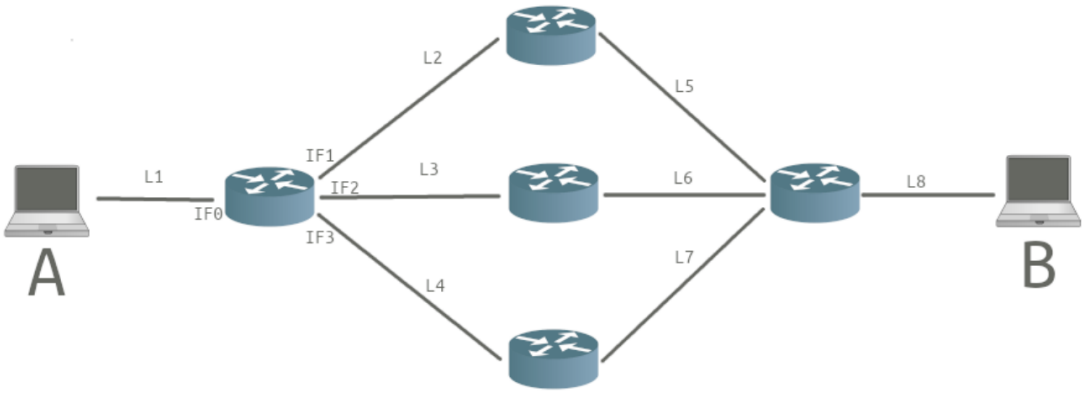
Modelo parcial

Latencia

Resolver

Se quiere calcular el RTT para medir la latencia entre dos host bajo la siguiente configuración:



Datos:
Packet Size = 1000 bytes

	L1	L2	L3	L4	L5	L6	L7	L8
Distancia	100 m	10 km	4 km	6 km	2 km	10 km	6 km	50 m
Ancho de Banda	10 Mbps	200 Mbps	200 Mbps	100 Mbps	100 Mbps	50 Mbps	200 Mbps	10 Mbps
Velocidad de Propagación	1.7×10^5 km/s	2×10^5 km/s	2×10^5 km/s	1.7×10^5 km/s	2×10^5 km/s	2×10^5 km/s	2×10^5 km/s	1.7×10^5 km/s

1 Mbps = 10^6 bits / seg

El RTT se debe calcular utilizando un segmento de prueba de tamaño **1000 Bytes**, y será el mismo para la ida y la vuelta.
Tener en cuenta la asimetría de caminos siendo:

Ruta A → B: $L_1 \rightarrow L_2 \rightarrow L_5 \rightarrow L_8$
Ruta B → A: $L_8 \rightarrow L_7 \rightarrow L_4 \rightarrow L_1$

Los tiempos de encolado y procesamiento son despreciables.
Detallar los pasos del cálculo obtenido y expresar la solución en milisegundos

Resolución

Primero notar que el camino es asimétrico, por lo tanto es necesario calcular el tiempo de A→B y luego de B→A. Luego recordando el concepto de latencia, se origina por 4 motivos:

- **Tiempo de inserción**: tiempo que tarda en introducirse el paquete al canal del enlace.
- **Tiempo de propagación**: tiempo que tarda en propagarse en el canal de enlace
- **Tiempo de procesamiento**: tiempo que tarda en el router en procesar el paquete y decidir el puerto de salida
- **Tiempo de encolado**: tiempo que tarda desde que es introducido a la cola de salida hasta que es transmitido efectivamente

⇒ tiempo en llegar de un host a otro se calcula como $T = \sum t_i$

Dado que la consigna informa que los tiempos de encolado y procesamiento son despreciables entonces estos dos serán de 0, y restará calcular el tiempo de inserción y propagación en cada tramo.

Cálculo ida A → B:

L1 → L2 → L5 → L8

L1:
- t1ins = L / R = 1000 bytes / 10 mbps = 10^-4 sec
- t1prop = d / c = 100 m / 1.7 x 10^5km/s = 5.88 x 10^-7 sec

L2:
- t2ins = L / R = 1000 bytes / 200 mbps = 10^-4 sec / 20 = 5 x 10^-6 sec
- t2prop = d /c = 10km / 2 x 10^5km/s = 5 x 10^-5 sec

L5:
- t5ins = L / R = 1000 bytes / 100 mbps = 10^-4 sec / 10 = 10^-5 sec
- t5prop = d / c = 2km / 2 x 10^5km/s = 10^-5 sec

L8:
- t8ins = L / R = 1000 bytes / 10 mbps = 10^-4 sec
- t8prop = d / c = 50 m / 1.7 x 10^5km/s = 5.88 x 10^-7 sec / 2 = 2.99 x 10^-7 sec

⇒ La latencia de A → B resulta = 10^-4 sec + 5.88 x 10^-7 sec + 5 x 10^-6 sec + 5 x 10^-5 sec + 10^-5 sec + 10^-5 sec + 10^-4 sec + 2.99 x 10^-7 sec = 2.76 x 10^-4 sec

Cálculo vuelta B → A:

L8 → L7 → L4 → L1

L8:
- t8ins = L / R = 1000 bytes / 10 mbps = 10^-4 sec
- t8prop = d / c = 50 m / 1.7 x 10^5km/s = 5.88 x 10^-7 sec / 2 = 2.99 x 10^-7 sec

L7:

- t7_ins = L / R = 1000 bytes / 200 mbps = 10^-4 sec / 20 = 5 x 10^-6 sec
- t7_prop = d / c = t5_prop * 3 = 3 x 10^-5 sec

L4:

- t4_ins = L / R = 1000 bytes / 100 mbps = 10^-4 sec / 10 = 10^-5 sec
- t4_prop = 60 * t1_prop = 3.528 x 10^-5 sec

L1:

- t1_ins = L / R = 1000 bytes / 10 mbps = 10^-4 sec
- t1_prop = d / c = 100 m / 1.7 x 10^5km/s = 5.88 x 10^-7 sec

⇒ La latencia de B → A resulta = … = 2.81 x 10^-4 sec

⇒ Finalmente el RTT resulta 2.76 x 10^-4 sec + 2.81 x 10^-4 sec = 5.57 x 10^-4 sec

Preguntas teóricas:

a. ¿Qué es la latencia?
b. ¿Cuáles son sus componentes? Describirlos brevemente.
c. ¿Qué métrica conoce para medirla? ¿Qué componente tiene mayor incidencia en el cálculo de la latencia?

a. Retardo entre un estímulo y la respuesta.
b. Descritos en el comienzo del ejercicio práctico.
c. Con el RTT podemos obtener una estimación de la latencia mandando un paquete al host y recibiendo otro paquete idéntico del mismo host i.e. con el comando ping. La respuesta es depende, ya que si quiero mandar un paquete desde ARG a RUSIA probablemente la componente que tenga mayor incidencia sea la de propagación ya que D va a ser muy grande, en cambio si hay mucha congestión en la red por ejemplo el tiempo de encolado puede ser el componente con mayor incidencia.

TCP

P2 - TCP

Por medio de una conexión TCP se transfiere desde un host A a un host B un archivo de 27326 B. De acuerdo con las tecnologías de enlace que utiliza el host A, MSS=1500B. Además, sabemos que su sistema operativo opera con TCP Tahoe, con una IW=2MSS. El sistema utiliza ssthresh=4MSS y el rwnd=16MSS. Sabemos que la conexión sufrirá la pérdida del séptimo segmento de datos transmitido.
Completar la tabla como justificación para responder las siguientes preguntas:

a. ¿Cuál es el valor de cwnd(n) antes de finalizar la transmisión? Es decir, el valor de la ventana de congestión durante la última ráfaga de segmentos transmitidos.

b. ¿El algoritmo entra en la etapa de Fast Retransmit? ¿Y Fast Recovery? En caso de entrar en Fast Retransmit, ¿cuál es el número del último segmento enviado antes de realizarlo?

RTT	CWND	RWND	FlightSize	Recv Bytes	SSTH	Comments

Datos:

- Total size = 27326 B = 18.21 ⇒ Se deben enviar 19 MSS
- MSS = 1500 B
- TCP Tahoe
- Initial Window (IW) = 2 MSS
- SSTRESH = 4 MSS (slow start threshold size)
- RWND = 16 MSS
- Se pierde el séptimo segmento de datos transmitido

RTT	CWND	RWND	FlightSize	Recv Bytes	SSTH	Comments
1	2 MSS	16 MSS	2 MSS	2 MSS	4 MSS	Slow start
2	4 MSS	16 MSS	4 MSS	6 MSS	4 MSS	Congestion Avoidance
3	5 MSS	16 MSS	5 MSS	10 MSS	4 MSS	Se pierde el 7mo paquete, los siguiente 4 retornan un ACK identico, por lo tanto Tahoe hace fast retransmit

4	1 MSS	16 MSS	1 MSS	11 MSS	2 MSS	Se reenvió el paquete perdido por fast retransmit, ahora se pasa a Slow start
5	2 MSS	16 MSS	2 MSS	13 MSS	2 MSS	Se pasa a congestion avoidance
6	3 MSS	16 MSS	3 MSS	16 MSS	2 MSS	-
7	4 MSS	16 MSS	3 MSS	19 MSS	2 MSS	Fin de envío del archivo

a. El valor de CWND antes de finalizar la transmisión es de 4 MSS = 6000 B
b. Si entra en Fast retransmit en la línea resaltada, dado que estamos en TCP tahoe no corresponde fast recovery en este caso. El último segmento enviado antes de fast retransmit es el segmento 11.

