Redes y Comunicaciones

Práctica 3 - Capa de transporte

Dudas:

- como sabe un host a que socket entregar los segmentos? TCP identifica el socket por IPdest, IPorig, Porig, Pdest, pero que pasa si dos programas se comunican al mismo servidor por el mismo puerto? quedaria igual la tupla? Los servidores web tienen sockets distintos por cada cliente, pero pueden recibir x el mismo puerto **Resuelto: segun wikipedia solo un proceso se puede asociar a una combinación IP-puerto usando un determinado protocolo**
- **Aca no hay NADA de control de congestión, pero esta en las diapositivas, creo que es importante**

Aclaraciones:

- Unidad de datos de la capa de transporte (PDUs)**: segmentos**
- **Tx** = transmisor; **Rx** = receptor

1. ¿Cuál es la función de la capa de transporte?

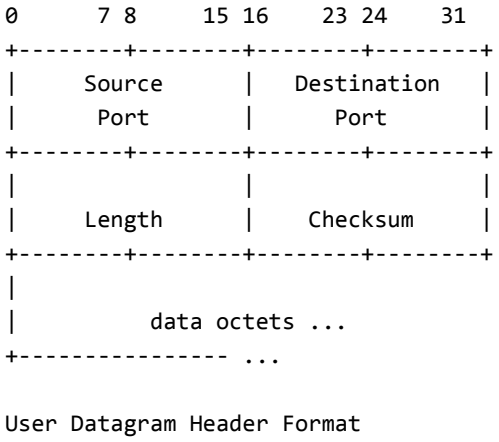
Proporciona directamente servicios de comunicación a los procesos de aplicación que se ejecutan en hosts diferentes. Amplía el servicio de entrega de la capa de red entre dos sistemas terminales a un servicio de entrega entre dos procesos de la capa de aplicación. **La capa de transporte entonces proporciona comunicación lógica entre *procesos* en distintos hosts mientras que la capa de red la proporciona comunicación lógica entre *hosts*.**

Es la que transporta los mensajes entre los puntos terminales de una red.

2. Describa la estructura del segmento TCP y UDP

UDP (User Datagram Protocol)

