTU grande:

- Pros:
  - Mejora la eficiencia y reduce el overhead
  - Reduce la carga de CPU al procesar menos paquetes
- Contras:
  - Requiere más memoria (buffers)
  - En caso de pérdida de paquetes, la pérdida es mayor

El nivel de transporte TCP hace uso de PMTUD para evitar la fragmentación. UDP no lo hace.

3- Ruido y fórmulas para los diferentes tipos de medios físicos.

Teorema de Nyquist: Si la señal consiste en V niveles discretos, el teorema establece lo siguiente:

$$\text{Tasa de datos máxima} = 2B \log_2 V \text{ (bits/seg)}$$

Por ejemplo, un canal sin ruido de 3kHz no puede transmitir señales binarias (de dos niveles) a una velocidad mayor de 6000 bps.
Si hay ruido aleatorio (térmico) presente debido al movimiento de las moléculas en el sistema (la cantidad de ruido térmico presente se mide con base en la relación entre la potencia de la señal y la potencia del ruido), el principal resultado de **Shannon** expresa la tasa de datos máxima (o capacidad) de un canal ruidoso, cuyo ancho de banda es B Hz y cuya relación señal a ruido es S/N, dada por:

$$\text{Número máximo de bits/seg} = B \log_2 (1 + S/N)$$

Esto nos indica las mejores capacidades que pueden tener los canales reales.

5- Datos fuera de banda (URG).

La cabecera TCP contiene una opción de flag “URGENTE” y un campo de puntero a datos urgentes (la posición del segmento donde terminan esos datos, porque siempre se ponen en el inicio del segmento). Los datos urgentes se entregan a la aplicación sin esperar a que esta ejecute un “read”. Los datos urgentes se adelantan a cualesquiera de los datos que hubiera en el buffer. Por ejemplo el comando “Ctrl + c” que indica la interrupción de un programa en loop en una sesión telnet

---

Final 6)

1- Indique 2 desventajas de la tecnología 10BaseT frente a la tecnología del cable coaxial (10Base5 o 10Base2).

2- ¿Por qué si los protocolos MAC alcanzan el 95% de uso del ancho de banda del canal, se prefirió utilizar paso de testigo para aplicaciones de tiempo real?

3- En qué casos utilizaría un puente, un repetidor o un switch? ¿Por qué?
   a) Interconectar 2 redes de diferentes tecnologías.
   b) Mejorar el rendimiento de una red saturada.
   c) Conectar un enlace de fibra de 500m a una red 10BaseT.
   d) Interconectar 5 redes con mucho tráfico.
   e) Unir 2 redes cableadas con 10Base2 de 10m de cable C/1.

1- Indique 2 desventajas de la tecnología 10BaseT frente a la tecnología del cable coaxial (10Base5 o 10Base2).
Una desventaja de la tecnología 10baseT (cable UTP) frente a 10base5 y 10base2 (coaxial) es la distancia máxima de los segmentos: 100m frente a 500 y 185 metros respectivamente.

El cable coaxial tiene una inmunidad alta a las interferencias por lo que tiene muy pocas pérdidas.
Otra importante desventaja de 10baseT es que la instalación debe estar conectada mediante hubs, lo que puede resultar costoso, especialmente para grandes redes.

3- En qué casos utilizaría un **puente**, un **repetidor** o un **switch**? ¿Por qué?

a) Interconectar 2 redes de diferentes tecnologías.
b) Mejorar el rendimiento de una red saturada.
c) Conectar un enlace de fibra de 500m a una red 10BaseT.
d) Interconectar 5 redes con mucho tráfico.
e) Unir 2 redes cableadas con 10Base2 de 10m de cable c/u

a) Para interconectar 2 redes con tecnologías diferentes utilizaría un **puente** porque permite la interconexión transparente a nivel de enlace (MAC) a pesar de la diferencia de protocolos. Lo elijo por sobre el switch porque un puente tiene menos puertos (de 2 a 6). De todas maneras, actualmente están en desuso.
b) Se puede mejorar el rendimiento de una red saturada utilizando un **switch** que divida el dominio de colisiones en subgrupos.
c) Como necesitamos conectar dos enlaces con tecnologías diferentes, y son sólo 2, podemos utilizar un **puente**
d) Este es un caso que necesita un **switch** porque va a controlar el tráfico impidiendo que los paquetes que son dirigidos a un host en la propia red viajen hacia todas las demás
e) Un **repetidor** permite expandir la longitud de un segmento, en este caso podría utilizarse uno