<u>Preguntas teóricas</u>

a. ¿Es posible para una aplicación tener transmisión de datos confiable aún cuando la aplicación utilice UDP?. ¿Cómo?
b. ¿Qué significa que un protocolo de transporte implemente un servicio de entrega confiable? Dé un ejemplo

a. Claro que sí, un claro ejemplo es el Trabajo Práctico que hicimos, se deberá implementar sobre UDP algún protocolo dado que garantice la transmisión confiable, en ese caso la aplicación si podrá utilizar rdt. Caso contrario no será confiable la transmisión de datos. Algunas formas de implementarlo es Stop & Wait, Go Back N, Selective repeat. Por ejemplo en stop and wait, cada vez que se manda un paquete se espera a recibir el ACK por parte del otro host para saber que recibió correctamente el paquete, en caso de llegar a un timeout (debido a que se perdió el paquete o nunca se recibió el ACK del servidor) se retransmitirá el paquete con el mismo número de secuencia, de esta forma en caso de que el servidor ya haya recibido ese paquete anteriormente y se haya perdido el ACK sabrá que el paquete retransmitido es el que ya tiene, por lo tanto mandará de nuevo la confirmación.

b. Que un protocolo tenga servicio de entrega confiable, significa que los paquetes que envía un host a través de este protocolo garantiza la llegada al host destino a pesar de problemas en las capas inferiores como corrupción de paquetes por interferencia en la capa de enlace, pérdidas de paquetes porque por diversos motivos i.e. un router dropeo un paquete por overflow del buffer.

chequeo de integridad
orden
que no se pierdan paquetes, que llegue todo
control de flujo

P3 - IP Routing

Un ISP tiene como clientes a la empresa A y a la empresa B. La asignación de prefijos es la siguiente:

Empresa A: 122.50.80.0/26
Empresa B: 122.50.64.0/18

El ISP tiene un único router con tres puertos:

- P1 conecta al resto de Internet, siendo la IP de salida IP_{dfgw}
- P2 conecta a Empresa A, con IP de router IP_A
- P3 conecta a Empresa B , con IP de router IP_B

a) Graficar topología y completar la tabla de ruteo del ISP.

Network Prefix	Subnet Mask	Next Hop	Outgoing Interface

b) ¿Qué prefijo agregado debe anunciar el ISP al resto de Internet para las empresas A y B?

Network Prefix	Subnet mask	Next hop	Outgoing interface
122.50.80.0	255.255.255.192	IP_A	P2
122.50.64.0	255.255.192.0	IP_B	P3
0.0.0.0	0.0.0.0	IP_dfgw	P1

122.50.01010000.00000000
122.50.01000000.00000000

Lo que habría que anunciar sería 122.50.64.0 /18

longest prefix match, me quedo siempre con el más específico (el que tiene mascara mas larga)

T3 - IP Routing

Dada una tabla de ruteo, se busca optimizar la configuración de las entradas. Explicar y dar un ejemplo de los siguientes casos:

a. La tabla de ruteo contiene 3 entradas que se pueden agregar en una única entrada.

b. La tabla de ruteo contiene una entrada ya contenida en otra entrada.

c. La tabla de ruteo contiene una entrada mal configurada, donde el prefijo es más específico de lo que la máscara permite.

Pregunta. Si tenemos el siguiente prefijo: 182.64.46.0/x.
¿Cuál es el mínimo valor que puede tomar x?. ***Justificar.***

a. La tabla de ruteo contiene 3 entradas que se pueden agregar en una única entrada

Network Prefix	**Outgoing interface**
122.50.80.0 /24	P1
122.50.81.0 /24	P1
122.50.82.0 /23	P1
0.0.0.0 /0	P2

Las primeras 2 entradas se pueden agregar dado que:

- El puerto de salida es el mismo
- La máscara es la misma /24
- Las redes son contiguas, solo difieren en el último bit

⇒ quedaría 122.50.80.0 /23 → P1

Luego de esta agregación, se podría agregar nuevamente con la siguiente entrada dado que:

- El puerto de salida es el mismo

- La máscara es la misma /23
- Las redes son contiguas, solo difieren en el último bit:
    - 122.50.01010000 → 122.50.80.0
    - 122.50.01010010 → 122.50.82.0

⇒ quedaría 122.50.80.0 /22 → P1

b. La tabla de ruteo contiene una entrada ya contenida en otra

Network Prefix	**Outgoing interface**
122.50.81.0 /24	P1
122.50.80.0 /23	P1
0.0.0.0 /0	P2

Primero esto lo vemos porque cuando agregamos el resultado fue la segunda entrada pero también lo podemos ver en binario para mayor clarificación

- 122.50.1010001.0 → P1
- 122.50.1010000.0 → P1

Observamos que la primera está incluida en la segunda

c. La tabla de ruteo contiene una entrada mal configurada, donde el prefijo es más específico de lo que permite la máscara.

Network Prefix	**Outgoing interface**
122.50.81.192 /24	P1
122.50.80.0 /23	P1
0.0.0.0 /0	P2

Pregunta. Si tenemos el siguiente prefijo: 172.128.56.0/x. ¿Cuál es el mínimo valor que puede tomar x?. ***Justificar.***

172.128.00111000.00000000

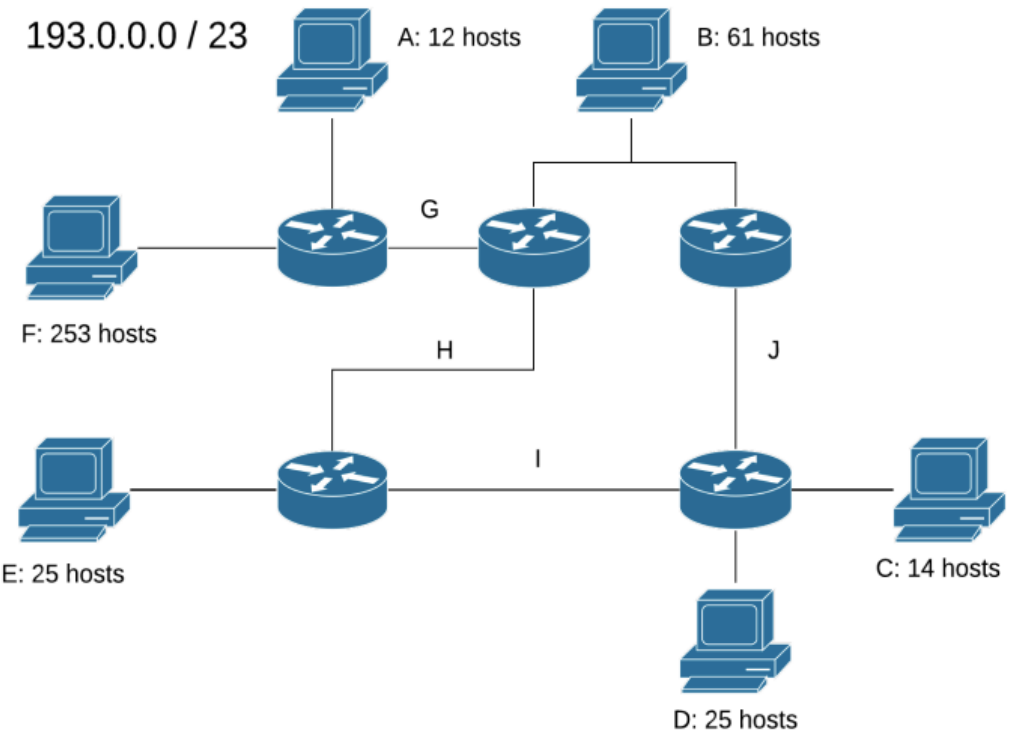
El valor mínimo que debería tomar el prefijo es /21, dado que sino la tabla estará mal configurada ya que el prefijo sería más específico de lo que permite la máscara. Si x fuera menor luego al hacer la operación AND se perderán los 1's que especifican la red. Ademas vemos que no matchea porque cuando le aplique la mascara al IP destino van a haber bits que nunca matcheen porque los puse en 0 con la AND.

Subnetting

Dada la siguiente configuración de hosts y routers, y el espacio 193.0.0.0/23, se pide separar en subredes minimizando la cantidad de IPs sin usar.
Ante igualdad de condiciones para ubicar varias subredes:

1. **Asignar bloques utilizando los prefijos en orden de numeración ascendente**
   (Ej: si tenemos la opcion de usar 117.0.1.0/24 o 117.0.0.0/24, debemos utilizar primero el espacio de direcciones 117.0.0.0/24).
2. **Asignar bloques de direcciones priorizando las redes con mayor cantidad de hosts**
   (Ej: si se deben asignar dos bloques de 64 direcciones IP para dos subredes distintas S_x y S_y, donde `x` e `y` representan la cantidad de hosts de cada subred y con $32 < x < y < 64$, S_y debe asignarse en un espacio de direcciones de menor numeración).
3. **Si dos subredes necesitan la misma cantidad de IPs, ubicar primero la subred cuya letra viene primero en el abecedario**
   (Ej: si las redes P y J tienen necesitan un bloque de 32 IPs, ubicar primero la J y luego la P).

Estas aclaraciones definen una única resolución posible de la configuración. Cualquier otra solución será considerada incorrecta.



Tenemos el espacio 193.0.0.0 /23, por lo tanto tengo 9 bits para direccionar, la cantidad de host que se puede tener es 2**(32 - 23) - 2 = 2**(9) - 2 = 510 !

Subnet	# Hosts	# router	Block	Prefix/mask
A	12	1	16	193.0.1.224/28
B	61	2	128	193.0.1.0/25
C	14	1	32	193.0.1.128/27
D	25	1	32	193.0.1.160/27
E	25	1	32	193.0.1.192/27
F	253	1	256	193.0.0.0/24
G	0	2	4	193.0.1.240/30
H	0	2	4	193.0.1.244/30
I	0	2	4	193.0.1.248/30
J	0	2	4	193.0.1.252/30

*Nota: recordar que a cada subred se le deben asignar 2 IPs, una para la dirección de red y otra para la dirección de broadcast
*Nota: la sumatoria de bloques da justo 512 que es el espacio que podemos direccionar con /23

T4 - Subnetting

Responda Verdadero o Falso. Justifique en caso de que la afirmación sea falsa.
Aclaración: Se debe responder correctamente TODOS los items del ejercicio.