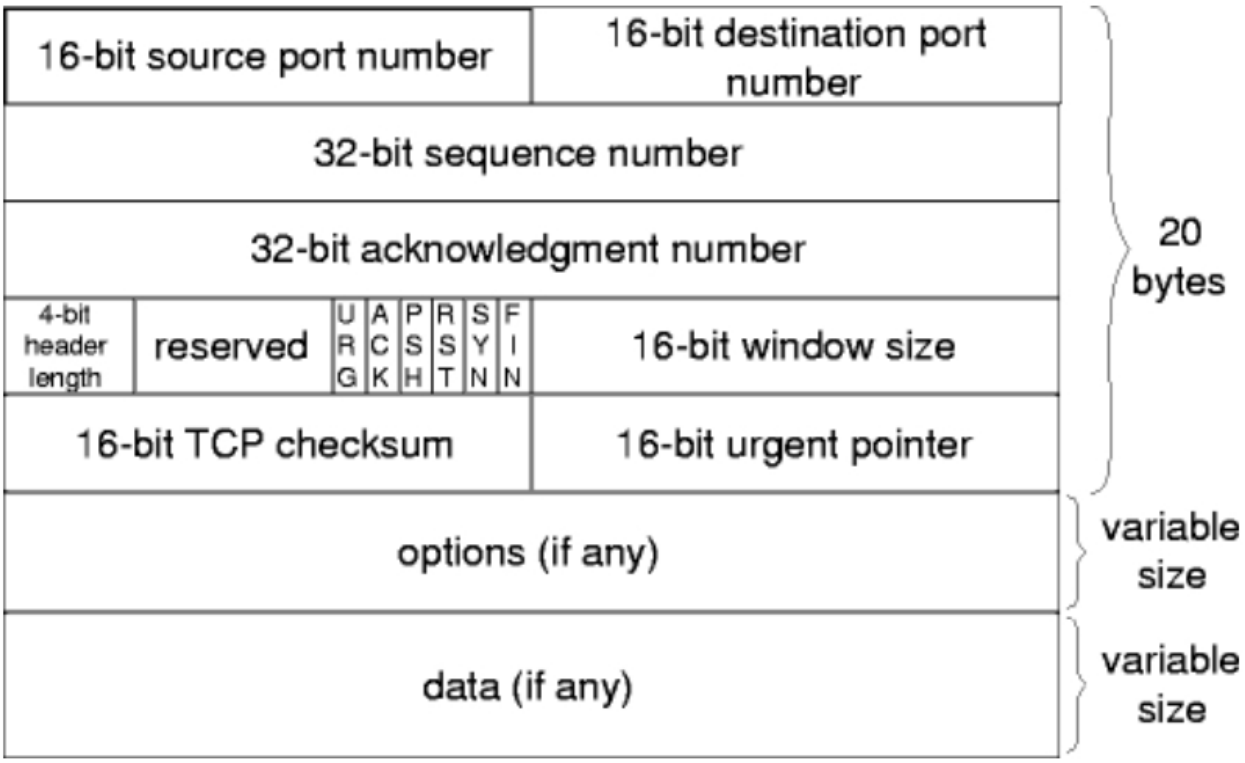
RFC 768



User Datagram Header Format

La cabecera de un segmento UDP solo tiene 4 campos y cada uno de ellos tiene 2 bytes: puerto de origen, puerto de destino, longitud y la suma de comprobación. Los números de puertos permiten dirigir el segmento al socket correspondiente, es decir, el socket UDP es identificado por puerto origen y puerto destino.

TCP (Transmission Control Protocol)



- Tanto el **campo de número de secuencia** como el **campo de reconocimiento** son usados para implementar un servicio de transferencias fiable.
- La **ventana de recepción** (16 bits) se usa para el control de flujo.
- El campo **header length** (long. de cabecera) de 4 bits indica específicamente la longitud de la cabecera en palabras de 32 bits (el campo opciones puede estar vacío y la cabecera puede tener 20 bytes)
- **Opciones** es opcional y de longitud variable. (no se q mas poner) q tiene? para q sirve?
- El **campo indicador** tiene 6 bits. El bit **ACK** se usa para avisar que lleva un *acknowledgement*. Los bits **RST**, **SYN** y **FIN** se usan para el establecimiento y cierre de conexiones. El bit **PSH** indica que el receptor deberá pasar los datos a la capa superior de forma inmediata. El campo **URG** se usa para indicar que hay datos en ese segmento que la entidad de la capa superior del lado emisor ha marcado como urgentes. Según dice el libro, en la vida real los bits PSH y URG no se usan, Andrés Rodriguez dijo que si.

3. ¿Cuál es el objetivo del uso de puertos en el modelo TCP/IP?

Conclusiones sacadas a partir de Wikipedia en inglés:

Un puerto es una estructura de software específica para un proceso o aplicación. Se asocia con la IP del host y el protocolo usado para la comunicación (generalmente TCP y UDP). Cada puerto se identifica con un número de 16 bits que va del 0 al 65535.

Los puertos son entonces estructuras de software que se usan para el direccionamiento de segmentos entrantes y salientes dentro de un host.

Práctica de AlternativaWEB:

El uso de puertos en el modelo TCP/IP permite la multiplexación de los paquetes que la capa de transporte recibe de los distintos procesos y pasa a la capa de Red en el emisor, y la demultiplexación de los segmentos recibidos desde la capa de Red en paquetes que se envían a los procesos adecuados en el receptor.

Si un terminal tuviese dos procesos haciendo solicitudes a un mismo servidor Web, ambos se comunicarán al port 80 -utilizado por el servidor Web en el host destino-. El servidor Web se forkea, es decir crea un nuevo proceso hijo para atender cada nueva solicitud (en este ejemplo tendría dos procesos) y así tener el proceso central escuchando el port 80. El port destino permite a la capa de Transporte del receptor entregar los paquetes (demultiplexacion) al servidor Web y no a otro proceso que esté corriendo en ese host (servidor). Del mismo modo, cada proceso del emisor tendrá asociado un port de origen diferente -comúnmente aleatorio-, permitiendo al servidor Web encaminar sus respuestas en dirección contraria a un proceso específico del host (cliente) que realizó las solicitudes.

En el caso que dos clientes (distintos, es decir, diferentes host) escojan un mismo port origen, el servidor podrá diferenciar a cada proceso utilizando la IP de origen, en el datagrama de capa de Red. Así, la tripla (port origen, port destino, IP origen) permite al host receptor encaminar los datos al socket correspondiente.

Multiplexación es la combinación de dos o más canales de información en un solo medio de transmisión usando un dispositivo llamado multiplexor. El proceso inverso se conoce como demultiplexación. Un concepto muy similar es el de control de acceso al medio.