---

Final 7)

1- a) Describa las funciones de un puente, hub, switch, indicando en qué capa del modelo OSI opera cada uno. Respuesta completa
b) Explique en detalle todo lo que pasa en los siguientes casos, donde se supone que la condición inicial es un switch que recién se enciende y que se utiliza un protocolo que no requiere el uso de ARP. Suponga que los eventos se suceden uno a continuación del otro y sin volver a condiciones iniciales.
a) A envía una trama a B.
b) C envía una trama de difusión.
c) D envía una trama a B.

2- Una vez que un datagrama ha sido fragmentado, explique ventajas y desventajas de reensamblarlo una vez que se ha atravesado la red con baja MTU que motivó en fragmentación, en lugar de recien reensamblarlo en destino. ¿Cuál de estas alternativas es la más usada en la práctica?

~~3- Algoritmo por el cual un host decide si debe realizar una ED o un ruteo de un datagrama IP.~~

~~4- 2 ethernet conectadas a 2 puentes, ¿qué sucede?~~

~~5- Algoritmo que se usan para determinar en qué momento se debe retransmitir un segmento TCP.~~

2- Una vez que un datagrama ha sido fragmentado, explique ventajas y desventajas de reensamblarlo una vez que se ha atravesado la red con baja MTU que motivó en fragmentación, en lugar de recien reensamblarlo en destino. ¿Cuál de estas alternativas es la más usada en la práctica?

Existen dos estrategias opuestas para recombinar los fragmentos de vuelta en el paquete original:

1- La fragmentación es transparente para redes que, por tener MTU lo suficientemente grande, el paquete puede pasar sin fragmentar. Esto lo hacen posible los routers que reenzamblan los paquetes luego que los fragmentos atravesaron la red con bajo MTU. De esta manera se ha hecho transparente el paso a través de la red de paquete pequeño. Las redes subsecuentes ni siquiera se enteran de que ha ocurrido una fragmentación.

2- La otra estrategia de fragmentación es abstenerse de recombinar los fragmentos en los enrutadores intermedios. Una vez que se ha fragmentado un paquete, cada fragmento se trata como si fuera un paquete original. Los enrutadores pasan los fragmentos y la recombinación ocurre sólo en el host de destino.

La mayor desventaja en 1 es el rendimiento bajo, ya que cada router debe procesar cada fragmento para reensamblarlo. Además, las rutas están restringidas porque todos los paquetes deben salir por el mismo router para poder reesamblarlos; esto baja el desempeño. También se considera que este trabajo puede ser un desperdicio, ya que puede ocurrir que el paquete deba atravesar una serie de redes de bajo MTU, y esto implique repetir el proceso varias veces.

La principal ventaja de la fragmentación no transparente es que los enrutadores tienen que trabajar menos. Pero la desventaja es que hay una mayor sobrecarga de encabezados y que la fragmentación representa una carga costosa para los hosts. De todas formas, esta es la opción más utilizada en la práctica.

La solución final (utilizada actualmente para internet) es NO FRAGMENTAR. Para lograrlo se utiliza PMTUD. La ventaja del descubrimiento de MTU de la ruta es que ahora la fuente sabe la longitud del paquete que puede enviar, no necesita fragmentar.

La desventaja del descubrimiento de MTU de la ruta es que puede haber retardos iniciales adicionales sólo por enviar un paquete.

Final 8)

1- ¿Cómo puede un switch prevenir un espionaje?

2- Explique el arranque lento.