1. Classful routing es el mecanismo por el cual se particionan las redes debido a que aprovecha mejor el espacio de direcciones.

2. Al subnetear un espacio de direcciones de clase C, un host puede tener asignada cualquiera de las 256 direcciones posibles.

3. Con classful routing las clases de red se pueden identificar sin necesidad de conocer la máscara.

4. Todas las direcciones asignadas a un mismo dispositivo deben pertenecer a la misma subred.

1. F, ya que por ejemplo si estoy en clase C y tengo todas las direcciones ocupadas y necesito 2 direcciones más el salto a clase B desaprovecharía muchas direcciones sin usar.
2. F, no hay 256 direcciones ya que se reservan 2 para red y broadcast.

3. Verdadero ya que se designa un rango de direcciones para cada clase. Viendo el primer octeto ya podemos darnos cuenta de que tipo de clase es.
4. F, notar que en el ejemplo anterior un mismo router tiene direcciones asignadas a distintas subredes

Fragmentación

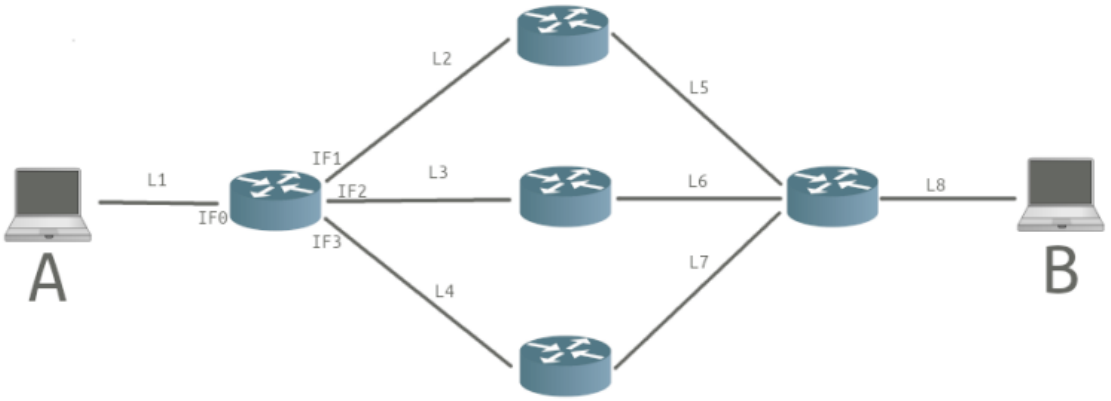
P5 - Fragmentación

Dada la siguiente configuración, el *host A* envía un paquete cuya IP destino corresponde al *host B*. El router conectado directamente al *host A* por el enlace *L1* tiene configurada su tabla de ruteo de manera tal que el paquete es forwardeado por la interfaz `<if1>`.

Teniendo en cuenta los siguientes datos:

Datagram Header Fields	**IP Dest (host B)**	**Header Size**	**Datagram Length**	**Identifier**	**Do Not Fragment**
	200.27.155.1	20 Bytes	1400 Bytes	0XF1B1	0

Enlace	L1	L2	L3	L4	L5	L6	L7	L8
MTU (bytes)	1500	1280	1180	1080	600	500	400	1500



Se pide describir a continuación **los campos del header IP** de los paquetes en cada enlace por el que atraviesa la red completando la siguiente tabla:

Link	MTU	Datagram ID	payload Size		Total Length	ID	Do Not Fragment	More fragments	Fragment offset
L1	1500	0xF1B1	1480		1400 + 20	0xF1B1	0	0	0
L2	1280	0xF1B1	1256		1256 + 20	0xF1B1	0	1	0
L2	1280	0xF1B1	1256		144 + 20	0xF1B1	0	0	157
L5	600	0xF1B1	576		576 + 20	0xF1B1	0	1	0

L5	600	0xF1B1	576		576 + 20	0xF1B1	0	1	72
L5	600	0xF1B1	576		104 + 20	0xF1B1	0	1	144
L5	600	0xF1B1	576		144 + 20	0xF1B1	0	0	157
L8	1500	0xF1B1	1480		576 + 20	0xF1B1	0	1	0
L8	1500	0xF1B1	1480		576 + 20	0xF1B1	0	1	72
L8	1500	0xF1B1	1480		104 + 20	0xF1B1	0	1	144
L8	1500	0xF1B1	1480		144 + 20	0xF1B1	0	0	157

Datagram size != payload size

Corregido:

Link	MTU	Datagram ID	payload Size		Total Length	ID	Do Not Fragment	More fragments	Fragment offset
L1	1500	1	1380		1400	0xF1B1	0	0	0
L2	1280	1.1	1256		1276	0xF1B1	0	1	0
L2	1280	1.2	124		144	0xF1B1	0	0	157
L5	600	1.1.1	576		596	0xF1B1	0	1	0
L5	600	1.1.2	576		596	0xF1B1	0	1	72
L5	600	1.1.3	104		124	0xF1B1	0	1	144
L5	600	1.2	124		144	0xF1B1	0	0	157
L8	1500	1.1.1	576		596	0xF1B1	0	1	0
L8	1500	1.1.2	576		596	0xF1B1	0	1	72
L8	1500	1.1.3	104		124	0xF1B1	0	1	144
L8	1500	1.2	124		144	0xF1B1	0	0	157

Preguntas teóricas:

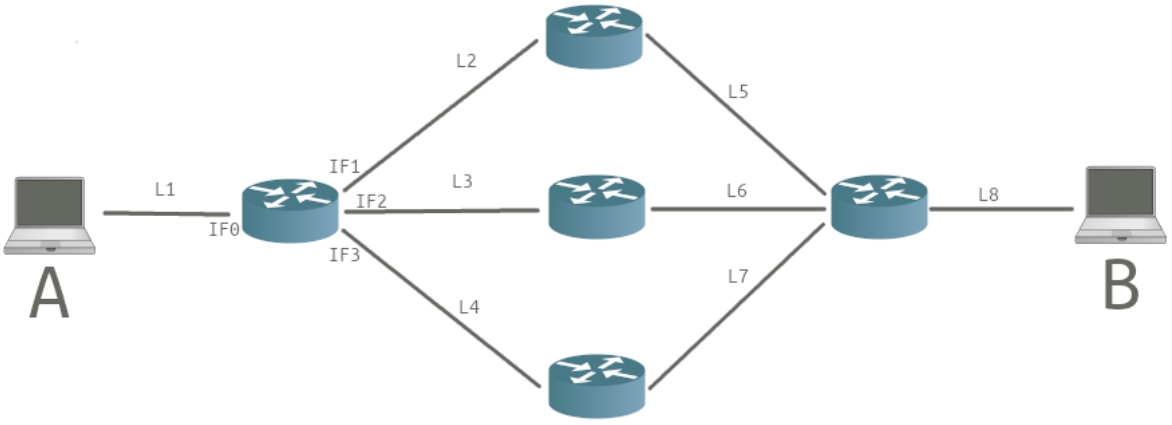
Un paquete P es fragmentado en N paquetes F_i al atravesar una red IPv4 antes de llegar al host destino. En el camino, uno de los paquetes F_i se pierde en el camino. ¿Qué consecuencia tiene la pérdida del paquete Fi sabiendo que el protocolo de la capa de transporte utilizado es TCP?

Primero que nada cuando los paquetes fragmentados lleguen al host, el host intentará reconstruir el datagrama, al perderse uno de los fragmentos no podrá por lo tanto luego de una espera el host terminará descartando ese datagrama y los fragmentos recibidos. Dado que se está utilizando TCP, y es un protocolo de transferencia confiable, retransmitirá el paquete completo, suponiendo que esta vez ningún fragmento se pierde en el camino, el host B recibirá el paquete correctamente.

Parcial - 2020 2C - 2op

Latencia - CDN

Se quiere calcular el RTT para medir la latencia entre dos host bajo la siguiente configuración:



Datos:
Packet Size = 1000 bytes

	L1	L2	L3	L4	L5	L6	L7	L8
Distancia	100 m	10 km	4 km	6 km	2 km	10 km	6 km	50 m
Ancho de Banda	10 Mbps	200 Mbps	200 Mbps	100 Mbps	100 Mbps	50 Mbps	200 Mbps	10 Mbps
Velocidad de Propagación	$1.7x\ 10^5$ km/s	$2 x\ 10^5$ km/s	$2 x\ 10^5$ km/s	$1.7x\ 10^5$ km/s	$2 x\ 10^5$ km/s	$2 x\ 10^5$ km/s	$2 x\ 10^5$ km/s	$1.7x\ 10^5$ km/s

1 Mbps = 10^6 bits / seg

El RTT se debe calcular utilizando un segmento de prueba de tamaño **1000 Bytes**, y será el mismo para la ida y la vuelta.
Tener en cuenta la asimetría de caminos siendo:

Ruta A → B: $L_1 \rightarrow L_4 \rightarrow L_7 \rightarrow L_8$
Ruta B → A: $L_8 \rightarrow L_6 \rightarrow L_3 \rightarrow L_1$

Los tiempos de encolado y procesamiento son despreciables.
Detallar los pasos del cálculo obtenido.