4. Compare TCP y UDP en cuanto a:

A) Confiabilidad

TCP es más confiable en cuanto a la entrega de la información enviada ya que tiene sistemas de confirmación de entrega, mientras que UDP no.

Práctica de AlternativaWEB:
Asegura la entrega de todos los paquetes enviados, permitiendo mantener en el receptor el orden en que le fueron enviados (utilizando confirmación de recepción, timeout, reenvío de paquetes y números de secuencia) y detectar errores en la recepción (mediante checksum).

TCP: Si, utiliza establecimiento de conexión de 3 pasos
UDP: No, sin conexión, sólo ofrece el best effort de IP

B) Multiplexación

Práctica de AlternativaWEB:
Extender la comunicación host-host que ofrece IP permitiendo diferenciar los procesos en el receptor y en el emisor, de modo que varios procesos en un host se comuniquen con varios procesos en otros hosts, sin confundir los paquetes (utilizando direcciones IP y números de puerto que identifican a los procesos).

TCP: Socket (IP Orig, Port Orig, IP Dest, Port Dest)
UDP: Socket (IP Dest, Port Dest)

C) Orientado a la conexión

Práctica de AlternativaWEB:
Controlar la velocidad de transmisión del emisor para no sobrecargar la red.
TCP: SI
UDP: NO

D) Controles de congestión

Práctica de AlternativaWEB:
Controlar la velocidad de transmisión del emisor para no sobrecargar al receptor (según el tamaño de su buffer, ventana de recepción)
TCP: SI
UDP: NO

E) Utilización de puertos

Práctica de AlternativaWEB:
Los ports destino y origen son precisos tanto en TCP como en UDP por fines de multiplexación/demultiplexación, aunque UDP sólo precisa el port

TCP: Port Origen y Port Destino
UDP: Port Origen y Port Destino

5. La PDU de la capa de transporte es el segmento. Sin embargo, en algunos contextos suele utilizarse el término Datagrama, indique cuándo.

Práctica de AlternativaWEB:
Datagrama => PDU
Un datagrama es una cabecera + datos, tal que la cabecera ofrece la información necesaria para que los datos puedan ser encaminados y alcancen un destino.
Se suele hablar de Datagrama en referencia a los segmentos en el protocolo UDP -Protocolo de Datagrama de Usuario-. Sin embargo, como datagrama es el nombre formal de los PDU de la capa de Red -protocolo IP- es preferible evitar esta denominación.

6. Describa el saludo de tres vías de TCP

Para generar una conexión, TCP usa un saludo de tres vías:

1. **SYN**: se le envía al servidor un segmento con el flag SYN, el cliente setea el número de secuencia del segmento a un número aleatorio A.
2. **SYN-ACK**: el servidor responde con un SYN-ACK. El número de acknowledgement se setea a A+1 y para el número de secuencia del segmento se elige un número B.
3. **ACK**: finalmente el cliente responde al servidor con un ACK. El número de acknowledgement se setea a B+1. Con este paquete ya pueden viajar datos.

7. Investigue que es multicast (multidifusión). ¿Sobre qué protocolo de capa 4 funciona?

En el **enrutamiento por difusión**, la capa de red proporciona un servicio de entrega para un paquete enviado desde un nodo de origen a **todos los demás** nodos de la red.

En el **enrutamiento por multidifusión** permite a un único nodo de origen enviar una copia de un paquete a un **subconjunto de los restantes** nodos de la red.
En la multidifusión se enfrentan dos problemas, identificar a los receptores y cómo dirigir un paquete enviado a esos receptores.

Un paquete multidifusión se direcciona utilizando la dirección de direcciones, se utiliza un único identificador para el grupo de receptores. En internet este identificador único es una dirección IP de clase D, conocida como grupo multidifusión. La dificultad es que cada host tiene una dirección IP de unidifusión independiente de la dirección del grupo de multidifusión.

Protocolo IGMP: Opera entre un host y un router directamente conectado, IGMP proporciona los medios a

un host para informar a su router que una aplicación desea unirse a un grupo multidifusión especificado. IGMP opera entre un host y un router se requiere otro protocolo para coordinar a los router multidifusión. Por lo tanto, la multidifusión de la capa de red en Internet tiene dos componentes complementarios: IGMP y los protocolos de enrutamiento por multidifusión.