~~3- Qué pasa si un router no puede transferir un paquete que llegó con el bit no fragmentado activado.~~

~~4- ¿Podría usarse ARP para IPV6? ¿Habría que hacer cambio de protocolo?~~

5- Escriba cómo funciona BOOTP.

1- ¿Cómo puede un switch prevenir un espionaje?

Ataques al switch - desbordamiento de tabla CAM
La tabla CAM se llena a partir de las direcciones de origen de los paquetes. Esta tabla tiene un tamaño grande, pero limitado. Cuando se llena, el switch empieza a descartar direcciones comenzando por las más antiguas, y si luego recibe un paquete cuyo destino no está en la tabla, empieza a difundir por inundación (distribuye todo a todas las interfaces). Básicamente, se convierte en un hub.
El ataque aprovecha esta funcionalidad enviando miles de paquetes por segundo, todos con MACs distintas (en ningún momento nadie verifica la unicidad entre la MAC y la IP, por lo que es posible enviar paquetes desde un host con la MAC cambiada), saturando la tabla CAM y provocando que se borren las direcciones conocidas. Esto produce que el switch propague todo por inundación (todos reciben todo) y así el "espía" puede escuchar todo el tráfico de la red.
La solución es configurar el switch para que no permita más de una (o unas pocas) dirección MAC por interfaz. Si detecta que por una interfaz se supera el límite, bloquea el puerto.

Ataque de Spanning Tree
Los switches están expuestos a ataques porque los mensajes (BPDUs) con que se comunican entre ellos no tienen ninguna capa de seguridad. Cualquiera podría enviar ese tipo de mensajes y cambiar el funcionamiento de la red, por ejemplo, cambiando el enrutador de raíz a sí mismo para canalizar todo el tráfico a su dispositivo y así "espiar".
Para evitar este tipo de ataques, se pueden configurar los switches para que no admitan BPDUs que provengan de un host que no sea otro switch, configurando cuáles puertos son los que interconectan directamente con otro switch. Si hay un mensaje que proviene de un puerto que no está configurado, bloquea el puerto (**BPDUs Guard**)
Otra posibilidad es filtrar los BPDUs: aceptar todos los mensajes excepto aquellos que intenten cambiar la raíz del árbol (**Root Guard**).

También existen los "ataques de Spoofing". Se hacen utilizando una vulnerabilidad en la comunicación con mensajes ARP. El "intruso impostor" participa en la red con una identidad falsa, haciéndose pasar por otro host .
Una solución es utilizar la tabla "DHCP binding table" (hubo que construirla previamente - DHCP Snooping) en el switch y compararla con de los ARP request/reply buscando incoherencias. En caso que se presente una diferencia, se bloquea el puerto. Hay casos en los que hay puertos "trust" que se deben configurar para pasar este filtro.

2- Explique el arranque lento.
El arranque lento es una técnica para evitar congestionamiento en TCP. Se aplica cuando se inicia una conexión o luego de recuperarse de un congestionamiento.
Funcionamiento:
1. el TCP inicia la ventana de congestionamiento en 1, envía un segmento inicial y espera
2. cuando llega el ACK, aumenta la ventana de congestionamiento a 2, envía 2 segmentos y espera
3. cuando llegan los 2 ACK, la ventana se aumenta a 4, envía 4 y espera.

4. luego con 8… y así sucesiva y lentamente va creciendo la ventana de congestionamiento hasta llegar al tamaño de ventana del receptor

5- Escriba cómo funciona BOOTP.

Es uno de los protocolos de resolución inversa de direcciones (encontrar IPs a partir de MACs). La resolución inversa de direcciones permite una gestión centralizada de las configuraciones, siempre se apoya en un servidor (escuchando y respondiendo pedidos de este tipo) que almacena las correspondencias MAC-IP.
Funcionamiento:

1. Cuando el cliente arranca envía un "**BOOTP Request**" a la dirección broadcast poniendo como dirección de origen 0.0.0.0 (aún no sabe cuál es su ip)
2. El servidor recibe el mensaje, busca en su tabla la MAC del solicitante y si la encuentra se prepara para responder con un "**BOOTP Reply**" con la correspondiente IP
3. Como el servidor no puede enviar la respuesta por unicast porque su ARP cache no tiene una entrada con la IP del cliente (mandar un ARP Request no sirve porque el cliente no sabe cuál es su IP), manda el "**BOOTP Reply**" con la respuesta en broadcast también
   Hay otra alternativa donde el servidor tiene privilegios de completar la ARP cache para introducir una nueva entrada y responder directamente al cliente

Los mensajes BOOTP Request y Reply se envían dentro de paquetes IP/UDP (por lo tanto pueden atravesar routers, entonces servidor y cliente pueden estar en distintas redes). Con toda la capacidad que tiene un paquete IP para poner información, BOOTP puede suministrar los parámetros:

- Dirección IP del cliente
- Máscara de subred
- Dirección/es de: router, servidor DNS, etc
- Cualquier otro ya que tiene un mecanismo para sumar opciones

---

Final 9)
"Final comunicaciones 27/12/13"

1- Se quiere aumentar la velocidad en un enlace que ya opera al límite de velocidad establecido por Shannon. Qué parámetro/s se debe modificar:

a) Aumentar la potencia
b) Aumentar la cantidad de bits por símbolo
c) Aumentar la velocidad de transmisión de símbolos
d) Disminuir la potencia
e) Disminuir la cantidad de bits por símbolo
f) Disminuir la velocidad de transmisión de símbolos
g) a + b
h) a + b + c
i) b + c
j) y hay había muchas otras combinaciones que no me las acuerdo

Justificar la/s eleccion/es. (Elegir un item mal te anulaba todo el ejercicio)

2- Preguntas varias:

a) Se quiere conectar una 10baseT y una 100baseT, qué dispositivos utilizamos y en qué capa opera. (suponer que tienen la misma MTU)
b) Cambiaría la respuesta de a) si se usan MTU distintas?
c) Misma pregunta que a) pero con una 10baseT y un Token Ring
d) 2 ventajas de UTP cat 6 frente a la fibra óptica en interiores
e) 2 diferencias entre ipv4 y ipv6 (que no sea el tamaño de las direcciones)
f) Cómo se hace broadcast en ipv6

3- ICMP redirect
a) Por que este tipo de mensajes no sirve si el host origen y el router que genera el menaje no están en la misma red capa 2
b) Discutir sobre los usos de este mensaje en una red punto a punto

~~4- Fragmentación completa en ipv4~~

~~5- Ventana tonta~~

1- Se quiere aumentar la velocidad en un enlace que ya opera al límite de velocidad establecido por Shannon (B log2 (1 + S/N)). Qué parámetro/s se debe modificar. Justificar la/s eleccion/es:
a) Aumentar la potencia
b) Aumentar la cantidad de bits por símbolo
c) Aumentar la velocidad de transmisión de símbolos
d) Disminuir la potencia
e) Disminuir la cantidad de bits por símbolo
f) Disminuir la velocidad de transmisión de símbolos
g) a + b
h) a + b + c
i) b + c
...

Pienso que la respuesta es **H**:
Podemos aumentar la velocidad aumentando la tasa de símbolos por segundo (c) y aumentando la cantidad de bits por símbolos. Pero esto implica la necesidad de aumentar la potencia de transmisión para aumentar la relación señal a ruido.

Final 10)
Octubre 2020

1- En qué capas coinciden exactamente los modelos OSI y el TCP/IP? Explique la función de estas capas.

2- Explique cómo funciona, y los campos de la cabecera IP involucrados en la fragmentación en ambas versiones de IP.

3- Explique brevemente el problema del síndrome de la ventana tonta provocada por el EMISOR.

4- Explique la función de los campos "serial", "refresh", "retry" y "expiry" del registro SOA de una zona DNS. Qué tipo de problemas evita el uso de este último ?

5- El comando iptables de Linux permite contar con un firewall del tipo stateful. De al menos 1 ejemplo de regla que utilice esto y explique para qué sirve.