La latencia se descompone por:

- tiempo de inserción = L / R
- tiempo de propagación = D / C
- tiempo de procesamiento → despreciable por consigna
- tiempo de encolado → despreciable por consigna

Luego dado que el camino es asimétrico el RTT será la suma de tiempos de A → B + B → A. Antes de comenzar con los cálculos para facilitar las cuentas pasamos el size del paquete de bytes a bits, que bits es la unidad que se utiliza en el throughput

⇒ packet = 1000 bytes = 1000 * 8 bits = 8000 bits

Cálculo A → B

- Tiempos en L1
  - t1_ins = L / R = 8000 bits / 10 Mbps = 8 x 10^-4 seg
  - t1_prop = D / C = 0.1 km / 1.7x 10^5 km/s = 5.88 x 10^-7 seg
- Tiempos en L4
  - t4_ins = t1_ins / 10 = 8 x 10^-5 seg
  - t4_prop = t1_prop * 60 = 3.53 x 10^-5 seg
- Tiempos en L7
  - t7_ins = t1_ins / 20 = 3 x 10^-5 seg
  - t7_prop = 6 km / 2 x 10^5 km/s = 3 x 10^-5 seg
- Tiempos en L8
  - t8_ins = t1_ins = 8 x 10^-4 seg
  - t8_prop = t1_prop / 2 = 2.94 x 10^-7 seg

⇒ Luego la sumatoria de estos tiempos da el T A → B = 1.79 x 10^-3

Cálculo A → B

- Tiempos en L8
  - t8_ins = t1_ins = 8 x 10^-4 seg
  - t8_prop = t1_prop / 2 = 2.94 x 10^-7 seg
- Tiempos en L6
  - t6_ins = t1_ins / 5 = 1.6 x 10^-5 seg
  - t6_prop = 10 km / 2 x 10^5 km/s = 5 x 10^-5 seg
- Tiempos en L3
  - t3_ins = t7_ins = t1_ins / 5 = 1.6 x 10^-4 seg
  - t3_prop = 4 km / 2 x 10^5 km/s = 2 x 10^-5 seg
- Tiempos en L1
  - t1_ins = L / R = 8000 bits / 10 Mbps = 8 x 10^-4 seg
  - t1_prop = D / C = 0.1 km / 1.7x 10^5 km/s = 5.88 x 10^-7 seg

⇒ Luego la sumatoria de estos tiempos da el T A → B = 1.85E-03

Finalmente, el resultado es 3.64E-03

<u>Preguntas teóricas</u>

Responda Verdadero o Falso. Justifique la respuesta.

a. Una CDN propone manejar los protocolos de ruteo y el procesamiento de paquetes (control plane) desde una unidad central (controller), y separar dichas funciones de aquellas netamente relacionadas con el envío de paquetes (data plane).

   Falso, de lo que está hablando es una SDN en realidad. La idea del CDN es tener contenido estático distribuido ya que mantener todo en un datacenter sería un single point of failure, además de que sería mucho menor el throughput ya que hay que hacer un camino más largo y si el video es muy solicitado también se estará desaprovechando ancho de banda.

b. La latencia puede ser reducida por medio de una CDN.

   Verdadero ya que al estar más cerca de los clientes los tiempos de propagación disminuyen y probablemente también disminuya la cantidad de routers que deba recorrer el paquete para llegar al cliente en comparación al camino del datacenter.

c. La CDN ayuda a maximizar el throughput.

   Verdadero por la misma razón comentada anteriormente.

d. La ubicación geográfica de los clientes es un factor determinante para la performance de una CDN.

   Falso primero habría que determinar que sería la performance del CDN, si a eso nos referimos a atender los request del cliente, la ubicación geográfica no sería un factor determinante para la CDN.

TCP

Dada una conexión TCP **recién establecida** entre dos host para la cual el RTT es de *250 ms* y el MSS es de *1 KB*. Se desea transmitir un archivo de *20 KB (20 MSS)*.
La red por la cual están conectados está congestionada, y se sabe que de enviar una ráfaga de **7 o más paquetes** , se pierden **todos** los paquetes de la ráfaga enviados por el cliente.

Otros datos:
- Initial Window = 1 KB (1 MSS)
- Loss Window = 1 KB (1 MSS)
- SS Threshold = 32 KB (32 MSS)
- RWND = 10 KB (10 MSS)
- Timeout equivale a 2 RTT

a. Calcular el tiempo que se tardará en enviar la totalidad del archivo.
Justificar el desarrollo de la transmisión utilizando la tabla propuesta.

La unidad de la tabla es KB. En realidad MSS, asumo que se usa TCP Tahoe

RTT	CWND	RWND	FlightSize	Recv Bytes	SSTH	Comments
1	1	10	1	1	32	Arranca en Slow start
2	2	10	2	3	32	Sigue en SS
3	4	10	4	7	32	Sigue en SS
4	8	10	8	7	32	Por enunciado se pierden todos los paquetes. Sigue en SS
5	8	10	0	7	32	Por enunciado se pierden todos los paquetes. Sigue en SS
6	1	10	1	8	4	Como ya pasaron 2 RTT se dispara el timeout por lo tanto la CWND pasa a ser LW y SSTH es CWND / 2. Sigue en SS
7	2	10	2	10	4	Sigue en SS
8	4	10	4	14	4	CWND >= SSTH por lo tanto entro en CA
9	5	10	5	19	4	Sigue en CA
10	6	10	1	20	4	Se finaliza el envio del archivo

Cómo tomó 10 RTT el envío y cada RTT equivale a *250 ms,* luego el tiempo total será de 2500 ms, es decir 2.5 segundos.

b. ¿Qué sucedería en caso de que, cuando se produce la pérdida, lo que se pierde son los paquetes que responde el servidor en lugar de los enviados por el cliente?
Justificar el desarrollo de la transmisión utilizando la tabla propuesta.

RTT	CWND	RWND	FlightSize	Recv Bytes	SSTH	Comments
1	1	10	1	1	32	Arranca en Slow start
2	2	10	2	3	32	Sigue en SS
3	4	10	4	7	32	Sigue en SS
4	8	10	8	15	32	Por enunciado se pierden los ACKs del servidor. Sigue en SS
5	8	10	0	15	32	No vuelve a enviar a enviar paquetes porque no recibió ACK de la ráfaga anterior
6	1	10	1	15	4	Como ya pasaron 2 RTT se dispara el timeout por lo tanto la CWND pasa a ser LW y SSTH es CWND / 2. Sigue en SS. Lo que va a pasar es que el cliente envía el paquete 8, pero el servidor responde con ACK del paquete 15 que fue el último que recibió ok, por lo tanto el cliente ahora sabe que hasta el 15 está todo ok y tiene que proseguir con el paquete 16.
7	2	10	2	17	4	Sigue en SS
8	4	10	3	20	4	CWND >= SSTH por lo tanto entro en CA

En este caso el envío se completa en 8 RTT, 2 menos que si se hubieran pérdido los paquetes del cliente al servidor por lo tanto, el tiempo total será de 2 segundos.

Pregunta Teórica

¿Qué significa que un protocolo de transporte implemente un servicio de entrega confiable? Dé un ejemplo.

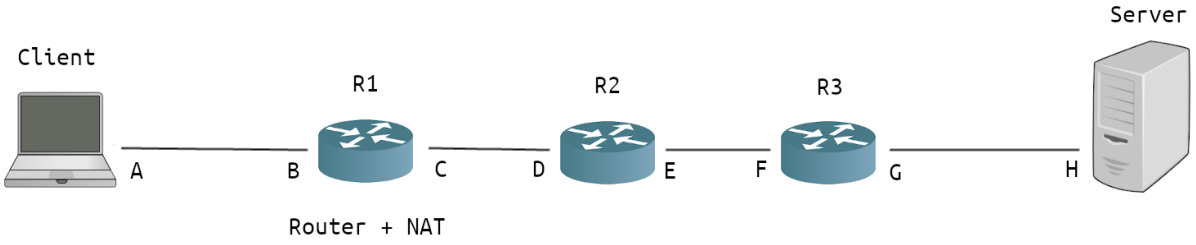
Un protocolo de transporte sea confiable significa que provee las siguientes características:

- Garantiza orden de llegada de los paquetes enviados
- Garantiza el envío de los paquetes, es decir si algún paquete se pierde en el camino el protocolo garantiza tomar las medidas necesarias para retransmitir y asegurar la llegada eventual del mismo.
- Control de integridad, garantiza la integridad de los paquetes, que lleguen sin ser corruptos a destino.
- Control de flujo, asegura establecer una tasa de transmisión tal que el receptor la pueda recibir sin provocar algún tipo de overflow.

Un ejemplo es TCP**,** esto lo logra ajustando la velocidad de transmisión de los datos (control de flujo), a través del uso de Seq. numbers, ACKs y timers. Pero también se pueden implementar protocolos de aplicación sobre la capa de transporte UDP para conseguir el servicio de entrega confiable como Stop & Wait, GO back N, selective repeat.

NAT

Completar las direcciones IP y puertos de las headers de capa 3 y 4 en cada punto del esquema para el paquete que genera el *cliente* con una *consulta HTTP* y el paquete que genera el *servidor web* con la *respuesta HTTP*. ***Justificar***.