¿Se podría adaptar para que funcione sobre el otro protocolo de capa 4? ¿por-qué?

Solo funciona sobre UDP, en TCP no funciona porque el saludo de tres vías hace que sea una conexion 1 a 1 (unicast).

8. Utilizando el Live CD. Use el analizador de paquetes Wireshark para capturar los paquetes enviados y recibidos en cada uno de los siguientes casos. Para ello, arranque la captura antes de realizar cada una de las acciones indicadas:

A) Abra un navegador e ingrese a la URL: www.redes.unlp.edu.ar

Analice la secuencia de segmentos TCP que permiten la apertura del canal de comunicación por el cual posteriormente viajarán los mensajes HTTP intercambiados. ¿Con que nombre se conoce a dicha secuencia?

Saludo de 3 vias (TCP 3 - Way Handshake)

¿Qué flags se utilizan en cada uno de los segmentos intercambiados?

Primer paso: El cliente envía SYN en 1 al servidor.
Segundo paso: El servidor le responde al cliente con SYN en 1 y ACK en 1 (El ack es el número de byte que se esta esperando recibir)
Tercer paso: El cliente envía el ACK esperado por el Servidor.

¿Qué indica cada uno de estos flags?

El flag SYN sirve para iniciar una conexión. El flag ACK sirve para indicar que un segmento llegó correctamente.

B) Cierre el navegador:

Analice la secuencia de segmentos TCP que ocurren al hacerlo ¿Cuál es el

objetivo de éstos?

Hay una secuencia de envío de paquetes para el cierre de la conexión. El objetivo de estos es que cada host terminal le avise al otro que la conexión se va a cerrar y que el otro le responda con un ACK. Primero el Cliente le manda un FIN al Servidor, este responde con un FIN-ACK y finalmente el Cliente le responde con un ACK.

¿Qué flags se utilizan en cada uno de dichos segmentos?

Se utilizan los flags FIN y ACK.

¿Qué indica cada uno de estos flags?

El flag FIN indica que uno de los host terminales desea terminar la conexión. El ACK es para indicar que el mensaje con el FIN fue recibido.

C) Para este ejercicio debe usar tanto el navegador Epiphany como Iceweasel. Utilice Epiphany para ingresar a la URL: www.redes.unlp.edu.ar/ y seguidamente utilice Iceweasel para ingresar nuevamente a la URL: www.redes.unlp.edu.ar/

i. Observe la información de "Puerto Origen" y "Puerto destino" de cada una de las comunicaciones. En base a lo observado, responda ¿Es posible conectarse 2 veces en forma simultánea al mismo lugar? ¿Qué distingue una conexión de otra? Capture el tráfico de red si considera necesario para observar dicha información.

Si, es posible conectarse dos veces en forma simultánea al mismo lugar. Lo que distingue estas conexioness son los puertos a los que van dirigidos los paquetes, uno distinto para cada programa.

ii. Identifique lo observado en el punto anterior utilizando el comando netstat. Este comando puede ser utilizado para obtener diferentes como ser:

1. **netstat -nat (muestra todas las conexiones TCP del sistema)**
2. **netstat -nau (muestra todas las conexiones UDP del sistema)**
3. **netstat -natl (muestra de las conexiones que mostraría el comando "netstat -nat", las que estan es estado de LISTEN solamente).**
4. **netstat -natp (muestra además de lo que muestra el comando "netstat -nat", el proceso del sistema asociado). Para esto es necesario ser administrador o root.**

D) Desde la consola de root use el servicio tftp:

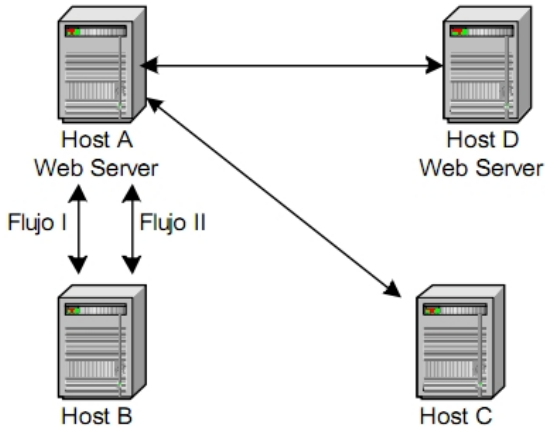
i. Ejecute “tftp localhost” y copie un archivo cualquiera desde su PC al servidor, a través de la opción put: “put captura.pcap” por ejemplo.

ii. Borre el archivo de su PC: “rm captura.pcap” y obtengalo ahora del servidor a través de la opción get: “get captura.pcap” por ejemplo.