INFO ADICIONAL

La capa de red ofrece un servicio no orientado a la conexión, por lo que cada paquete puede llegar por caminos diferentes y en cualquier orden. En cambio TCP es orientado a la conexión (UDP no) y mantiene la conexión y el orden de los segmentos (a partir de la numeración que les asigna) aunque no hayan llegado en orden

---

ENTREGA CONFIABLE

En el nivel más bajo, las redes proporcionan una entrega de paquetes no confiable (pueden perderse o destruirse por errores de transmisión, por ejemplo). La mayor parte de los protocolos confiables utilizan una técnica conocida como “Acuse de recibo positivo con retransmisión”:

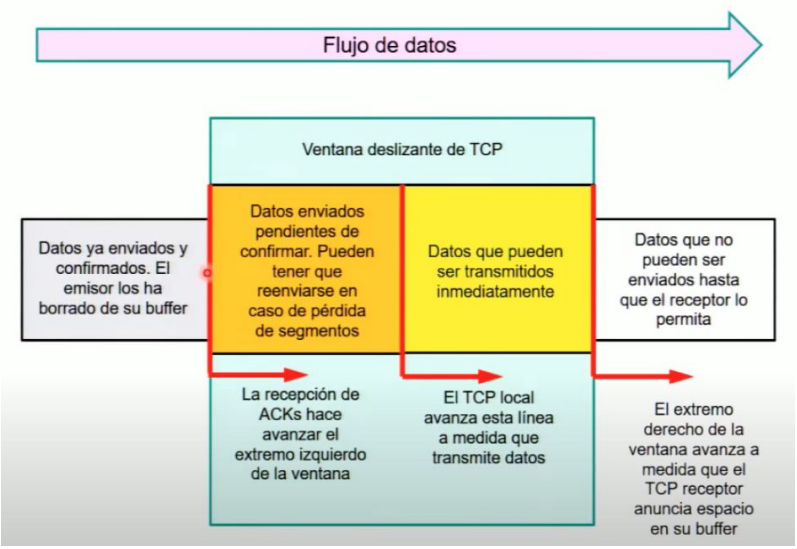
- el transmisor envía un paquete
- el receptor se comunica con el origen y le envía un mensaje de acuse de recibo (ACK) conforme recibe los datos
- el transmisor guarda un registro de cada paquete que envía y espera un acuse de recibo antes de enviar el siguiente paquete (si no recibe después de un tiempo, reenvía el paquete [tiene un timer])

Puede ocurrir que los paquetes se dupliquen (caso de canales de comunicación lentos) porque se demora la respuesta ACK. Para eso al enviar se asigna a cada paquete un número y los ACK contienen esos números para confirmar cada paquete.

VENTANA DESLIZABLE (control de flujo)

Especialmente en conexiones lentas, se utiliza la función de ventana deslizable para mejorar la eficiencia de los envíos/confirmación. Consiste en enviar varios segmentos en ráfaga sin necesidad de esperar el ACK de uno para enviar el siguiente. Los acuse de recibo indican luego cuántos bytes se recibieron, confirmando así los segmentos que llegaron.

El mecanismo TCP de ventana deslizable opera a nivel de octeto, no a nivel de segmento ni de paquete. Los octetos del flujo de datos se numeran de manera secuencial, y el transmisor guarda **tres** apuntadores asociados con cada conexión. Los apuntadores definen una ventana deslizable. El primer apuntador marca el extremo izquierdo de la ventana, separa los octetos que ya se enviaron y confirmaron. Un segundo apuntador, que marca el extremo derecho de la ventana, define el octeto más alto en la secuencia que se **puede** enviar antes de recibir más acuses de recibo. El tercer apuntador señala la frontera dentro de la ventana que separa los octetos que ya se enviaron de los que todavía no se envían.



Cada acuse de recibo, que informa cuántos octetos se recibieron, contiene un aviso de ventana que especifica cuántos octetos adicionales de datos está preparado para aceptar el receptor. El mecanismo de ventana deslizable, permite que el tamaño de la ventana varíe. Los ACKs contienen un aviso de ventana que especifica al transmisor si aumentar o disminuir el tamaño de la ventana de acuerdo a su capacidad de buffer. De esta manera proporciona el control de flujo.