	A	B	C	D	E	F	G	H
IP	192.168.0.3	192.168.0.1	210.210.210.1	210.210.210.2	199.199.199.1	199.199.199.2	205.200.203.1	205.200.203.9

Http Request

	IP origen	IP destino	Puerto Origen	Puerto Destino
A	192.168.0.3	205.200.203.9	PORT X	80
B	192.168.0.3	205.200.203.9	PORT X	80
C	210.210.210.1	205.200.203.9	PORT Y	80
D	210.210.210.1	205.200.203.9	PORT Y	80
E	210.210.210.1	205.200.203.9	PORT Y	80
F	210.210.210.1	205.200.203.9	PORT Y	80
G	210.210.210.1	205.200.203.9	PORT Y	80
H	210.210.210.1	205.200.203.9	PORT Y	80

Http Response

	IP origen	IP destino	Puerto Origen	Puerto Destino
H	205.200.203.9	210.210.210.1	80	PORT Y
G	205.200.203.9	210.210.210.1	80	PORT Y

F	205.200.203.9	210.210.210.1	80	PORT Y
E	205.200.203.9	210.210.210.1	80	PORT Y
D	205.200.203.9	210.210.210.1	80	PORT Y
C	205.200.203.9	210.210.210.1	80	PORT Y
B	205.200.203.9	192.168.0.3	80	PORT X
A	205.200.203.9	192.168.0.3	80	PORT X

Preguntas teóricas

Dada una tabla de ruteo, se busca optimizar la configuración de las entradas. Explicar y dar un ejemplo de los siguientes casos:

a. La tabla de ruteo contiene 4 entradas que se pueden agregar en una única entrada.

192.168.0.0	255.255.255.0	IF2
192.168.1.0	255.255.255.0	IF2
192.168.2.0	255.255.255.0	IF2
192.168.3.0	255.255.255.0	IF2

Las primeras dos se agregan en

192.168.0000 0000.0 /24
192.168.0000 0001.0 /24 ⇒ 192.168.0.0 /23

192.168.0000 0010.0 /24
192.168.0000 0011.0 /24 ⇒ 192.168.2.0 /23

192.168.0.0 /23 ⇒ 192.168.0000 0000.0
192.168.2.0 /23 ⇒ 192.168.0000 0010.0 ⇒ 192.168.0.0 /22

b. La tabla de ruteo contiene una entrada ya contenida en otra entrada.

192.168.0.0	255.255.252.0	if2
192.168.3.0	255.255.255.0	if2

c. La tabla de ruteo contiene una entrada mal configurada, donde el prefijo es más específico de lo que la máscara permite.

192.168.0.0	255.255.252.0	if2
192.168.3.255	255.255.255.0	if2

Pregunta. Si tenemos el siguiente prefijo: 172.128.56.0/x.
¿Cuál es el mínimo valor que puede tomar x?. ***Justificar.***

El mínimo valor que podría tomar es

172.128.00111000.0 sería 21, ya que sino quedaría afuera el último 1 (o más bits en 1 dependiendo que tanto más chica sea la máscara) y cuando se haga la operación AND con la máscara quedará siempre 0 donde está ese 1 y no podrá matchear con el prefijo de red de la tabla, lo cual sería un error.

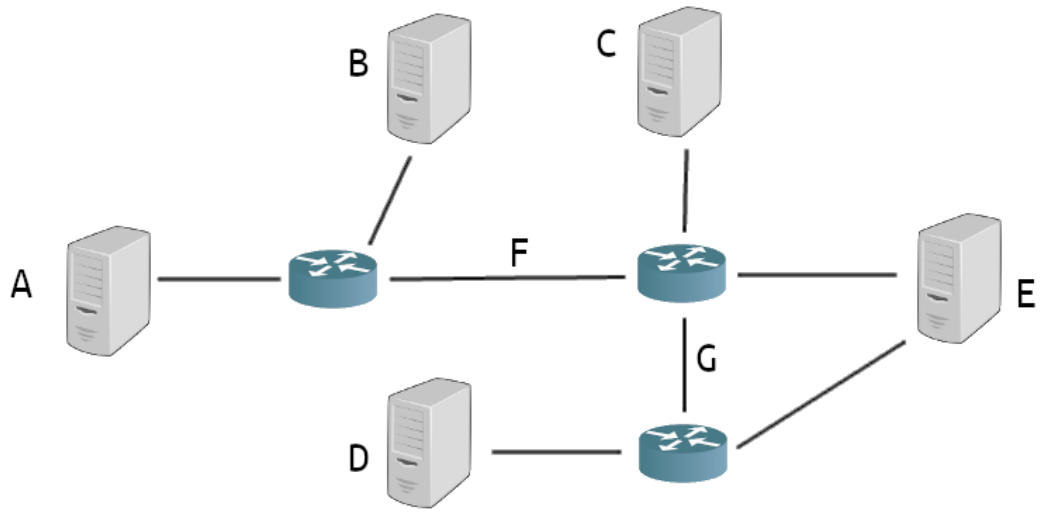
Subnetting

Dada la siguiente configuración de hosts y routers, y el espacio ***172.200.108.0/22***, se pide separar en subredes minimizando la cantidad de IPs sin usar.

Ante igualdad de condiciones para ubicar varias subredes:

1. **Asignar bloques utilizando los prefijos en orden de numeración ascendente**
   (Ej: si tenemos la opcion de usar 117.0.1.0/24 o 117.0.0.0/24, debemos utilizar primero el espacio de direcciones 117.0.0.0/24).
2. **Asignar bloques de direcciones priorizando las redes con mayor cantidad de hosts**
   (Ej: si se deben asignar dos bloques de 64 direcciones IP para dos subredes distintas S_x y S_y, donde x e y representan la cantidad de hosts de cada subred y con $32 < x < y < 64$, S_y debe asignarse en un espacio de direcciones de menor numeración).
3. **Si dos subredes necesitan la misma cantidad de IPs, ubicar primero la subred cuya letra viene primero en el abecedario**
   (Ej: si las redes P y J tienen necesitan un bloque de 32 IPs, ubicar primero la J y luego la P).

Este criterio arbitrario define una única resolución posible de la configuración. Cualquier otra solución será considerada incorrecta.



	A	B	C	D	E
# Hosts	500	128	80	61	14

SubNet	# Host	# Router	Block	Prefix/Mask
A	500	1	512	**172.200.108.0/23**
B	128	1	256	**172.200.110.0/24**
C	80	1	128	**172.200.111.0/25**
D	61	1	64	**172.200.111.128/26**
E	14	2	32	**172.200.111.192/27**
F	0	2	4	**172.200.111.224/30**
G	0	2	4	**172.200.111.228/30**

164.92.136.0/22

SubNet	# Host	# Router	Block	Prefix/Mask
A	255	1	512	**164.92.136.0/23**
B	50	1	64	**164.92.138.0/26**
C	64	1	128	**164.92.137.128/25**
D	32	1	64	**164.92.138.64/26**
E	80	1	128	**164.92.137.0/25**
R	0	3	8	**164.92.138.128/29**

Notar que cuando elijo el bloque, elijo alguna potencia de 2 tal que sea >= que #Hosts + #router + 2. Donde 2 representa la dirección de broadcast y de red.ç

*Nota: recordar que siempre comienzo asignando los bloques más grandes, esto es para no tener problemas de alineación luego.

Preguntas teóricas

Responder Verdadero o Falso. *Justificar*

a. Dada las subredes: A (100 hosts), B (100 hosts), C (80 hosts), todas conectadas a un mismo router. Es posible subnetear la configuración utilizando el prefijo 200.128.64.0/24.

Falso para el A y el B necesitamos bloques de 128, por lo tanto ya excedo /24 ya que solo nos permite direccionar 256.

b. Dada las subredes: A (100 hosts), B (80 hosts), C (10 hosts), todas conectadas a un mismo router. Es posible subnetear la configuración utilizando el prefijo 200.128.64.0/24 dado que tengo 256 direcciones posibles.

   Falso necesito un bloque de 128 para el A y para el B nuevamente otro de 128, por lo tanto nuevamente estamos excediendo la cantidad que podemos mapear con /24.

c. Dada las subredes: A (100 hosts), B (60 hosts), C (60 hosts), todas conectadas a un mismo router. Es posible subnetear la configuración utilizando el prefijo 200.128.64.0/24 dado que tengo 256 direcciones posibles.

   Verdadero necesito un bloque de 128 para A para B y C necesito 63 direcciones por lo tanto con bloques de 64 me alcanza, por lo tanto con un /24 si puedo direccionar las subnets.

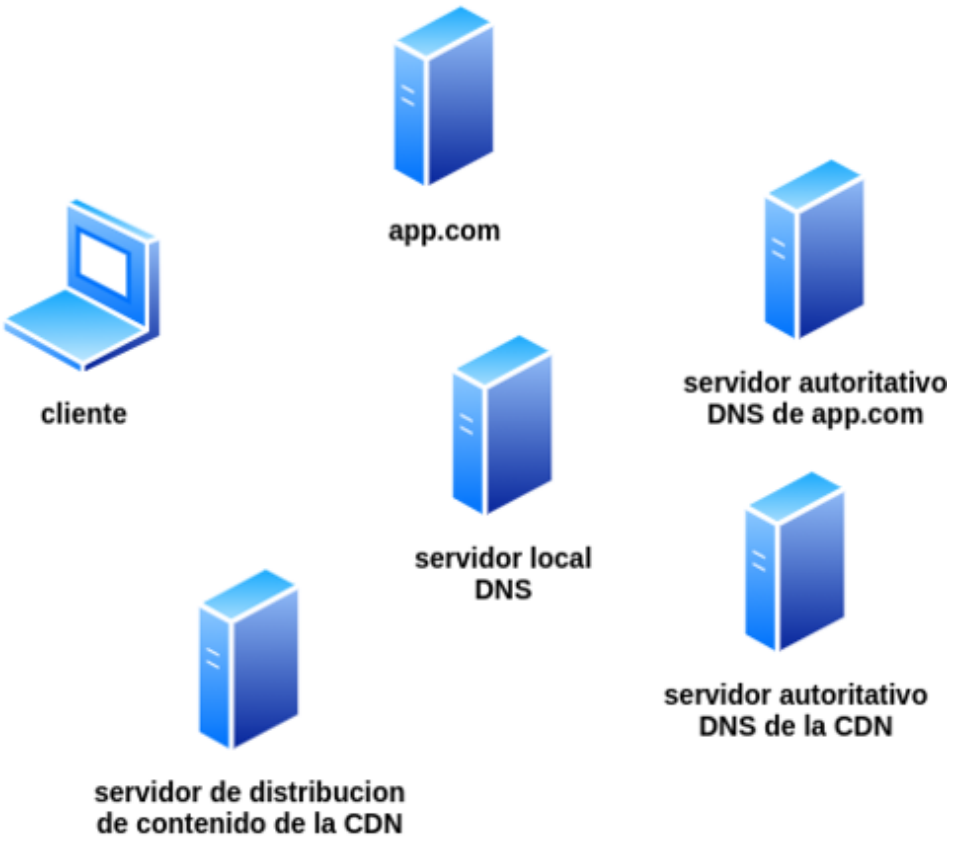
d. Dada las subred: A (300 hosts), B (60 hosts), C (60 hosts), todas conectadas a un mismo router. Es posible subnetear la configuración utilizando el prefijo 200.128.128.0/23 dado que tengo 512 direcciones posibles.