E) ¿Qué diferencias encuentra en cuanto a mensajes intercambiados entre los puntos A, B respecto del punto D?

El handshake para el inicio de la conexión es distinto y el cierre de la conexión también. Una vez que el cliente envió una petición al servidor a su puerto bien conocido 69 el servidor responde por un nuevo puerto al cliente, no por el 69.

9. Dado el siguiente gráfico, complete los siguientes cuadros



Tener en cuenta:
- El Servicio Web corre en el puerto 80 en ambos Hosts.
- B y C acceden al servicio WEB del Host A.
- A accede al servicio WEB del host D.

Flujo I entre A y B (1er comunicación)

Origen es A	IP Origen A	Puerto Origen 80	IP Destino B	Puerto Destino W
Origen es B	IP Origen B	Puerto Origen W	IP Destino A	Puerto Destino 80

Flujo II entre A y B (2da comunicación. Es independiente de la primera)

Origen es A	IP Origen A	Puerto Origen 80	IP Destino B	Puerto Destino X
Origen es B	IP Origen B	Puerto Origen X	IP Destino A	Puerto Destino 80

Flujo III entre A y C (3er comunicación)

Origen es A	IP Origen A	Puerto Origen 80	IP Destino C	Puerto Destino Y
Origen es C	IP Origen C	Puerto Origen Y	IP Destino A	Puerto Destino 80

Flujo II entre A y D (4ta comunicación)

Origen es A	IP Origen A	Puerto Origen Z	IP Destino D	Puerto Destino 80
Origen es D	IP Origen D	Puerto Origen 80	IP Destino A	Puerto Destino Z

10.¿Qué es ARQ (Automatic Repeat Request)? ¿Qué capacidades requieren ser implementadas en los protocolos ARQ para detectar la presencia de errores en los datos?

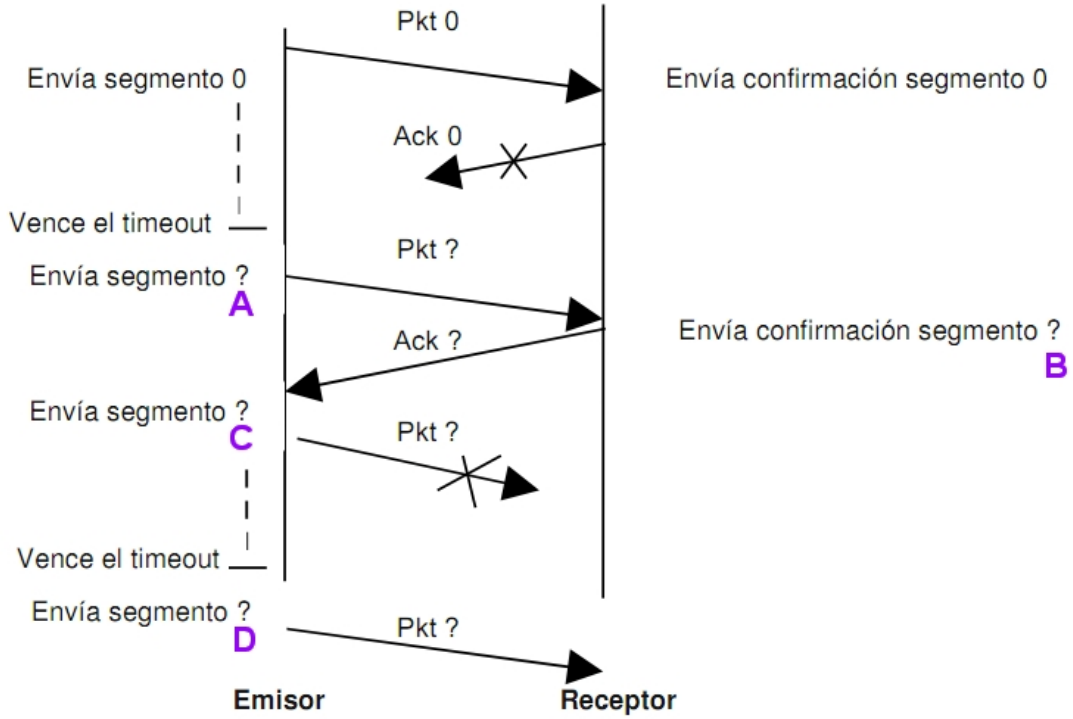
Wikipedia:

Automatic Repeat Request es un método para transmisión de datos que tiene control de errores y usa acknowledgements y timeouts. Si el cliente no recibe un acknowledgement antes del timeout reenvía el paquete.