El protocolo recuerda qué paquetes tienen ACKs y mantiene un timer para los que no. Si se pierde un paquete (se terminó el timer), el transmisor reenvía el paquete.

SACK (Selective ACK)
Al principio TCP requería que los ACK fueran acumulativos, de forma que si se perdía un segmento no se podía enviar ACK de los siguientes, aunque hubieran llegado bien. Esto significaba que cuando se perdía un segmento se tenía que reenviar todo a partir de ese punto. En un RFC posterior se estableció los SACK que permiten acusar el recibo de segmentos no consecutivos.
Para usarlo tiene que estar activada la opción ”SACK-permitted”, que se negocia al inicio de la conexión entre los extremos.

GLUE RECORDS
“Los **Glue Records** son necesarios para permitir que una consulta de DNS para el subdominio devuelva una referencia que contenga tanto el nombre del servidor de nombres como su dirección IP”

Estrictamente hablando, los **Glue Records** (la dirección IP del servidor de nombres definido mediante un A o AAAA RR) solo se requieren para cada servidor de nombres que se encuentre **dentro del dominio o la zona** para la que es un servidor de nombres. La respuesta a la consulta (la referencia) debe proporcionar tanto el nombre como la dirección IP de los servidores de nombres que se encuentran dentro del dominio que se consulta.

Vamos a personificar el proceso para encontrar la dirección correcta:

P: "Estoy tratando de encontrar www.ejemplo.com. ¿Cuál es la dirección IP?"

R: "No tengo la dirección. Busca los servidores DNS de este dominio."

P: "Está bien ejemplo.com. ¿Cuales son tus servidores DNS?"

R: "ns1.ejemplo.com"

P: "De acuerdo. ¿Cuál es la dirección IP de ns1.ejemplo.com?"

R: "No tengo la dirección. ns1.ejemplo.com es un subdominio de ejemplo.com. Busca los servidores DNS de este dominio."

P: "Vale. ¿Cuál es el servidor de DNS de ejemplo.com?"

R: "ns1.ejemplo.com"

P: "¡Argh! Me acabas de indicar esta dirección. ¿Qué hago?"

¡Aquí está el problema! Si el host "ns1.ejemplo.com" pertenece al dominio que estamos buscando, ¿cómo vamos a resolver su IP?
La solución son los Glue Records. De esta forma el registry crea entradas "A" en sus servidores para que al intentar resolver ns1.ejemplo.com tengamos su IP.