   Falso ya con A consumo todo el bloque /23.

DNS

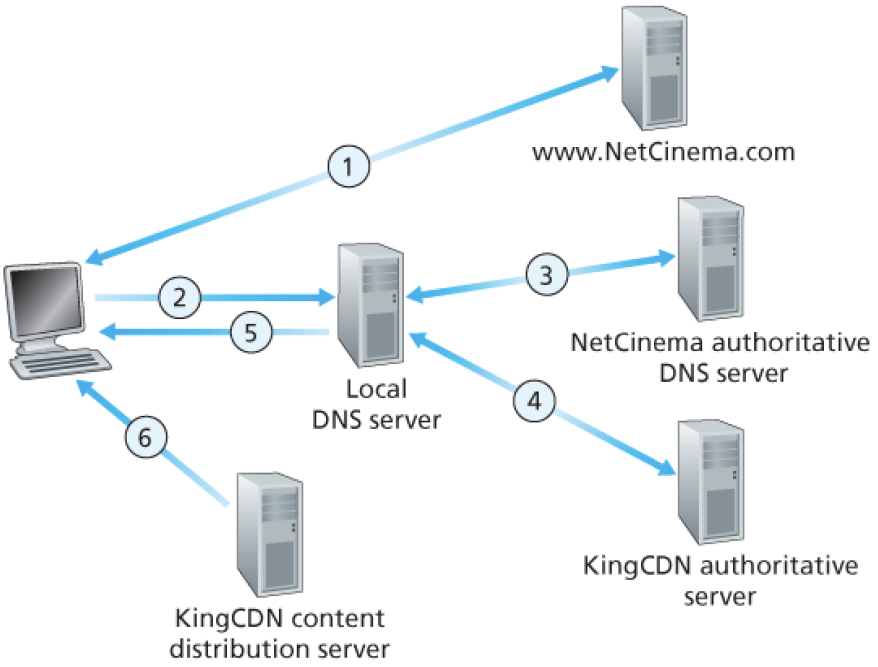
Describir la secuencia de mensajes, explicando el contenido de los mismos que se mandan los host entre sí para acceder al contenido del link ***http://video.app.com/6Y7B23V***, que está alojado en el servidor de distribución de contenido de la CDN.
¿Cuál debe ser el estado de los caches en el cliente y el servidor local para que sólo intervengan los servidores autoritativos de la figura?

Detallar el proceso utilizando flechas con numeración, explicando qué sucede en cada paso.



1. El usuario visita app.com y clickea en la URL ***http://video.app.com/6Y7B23V***
2. Al clickear en la página el user le manda una query DNS a video.app.com.
3. El DNS server local, ya tiene cacheado app.com, por lo tanto le pasa la query al DNS server autoritativo de app.com donde luego observa el “video.” de la URL por lo tanto quiere acceder al contenido multimedia, luego le devuelve al DNS local un hostname del video en el dominio de la CDN i.e. 1231.privateCDN.com
4. El DNS local debe tener cacheada la DNS del CDN privado privateCDN.com, luego hace una segunda query al DNS de la CDN y en este punto le retorna la IP de un servidor de distribución de contenido.
5. El DNS local devuelve la IP del servidor de contenido CDN al user host.
6. Finalmente el host abre una conexion TCP con la CDN y comienza a descargar el contenido mediante un GET HTTP

Respondiendo a la pregunta, el local DNS deberá tener cachedo la IP del DNS autoritativo de app.com sino debería hacer la consulta al root DNS y hacer el proceso iterativo hasta llegar al DNS de app.com, también deberá estar cacheado el DNS de la CDN.



Preguntas teóricas

Responder: ¿De qué forma una CDN puede utilizar DNS para acercar el contenido a los usuarios?

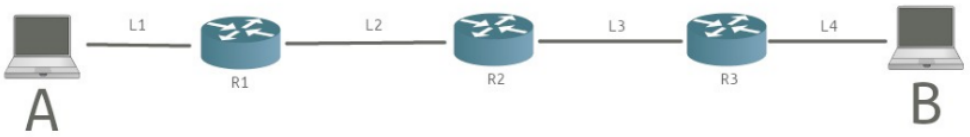
La idea es que haya múltiples servidores CDN distribuidos por el mundo y que dependiendo la ubicación de los request de los usuarios DNS devuelve la IP del CDN más cercano del usuario, de esta forma beneficiando al usuario de tener menos latencia y acceso al contenido más rápido y también como un beneficio global la red se ve menos saturada ya que los paquetes deben hacer un recorrido menor.

Parcial - 2020 2C - 1op

Latencia

1. Latencia

Se quiere calcular el RTT para medir la latencia entre dos host bajo la siguiente configuración:



Datos:
Packet Size = 500 bytes **(L)**

	L1	L2	L3	L4
Distancia **(d)**	100 m	50 km	30 km	50 m
Ancho de Banda **(R)**	25 Mbps	200 Mbps	100 Mbps	25 Mbps
Velocidad de Propagación **(c)**	$1.7x\ 10^5$ km/s	$2\ x\ 10^5$ km/s	$2\ x\ 10^5$ km/s	$1.7x\ 10^5$ km/s

1 Mbps = 10^6 bits / seg

El RTT se debe calcular utilizando un segmento de prueba de tamaño **500 Bytes**, y será el mismo para la ida y la vuelta.
Los tiempos de encolado y procesamiento son despreciables.
Detallar los pasos del cálculo obtenido.

Antes que me olvide, packet size = 500 bytes = 500 * 8 bits = 4000 bits

Dado que los tiempos de encolado y procesamiento son despreciable nos resta calcular el tiempo de inserción y propagación para cada tramo. Por otro lado también es importante notar que el camino de ida y vuelta es el mismo, es decir es simétrico, por lo tanto el RTT será el camino de ida * 2.

- Tiempo en L1
  - t_1_ins = L / R = 4 kbits / 25 mbps = 0.00016 s
  - t_1_prop = D / C = 0.1 km / 1.7e5 km/s = 5.88e-07 s

- Tiempo en L2
    - t_2_ins = 4 kbits / 200 mbps = 2e-05 s
    - t_2_prop = D / C = 50 km / 2e5 km/s = 0.00025 s
- Tiempo en L3
    - t_3_ins = L / R = 4 kbits / 100 mbps = 4e-05 s
    - t_3_prop = D / C = 30 km / 2e5 km/s = 0.00015 s
- Tiempo en L4
    - t_4_ins = L / R = 4 kbits / 25 mbps = 0.00016 s
    - t_4_prop = D / C = 0.05 km / 1.7e5 km/s = 2.94e-07 s

Por lo tanto el tiempo del camino de ida será la sumatoria de todos estos tiempos, que en total sería 0.0007808823529411765 seg.

Siendo el RTT de 0.0015617647058823530 seg que sería aproximadamente 1.56 mseg

Preguntas teóricas

¿Qué es la latencia? ¿Cuáles son sus componentes? ¿Qué es la asimetría de caminos y cómo afecta a la latencia?

La latencia es la demora en la respuesta de realizar un estímulo. Sus componentes son 4:

- Tiempo de inserción: Es la demora en insertar el paquete en el canal
- Tiempo de propagación: Es la demora del paquete en recorrer el canal.
- Tiempo de procesamiento: Es la demora en leer los headers del paquete del router y procesar el paquete en el router.
- Tiempo de encolado: Es la demora del paquete desde que ingresa a la cola hasta que efectivamente sale de ella.

La asimetría de caminos significa que el paquete al enviarse va por un camino distinto al de la respuesta, y debido a esto impacta en la latencia dado que quizá el camino de la ida es mucho más rápido que el de la vuelta i.e. en la vuelta pasó por un router que justo se encontraba congestionado y el tiempo de encolado fue considerablemente alto.

TCP

2. TCP

Dada una conexión TCP **ya establecida** entre dos host para la cual el RTT es de *100 ms* y el MSS es de *2KB*. La red por la cual están conectados está congestionada, y se sabe que de enviar una ráfaga de **más de 5 paquetes** , se pierde el 1er paquete de la ráfaga. El archivo que se busca mandar es de 86KB. Ya se han recibido 28KB del archivo correctamente, la última CWND fue de 3 MSS, el Slow Start threshold es de 4KB.

Otros datos:
- Initial Window = 4KB
- Loss Window = 2KB
- RWND = 20KB
- Timeout muy grande respecto a RTT

a. ¿Cuánto tiempo se tardará en completar la transmisión del archivo desde el estado actual de la transmisión, usando un algoritmo de control congestión con Fast Retransmit/Fast Recovery?

b. ¿Qué sucedería si la ventana de control de flujo se redujera a 10KB? ¿Tardaría más que en la versión anterior?