Práctica de AlternativaWEB:

El protocolo ARQ permite la transferencia confiable de datos sobre medios no confiables,basándose en la confirmación de recepción de paquetes, la detección de errores y solicitudes de retransmisión.
Se debe implementar alguna técnica de detección de errores -campo checksum-. A su vez, el receptor debe responder a cada paquete con una señal ACK -recepción correcta- o NACK -recepción corrupta-. Si se devuelve un NACK o el paquete de respuesta está corrupto, el emisor retransmite el paquete. Si se trataba de un ACK corrupto, los datos habían llegado bien originalmente pero aún así el emisor duplica el paquete: para que el receptor pueda identificar paquetes nuevos de retransmisiones, se le agrega a los paquetes un número de secuencia. Otra posibilidad, es que el paquete de ACK o NACK nunca llegue al emisor, o que el paquete de datos original nunca llegue al receptor. Para solucionar estos dos inconvenientes, el emisor implementa un timer. Si se produce un timeout para un paquete enviado y no se ha recibido respuesta ACK ni NACK se asume que nunca llegó el paquete o nunca llegó al respuesta, y se retransmite.
Algunos protocolos ARQ son: Stop-And-Wait, Go-Back-N y Selective Repeat. La misma funcionalidad que se logra con ACK y NACK, puede lograrse sólo con ACK: si se recibe mal un paquete se envía un ACK con el número de secuencia del último paquete bien recibido, en lugar de un NACK. Cuando el emisor recibe un ACK duplicado, reenvía el segmento actual en Stop-and-Wait, o el segmento siguiente en Go-back-N y Selective Repeat.

11.Complete los (?) de la siguiente secuencia Stop and Wait:



A Envía paquete 0
B Confirma paquete 0
C Envia paquete 1
D Envia paquete 1

12. Explique la lógica de Go Back – N.

Práctica de AlternativaWEB:

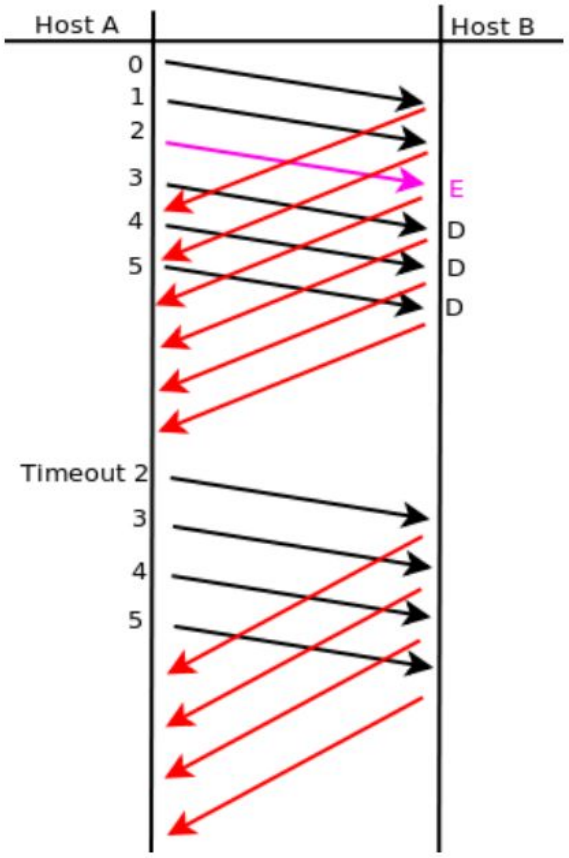
Un protocolo Go Back - N (GNB) es un protocolo ARQ con pipelining (es decir, soporta múltiples segmentos en tránsito pendientes de confirmación, implementando buffers en el emisor; el rango de los números de secuencia debe aumentarse), por esto el emisor puede transmitir varios paquetes sin esperar ningún reconocimiento, pero tiene restringido el número máximo, N, de paquetes no reconocidos en el entubado, llamado ventana del emisor. El emisor también tiene un timer para controlar el timeout.