Ejercicio de Firewall – 2017

```
#! /bin/sh
# Definición de variables
F=/sbin/iptables
LAN=10.23.0.0/16
DMZ=10.1.1.0/29
WEBSRV=10.1.1.2
RELAY=10.1.1.3
MAILSRV=10.10.1.3
DBSRV=10.10.1.2
IF_LAN=eth2
IF_SRV=eth1
IF_DMZ=eth3
IF_EXT=eth0
IP_EXT=200.123.131.112
ADM_PC=10.23.23.5

# Valores por omisión de las cadenas
# Deniego entrada y salida por omisión.
# (no hace falta poner "-t filter" porque lo asume por omisión)

$F -P INPUT DROP
$F -P OUTPUT DROP
# Enunciado 9
$F -P FORWARD DROP

######### tabla filter ############
##### cadenas INPUT y OUTPUT #####

# Rechazo "conexiones" inválidas y permito respuestas a conexiones permitidas
for i in INPUT OUTPUT ; do
$F -A $i -m state –state INVALID -j DROP
$F -A $i -m state –state RELATED,ESTABLISHED -j ACCEPT
done

# A partir de aquí, solo quedan las conexiones en estado "NEW" para las cadenas
INPUT y OUTPUT

# Enunciado 8
# Acceso SSH para el administrador (utilizo -i para prevenir spoofing de direcciones
desde otra red)
$F -A INPUT -i $IF_LAN -s $ADMIN_PC -p tcp --dport 22 -j ACCEPT

# Como puse -P DROP, no sería necesario denegar todo el resto explicitamente
#$F -A INPUT -j DROP

# Consultas DNS
$F -A OUTPUT -d $RELAY -p tcp --dport 53 -j ACCEPT
$F -A OUTPUT -d $RELAY -p udp --dport 53 -j ACCEPT

# No se aceptan más paquetes en la cadena OUTPUT (comando -P de más arriba)

##### cadena FORWARD #####
```

```
# Otra vez las reglas de estado. Conviene ponerlo al principio pues las reglas que
contienen ESTABLISHED y RELATED son las que más paquetes matchearán
$F -A FORWARD -m state –state INVALID -j DROP
$F -A FORWARD -m state –state RELATED,ESTABLISHED -j ACCEPT

# Enunciado 1
# Nota: Los puertos se pueden poner usando la denominación del /etc/services además
de numéricas
$F -A FORWARD -i $IF_LAN -s $LAN -d $MAILSRV -p tcp \
      -m multiport --dport pop3,pop3s,domain,imap,imaps,smtp,smtps -j ACCEPT
$F -A FORWARD -i $IF_LAN -s $LAN -d $RELAY -p udp --dport 53 -j ACCEPT
$F -A FORWARD -i $IF_LAN -s $LAN -d $DBSRV -p tcp --dport 443 -j ACCEPT

# el "y nada más de las redes internas" del enunciado 1 queda cumplido
con el -P
# Enunciado 5 (ésto solo no alcanza, tambien hay que natear – ver abajo)
$F -A FORWARD -i $IF_LAN -s $LAN -o $IF_EXT -j ACCEPT

# Enunciado 2
# acceso a DNS por udp y tcp
$F -A FORWARD -i $IF_SRV -s $MAILSRV -d $RELAY -p udp --dport 53 -j ACCEPT
$F -A FORWARD -i $IF_SRV -s $MAILSRV -d $RELAY -p tcp -m multiport \
      --dport 53,465 -j ACCEPT

# Enunciado 3
$F -A FORWARD -i $IF_DMZ -s $RELAY -d $MAILSRV -p tcp --dport 465 -j ACCEPT

# Enunciado 4
$F -A FORWARD -i $IF_DMZ -s $WEBSRV -d $DBSRV -p tcp --dport 3306 -j ACCEPT

# Enunciado 6
# dns por udp desde el exterior
$F -A FORWARD -i $IF_EXT -d $RELAY -p udp --dport 53 -j ACCEPT

# acceso a mail y dns por tcp
$F -A FORWARD -i $IF_EXT -d $RELAY -p tcp -m multiport --dports 53,25 -j ACCEPT

# acceso a web
$F -A FORWARD -i $IF_EXT -d $WEBSRV -p tcp -m multiport --dports 80,443 -j ACCEPT

# Enunciado 7
for i in tcp udp ; do
$F -A FORWARD –i $IF_DMZ -o $IF_EXT -p $i --dport 53 -j ACCEPT
done
$F -A FORWARD -i $IF_DMZ -s $RELAY -o $IF_EXT -p tcp --dport 25 -j ACCEPT

########## TABLA nat #############
# Enunciado 6
$F -t nat -A PREROUTING -d $IP_EXT -p tcp –m multiport --dports 25,53 \
      -j DNAT --to $RELAY
$F -t nat -A PREROUTING -d $IP_EXT -p udp --dport 53 \
      -j DNAT --to $RELAY
$F -t nat -A PREROUTING -p tcp -m multiport --dports 80,443 \
      -j DNAT --to $WEBSRV
```

```
# Para que se cumplan los Enunciados 5 y 7, además de permitir el acceso, debemos
natear.

#### Cadena POSTROUTING ####
# en esta cadena no podemos usar -i ...

$F -t nat -A POSTROUTING -s $LAN -o $IF_EXT -j SNAT --to $IP_EXT
$F -t nat -A POSTROUTING -s $DMZ -o $IF_EXT -j SNAT --to $IP_EXT
```