- RTT = 100ms
- MSS = 2kb
- rafaga > 5 paquetes ⇒ se pierde el 1er paquete de la rafaga

- ArchivoSize = 86 kb = 43 MSS,
- Ya se recibieron 28 kb = 14 MSS ⇒ quedan enviar 29 MSS
- CWND anterior = 3 MSS ⇒ CWND actual = 4 MSS
- SSTRESH = 4KB = 2 MSS
- IW = 4KB = 2 MSS
- LW = 2KB = 1 MSS
- RWND = 20 KB = 10 MSS
- TImeout >> RTT
- Dado que no especifica, asumimos que usamos TCP RENO

RTT	CWND	RWND	FlightSize	Recv Bytes	SSTH	Comments
1	4	10	3	18	2	Dado que el CWND es >= SSTH estamos en congestion avoidance, y W = min(CWND, RWND) = 3
2	5	10	4	23	2	Seguimos en CA, no hay pérdida de paquetes
3	6	10	5	28	2	Por consigna se pierde el primer paquete, por lo tanto se obtienen 5 ACK duplicados del último paquete enviado en la rafaga anterior. Por lo tanto al estar en TCP reno en el próximo RTT entra en modo fast recovery
4	3	10	1	29	3	Fast retransmit, se envía el paquete 24 que se había perdido con una LW = 1MSS, luego aplica fast recovery: se sigue en CA, se disminuye SSTRESH y CWND al valor de $CWND_N$ / 2
5	4	10	4	33	3	Seguimos en CA
6	5	10	5	38	3	Seguimos en CA
7	6	10	5	42	3	Se pierde el primer paquete por consigna y se reciben 5 ACKs repetidos del ultimo paquete de la rafaga anterior
8	3	10	1	43	3	Fast retransmit - fast recovery se envía el paquete perdido

Finalmente el tiempo para enviar el archivo restante es de 8 RTT = 800 ms.

Preguntas teóricas

¿Es posible para una aplicación tener transmisión de datos confiable aún cuando la aplicación utilice UDP?. ¿Cómo?

Por supuesto, i.e. el TP realizado durante el cuatrimestre, se deberá implementar en la capa de aplicación algún protocolo que al usar UDP garantice la transmisión de datos confiable como por ejemplo stop & wait, go back N y selective repeat. Caso contrario UDP no proporciona transmisión de datos confiable y los paquetes transmitidos podrán perderse y nunca llegar a destino.

En el caso de Stop & wait se encapsulan los datos en paquetes y por cada paquete que se envía al servidor se espera un ACK del server para saber que llego bien, en caso de no recibir el ACK 2 cosas podrían haber pasado:

- El servidor nunca recibió el paquete, por lo tanto luego de un tiempo máximo de espera, se producirá el timeout y el cliente reenviará el paquete
- El cliente no recibe ACK del servidor, por lo tanto nuevamente recurre al mecanismo del timeout pero aquí cabe destacar que el paquete que se envía tiene un sequence number que el servidor utiliza para identificar que el paquete es el que ya recibió y no lo confunda con un paquete nuevo.

Ese es el mecanismo básico, con dos sequence number nos alcanzará para poder utilizar este mecanismo de detección de repetidos.

TCP resumen

Siempre conviene manejarse en la unidad MSS (maximum segment size) que es el tamaño de los segmentos que se van a enviar en TCP.
Luego de establecerse el 3WH se comienza el envío de información en Slow start

SS ⇒ Crecimiento exponencial, $CWND_{n+1} = CWND_n + \#ACKs$
⇒ Básicamente si llega todo ok se duplica el tamaño de la ventana

Luego cuando CWND >= SSTHRESH se pasa a la etapa de Congestion Avoidance

CA ⇒ Aumenta en 1 cuando recibimos los ACKs de toda la rafaga (siempre tiene que ser número entero, ya que se mide en MSS, segmentos. En caso de no llegar todo y que todavía no se produzca algún RTO o Fast retransmit/ Fast recovery, se redondea para abajo)
⇒ $CWND_{n+1} = CWND_n + \#ACKs / CWND_n$ basicamente si no recibo todo ACK, no aumento el valor de la ventana.

Pérdida de paquetes

⇒ RTO:

- $SSTHRESH = CWND_N / 2$ (ventana antes de recalcular)
- $CWND_{N+1} = 1$ (Loss Window)

- Vuelvo a Slow start

⇒ Se pierde un solo paquete, y se obtienen 4 ACKs con el mismo número de secuencia. Se activa:

- **TAHOE, Fast retransmit**
  - Mismo protocolo que para RTO
- **RENO, Fast retransmit - Fast Recovery**
  - $CWND_{N+1} = CWND_N / 2$
  - $SSTHRESH = CWND_N / 2$
  - Se continua en Congestion avoidance
  - OJO, se transmite primero el paquete perdido con el valor recalculado de CWND (fast retransmit) y luego si se transmiten los otros paquetes como siempre con los valores recalculados (fast recovery).

IP Routing

Considere la siguiente tabla de ruteo

Network destination Netmask Interface

152.72.14.0	255.255.255.192	if1
158.93.224.0	255.255.224.0	if3
158.93.192.0	255.255.224.0	if3
158.34.39.64	255.255.255.192	if1
158.34.39.128	255.255.255.192	if1
158.92.192.0	255.255.240.0	if2
158.92.208.0	255.255.240.0	if2
158.92.224.0	255.255.240.0	if2
158.92.240.0	255.255.240.0	if2
158.92.241.0	255.255.240.0	If2 (dato con error)
158.93.0.0	255.255.255.0	if0
158.93.1.0	255.255.255.0	if1

Se pide:

a. Optimizar las entradas de la tabla

IF0:

158.93.0.0	255.255.255.0	if0

No se puede simplificar.

IF1:

152.72.14.0	255.255.255.192	if1
158.34.39.64	255.255.255.192	if1
158.34.39.128	255.255.255.192	if1
158.93.1.0	255.255.255.0	if1

- La primera entrada difiere en el primer byte MSB, por lo tanto no es contiguo con el resto.
- Entrada 2 y 3, tienen misma máscara y output port, quedaría ver si son prefijos contiguos:
  - 158.34.39.0100 0000
  - 158.34.39.1000 0000 ⇒ difieren en mas de 1 bit, no son contiguas
- La tercer entrada es el prefijo de red es distinto al resto por lo tanto no se puede simplificar

IF2:

158.92.192.0	255.255.240.0	if2
158.92.208.0	255.255.240.0	if2
158.92.224.0	255.255.240.0	if2
158.92.240.0	255.255.240.0	if2
158.92.241.0	255.255.240.0	If2 (dato con error)

La máscara y el output port de todas las entradas son iguales, hay que evaluar si son contiguos:

- 158.92.1100 0000.0
- 158.92.1101 0000.0
- 158.92.1110 0000.0
- 158.92.1111 0000.0

⇒ Se pueden agregar las 4 dado que se tienen las 4 combinaciones 00, 01, 10 y 11, quedaría ⇒ 158.92.1100 0000.0 ⇒ 158.92.192.0 con mascara 255.255.192.0

⇒ La última entrada es incorrecta dado que es más específica que la máscara permi
IF3:

158.93.224.0	255.255.224.0	if3
158.93.192.0	255.255.224.0	if3

La máscara y el output port de todas las entradas son iguales, hay que evaluar si son contiguos:

- 158.92.1110 0000.0
- 158.92.1100 0000.0 ⇒ Son contiguas

⇒ 158.93.192.0 255.255.192.0

Por lo tanto la tabla óptima sería:

152.72.14.0	255.255.255.192	if1
158.34.39.64	255.255.255.192	if1
158.34.39.128	255.255.255.192	if1
158.93.1.0	255.255.255.0	if1
158.92.192.0	255.255.192.0	if2
158.93.192.0	255.255.192.0	if3
158.93.0.0	255.255.255.0	if0

b. Configurar un default gateway, agregando una entrada cuya interfaz de salida sea <if0>

152.72.14.0	255.255.255.192	if1
158.34.39.64	255.255.255.192	if1
158.34.39.128	255.255.255.192	if1
158.93.1.0	255.255.255.0	if1
158.92.192.0	255.255.192.0	if2
158.93.192.0	255.255.192.0	if3
158.93.0.0	255.255.255.0	if0
0.0.0.0	0.0.0.0	if0

c. Para la tabla optimizada y con default gateway, determinar la interfaz de salida para la ip de destino

- 158.93.252.12

A priori vemos que las candidatas serían:

158.93.192.0	255.255.192.0	if3

Aplicando la mascara a la direccion ⇒ 158.93.192.0 ⇒ coincide con el prefijo y es el único ya que no hay ninguna otra coincidencia con otra entrada en la tabla que no sea la default gateway, por lo tanto por ser la más específica tomará el output port if3.

- 160.92.192.8

Observamos que el primer byte MSB no coincide con ninguno de la tabla, así que descartamos todos, por lo tanto tomará el default gateway, y saldrá por el output port if0.

Preguntas teóricas

Responda Verdadero o Falso. Justifique en caso de que la afirmación sea falsa.
Aclaración: Se debe responder correctamente TODOS los items del ejercicio.

a. Classful routing es el mecanismo por el cual se particionan las redes debido a que aprovecha mejor el espacio de direcciones.

Falso, dado que en classful routing se tienen clases de tamaño fijo y en este caso podríamos agotar rápidamente las direcciones disponibles, notar que la clase más chica que es la C provee 256 direcciones que para una casa de 2 dispositivos probablemente resulte excesivo.

b. Al subnetear un espacio de direcciones de clase C, un host puede tener asignada cualquiera de las 256 direcciones posibles.

Falso, no son 256 direcciones posibles dado que se tienen que asignar una para la dirección de red y otra para la dirección de broadcast.

c. Con classful routing las clases de red se pueden identificar sin necesidad de conocer la máscara.