La recepción del ACK de un segmento, confirma a todos los anteriores pendientes -acuse de recibo acumulado-. El vencimiento del timer para un segmento implica la retransmisión de todos los segmentos siguientes en la ventana. La ventana se desplaza a medida que se reciben ACK's.

El receptor envía el ACK sólo del segmento recibido correctamente de mayor número de secuencia en orden -es decir, que no le falte ninguno-. Puede generar acuses duplicados.

No precisa buffers, ya que descarta los segmentos fuera de orden y duplicados (al descartar re envía el ACK del mayor número de secuencia que recibió).

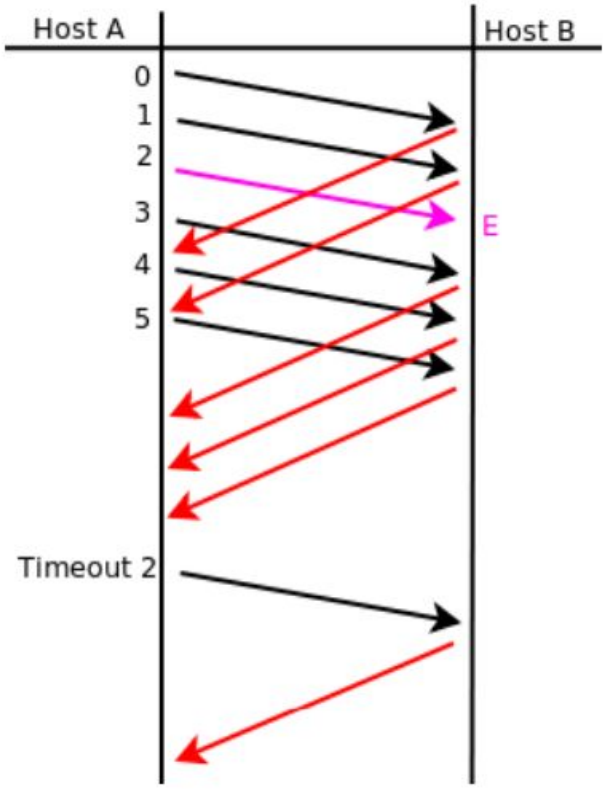
13.Suponiendo Go Back N; tamaño de ventana 4 y sabiendo que E indica que el mensaje llegó con errores y que D significa que el mensaje será descartado por llegar fuera de secuencia. Indique en el siguiente gráfico, la numeración de los ACK que el host B envía al Host A.



ACKs: 0, 1, 1, 1, 1, 1, 2, 3, 4, 5

14.Suponiendo Selective Repeat; tamaño de ventana 4 y sabiendo que E indica que el mensaje llegó con errores. Indique en el siguiente gráfico, la numeración

de los ACK que el host B envía al Host A.



ACKs: 0, 1, 3, 4, 5, 2

15.¿Qué restricción existe sobre el tamaño de ventanas en el protocolo Selective Repeat?

La ventana de Selective Repeat no puede ser más grande que la mitad de la cantidad de números de secuencia disponibles. Esto es para evitar la siguiente situación:
Celeste: enviados sin recibir ack
Rosa: recibidos y ACK enviado
Subrayado: ventana
SR=4
N= 5
Tx: **0 1 2 3 4 0 1 2 3 4**
Rx: **0 1 2 3 4 0 1 2 3 4**

Como se vé en el ejemplo, el tamaño de la ventana (4) es más que la mitad (2,5) y el problema que se genera es que el transmisor va a reenviar el paquete 0 pensando que no llegó y el receptor va a aceptarlo pensando que es el segundo 0 aunque en realidad es el primero.

16.Utilice el comando netstat durante la ejecución del ejercicio 6 (A,C y D) para determinar y completar la siguiente información:

A) Comando utilizado (debe incluir los argumentos utilizados con el comando netstat)

B) Protocolo de transporte utilizado

C) Puerto del servidor y nombre de la aplicación.

D) Puerto del o los clientes y nombre de la aplicación.

17.Investigue cómo funciona el protocolo de aplicación FTP teniendo en cuenta las diferencias en su funcionamiento cuando se utiliza el modo activo de cuando se utiliza el modo pasivo. ¿en qué se diferencian estos tipos de comunicaciones del resto de los protocolos de aplicación vistos?