Ejercicio Firewall - 2018

```
# definición de variables
F=/sbin/iptables
IF_EXT=eth2
IF_DMZ=eth1
IF_LAN=eth0
DMZ=181.16.1.16/28
LAN=10.0.1.0/24
IP_EXT=200.3.1.3
PROXY=181.16.1.19
WEB=181.16.1.18
ADMIN=10.0.1.22

# políticas por defecto
$F -P INPUT DROP
$F -P OUTPUT DROP
$F -P FORWARD DROP

# Descartamos paquetes extraños
$F -A INPUT -m state --state INVALID -j DROP
$F -A OUTPUT -m state --state INVALID -j DROP
$F -A FORWARD -m state --state INVALID -j DROP

# Permitimos conexiones establecidas y relacionadas
$F -A INPUT -m state --state ESTABLISHED, RELATED -j ACCEPT
$F -A OUTPUT -m state --state ESTABLISHED, RELATED -j ACCEPT
$F -A FORWARD -m state --state ESTABLISHED, RELATED -j ACCEPT

# punto 1
$F -A INPUT -i $IF_LAN -s $ADMIN -p tcp --dport ssh -j ACCEPT
$F -A FORWARD -i $IF_LAN -s $ADMIN -d $PROXY -p tcp --dport ssh -j ACCEPT
$F -A FORWARD -i $IF_LAN -s $ADMIN -d $WEB -p tcp --dport ssh -j ACCEPT

# punto 2
$F -A FORWARD -i $IF_LAN -s $LAN -d $PROXY -p tcp -m multiport /
      --dport 3128,dns -j ACCEPT
$F -A FORWARD -i $IF_LAN -s $LAN -d $WEB -p tcp -m multiport /
      --dport http,https,dns -j ACCEPT
$F -A FORWARD -i $IF_LAN -s $LAN -d $PROXY -p udp -m multiport /
      --dport dns -j ACCEPT
$F -A FORWARD -i $IF_LAN -s $LAN -d $WEB -p udp -m multiport /
      --dport dns -j ACCEPT

# “No se permiten más accesos que los anteriores” = si bien tengo DROP por default,
# conviene para diagnóstico interno:
$F -A FORWARD -i $IF_LAN -o $IF_DMZ -j REJECT

# punto 3
$F -A FORWARD -i $IF_LAN -p tcp --dport http,https -j REJECT
$F -A FORWARD -i $IF_LAN -s $LAN -j ACCEPT

# punto 4
$F -A FORWARD -i $IF_DMZ -s $DMZ -o $IF_LAN -d $LAN -j REJECT
```

```
$F -A FORWARD -i $IF_DMZ -s $DMZ -o $IF_EXT -d $IP_EXT /
      -p tcp -m multiport --dport http,https,dns -j ACCEPT
$F -A FORWARD -i $IF_DMZ -s $DMZ -o $IF_EXT -d $IP_EXT /
      -p udp --dport dns -j ACCEPT

# punto 5
$F -A FORWARD -i $IF_EXT -s $IP_EXT -o $IF_DMZ -d $WEB /
      -p tcp -m multiport --dport http,https,dns -j ACCEPT
$F -A FORWARD -i $IF_EXT -s $IP_EXT -o $IF_DMZ -d $PROXY /
      -p tcp --dport dns -j ACCEPT
$F -A FORWARD -i $IF_EXT -s $IP_EXT -o $IF_DMZ -d $WEB /
      -p udp --dport dns -j ACCEPT
$F -A FORWARD -i $IF_EXT -s $IP_EXT -o $IF_DMZ -d $PROXY /
      -p udp --dport dns -j ACCEPT

# punto 6
$F -A OUTPUT -o $IF_DMZ -d $PROXY -p tcp --dport 3128 -j ACCEPT

#### NAT ####
$F -t nat -A POSTROUTING -s $LAN -o $IF_EXT -j SNAT --to $IP_EXT

# No se natea los servidores de la DMZ porque tienen IPs públicas
```