FTP funciona a través de los puertos 20 (puerto de datos) y 21 (puerto de control). Una conexión FTP comienza con el cliente que reserva un puerto aleatorio y el puerto siguiente a ese (ej 1026 y 1027). A continuación contacta al puerto 21 del servidor FTP y este le responde. Hasta aca todo igual, pero continua distinto dependiendo de si es pasivo o activo:

- Activo: a este le gusta dar, así que agarra y por el puerto 20 le empieza a mandar paquetes, a los que el cliente responde con confirmaciones.
- Pasivo: en este sistema cuando el servidor respondió anteriormente al cliente le mandó un número de puerto aleatorio por el que estará escuchando para que la conexión la inicie el cliente, el cliente inicia la conexión y el servidor comienza la transferencia. **(O sea que no usa el puerto 20?)**

Estos dos modos existen por una medida de seguridad, los firewalls no suelen dejar conexiones entrantes, sólo aceptan las que fueron iniciadas por el cliente y de esta manera se evita este problema.

18.Utilizando el Live CD conéctese al servidor ftp utilizando el comando ftp ftp.redes.unlp.edu.ar utilizando los siguientes datos:

A) Nombre de usuario: lihuen

B) Password: lihuen

C) Pruebe la transferencia de un archivo cualquiera hacia y desde el servidor.

D) Utilizando Wireshark para obtener capturas de transferencias de archivos usando primero el modo activo y luego el modo pasivo.

19.Dada la siguiente captura del comando netstat.

```
lihuen:~# netstat -natp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:2628            0.0.0.0:*               LISTEN      4627/dict
tcp        0      0 127.0.0.1:718           0.0.0.0:*               LISTEN      4782/famd
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      4054/portmap
tcp        0      0 0.0.0.0:21              0.0.0.0:*               LISTEN      4721/pure-ftpd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      4587/sshd
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN      4706/master
tcp        0      0 127.0.0.1:45696         127.0.0.1:80            ESTABLISHED 6067/firefox-bin
tcp        0      0 127.0.0.1:45697         127.0.0.1:80            ESTABLISHED 6057/epiphany-brows
tcp        0      0 163.10.10.107:22        163.10.10.112:49302     ESTABLISHED 6397/sshd
tcp        0      0 0.0.0.0:110             0.0.0.0:*               LISTEN      4623/couriertcpd
tcp        0      0 0.0.0.0:143             0.0.0.0:*               LISTEN      4617/couriertcpd
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      4953/apache2
tcp        0      0 0.0.0.0:53              0.0.0.0:*               LISTEN      4522/named
tcp        0      0 0.0.0.0:953             0.0.0.0:*               LISTEN      4522/named
tcp        0      0 127.0.0.1:80            127.0.0.1:45697         ESTABLISHED 5223/apache2
tcp        0      0 127.0.0.1:80            127.0.0.1:45696         ESTABLISHED 5005/apache2
lihuen:~#
```

A) Que puede decir respecto de:

i. ¿Qué protocolo o protocolos de transporte se observan?

TCP

ii. ¿Cuántos puertos hay abiertos a la espera de posibles nuevas conexiones?

11

iii. ¿Cuántas conexiones hay establecidas en este momento?

5

iv. El cliente y el servidor de las comunicaciones HTTP (puerto 80), ¿residen en la misma máquina?

Si

v. El cliente y el servidor de la comunicación SSH (puerto 22), ¿residen en la misma máquina?

No, la dirección IP local es 163.10.10.107 y la del servidor es 163.10.10.112, son distintas máquinas

vi. Liste los nombres de todos los procesos que representan el lado del servidor de la comunicación.

apache2, sshd (solo conexiones establecidas)

vii. Liste los nombres de todos los procesos que representan el lado del cliente de la comunicación.

firefox-bin, epiphany-brows (solo conexiones establecidas)

20.¿Cúal es el puerto por defecto que utiliza?

i. Un servidor web: 80
ii. Un servidor SSH: 22
iii. Un servidor DNS: 53
iv. Un servidor web seguro: 443
v. Un servidor POP3: 110
vi. Un servidor IMAP: 143
vii. Un servidor SMTP: 25

Ordenados:

FTP-Data	20
FTP-Ctl	21
SSH	22
SMTP	25
DNS	53
HTTP	80
POP3	110
IMAP	143
HTTPS	443

viii. Analice el archivo /etc/services (Linux) o c:\windows\system32\drivers\etc\services (Windows) ¿Que información contiene?

El archivo /etc/services asocia un servicio con un puerto para un protocolo ej.

```
ssh             22/tcp                  # SSH Remote Login Protocol
ssh             22/udp
```