Verdadero, la clase ya me dice la mascara.

d. Todas las direcciones asignadas a un mismo dispositivo deben pertenecer a la misma subred.

Falso, por ejemplo el router maneja múltiples direcciones y no todas pertenecen a la misma subred.

e. Todas las direcciones asignadas a un mismo dispositivo deben pertenecer a la subredes diferentes.

Falso, no hay ninguna restricción que imponga esto.

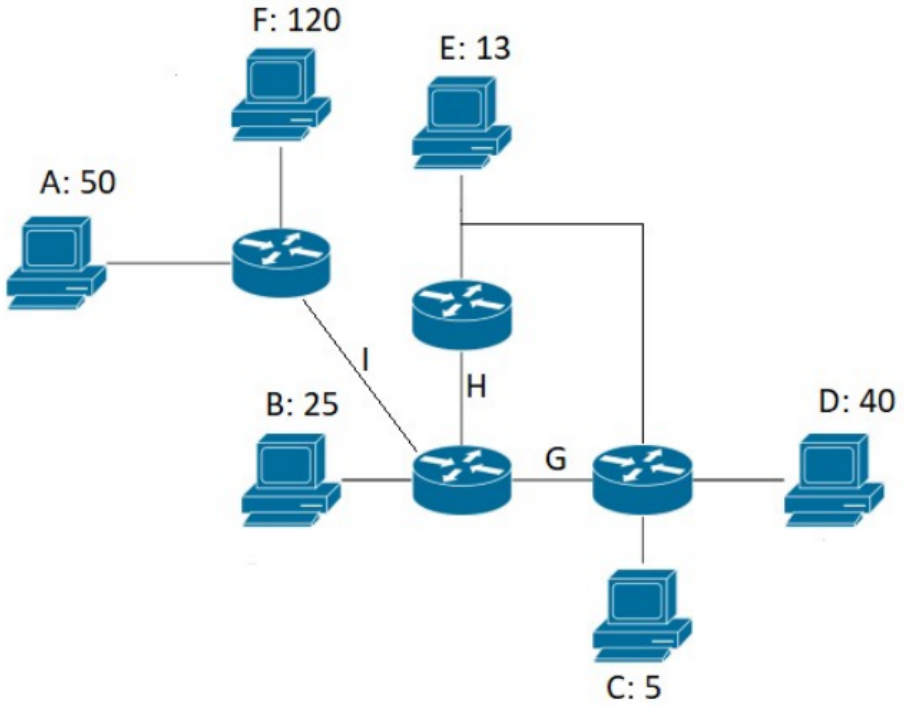
Subnetting

Dada la siguiente configuración de hosts y routers, y el espacio 196.132.0.0/23, se pide separar en subredes minimizando la cantidad de IPs sin usar.

Ante igualdad de condiciones para ubicar varias subredes:

1. **Asignar bloques utilizando los prefijos en orden de numeración ascendente**
   (Ej: si tenemos la opcion de usar 117.0.1.0/24 o 117.0.0.0/24, debemos utilizar primero el espacio de direcciones 117.0.0.0/24).
2. **Asignar bloques de direcciones priorizando las redes con mayor cantidad de hosts**
   (Ej: si se deben asignar dos bloques de 64 direcciones IP para dos subredes distintas S_x y S_y, donde x e y representan la cantidad de hosts de cada subred y con $32 < x < y < 64$, S_y debe asignarse en un espacio de direcciones de menor numeración).
3. **Si dos subredes necesitan la misma cantidad de IPs, ubicar primero la subred cuya letra viene primero en el abecedario**
   (Ej: si las redes P y J tienen necesitan un bloque de 32 IPs, ubicar primero la J y luego la P).

Este criterio arbitrario define una única resolución posible de la configuración. Cualquier otra solución será considerada incorrecta.



SubNet	# Host	# Router	Block	Prefix/Mask
A	50	1	64	196.132.0.128 /26
B	25	1	32	196.132.1.0 /27
C	5	1	8	196.132.1.64 /29
D	40	1	64	196.132.0.192 /26
E	13	2	32	196.132.1.32 /27
F	120	1	128	196.132.0.0 /25
G	0	2	4	196.132.1.72 /30
H	0	2	4	196.132.1.76 /30
I	0	2	4	196.132.1.80 /30

Preguntas teóricas

Responda Verdadero o Falso. Justifique en caso de que la afirmación sea falsa.

Aclaración: Se debe responder correctamente TODOS los items del ejercicio.

a. Sólo los routers pueden fragmentar paquetes en una red IPv4. **Falso, si del host al router el datagrama es > MTU, entonces también deberá fragmentar el paquete.**

b. El ensamblado de los fragmentos se realiza en el último router, antes de llegar al host receptor. **Falso, se realiza en el host de recepción, se ensambla y luego se pasa el datagrama a la capa superior. No se podría hacer en el router ya que los fragmentos podrían ir por caminos distintos.**

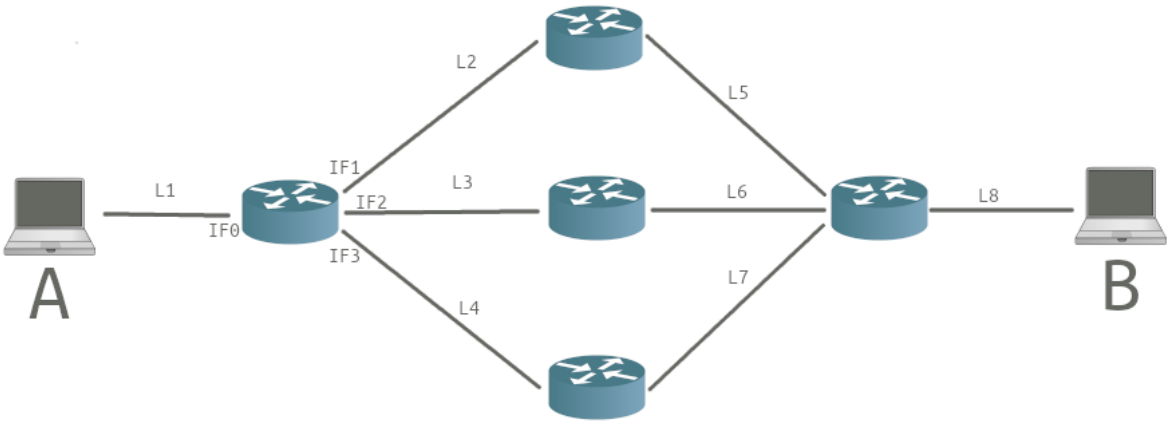
c. Para mejorar la eficiencia de la red IPv4 al utilizar TCP, cuando se fragmenta un paquete y uno de los paquetes fragmento se pierde, el host emisor sólo retransmite el paquete fragmento, en vez de retransmitir el paquete original. **Falso, TCP desconoce sobre el proceso de fragmentación, en caso de que se pierda un fragmento la capa de red terminará dropeando ese datagrama, TCP deberá reenviar el segmento completo.**

d. El mecanismo de fragmentación de IPv4 no introduce una vulnerabilidad en el protocolo. **Falso, se podría enviar un solo fragm ento para que el receptor reserve recursos y nunca enviar el resto y de esta manera agotar los recursos del receptor.**

Parcial - 2020 2C - 2op (tema 1)

Latencia

Se quiere calcular el RTT para medir la latencia entre dos host bajo la siguiente configuración:



Datos:
Packet Size = 1000 bytes

	L1	L2	L3	L4	L5	L6	L7	L8
Distancia	100 m	10 km	4 km	6 km	2 km	10 km	6 km	50 m
Ancho de Banda	10 Mbps	200 Mbps	200 Mbps	100 Mbps	100 Mbps	50 Mbps	200 Mbps	10 Mbps
Velocidad de Propagación	$1.7x\ 10^5$ km/s	2×10^5 km/s	2×10^5 km/s	$1.7x\ 10^5$ km/s	2×10^5 km/s	2×10^5 km/s	2×10^5 km/s	$1.7x\ 10^5$ km/s

1 Mbps = 10^6 bits / seg

El RTT se debe calcular utilizando un segmento de prueba de tamaño **1000 Bytes**, y será el mismo para la ida y la vuelta.
Tener en cuenta la asimetría de caminos siendo:

Ruta A → B: $L_1 \rightarrow L_4 \rightarrow L_7 \rightarrow L_8$
Ruta B → A: $L_8 \rightarrow L_6 \rightarrow L_3 \rightarrow L_1$

Los tiempos de encolado y procesamiento son despreciables.

Detallar los pasos del cálculo obtenido.

La latencia se puede calcular como la sumatoria de 4 componentes:

- Tiempo de inserción
- Tiempo de propagación
- Tiempo de procesamiento
- Tiempo de encolado

Luego habrá que tener en cuenta que el camino no es simétrico, por lo tanto habrá que primero calcular el camino de ida y luego la vuelta para obtener el RTT total. Por lo tanto:

A → B

tiempos de inserción:

- t1_ins = L / R = 1000 * 8 bits / 10 mbps = 8e-04 sec
- t4_ins = L / R = 1000 * 8 bits / 100 mbps = 8e-05 sec
- t7_ins = L / R = 1000 * 8 bits / 200 mbps = 4e-05 sec
- t8_ins = L / R = 1000 * 8 bits / 10 mbps = 8e-04 sec

tiempos de propagación:

- t1_prop = D / C = 0.1 km / 1.7e5 km/s = 5.88e-07 sec
- t4_prop = D / C = 6 km / 1.7e5 km/s = 3.53e-05 sec
- t7_prop = D / C = 6 km / 2e5 km/s = 3e-05 sec
- t8_prop = D / C = 0.05 km / 1.7e5 km/s = 2.94e-07 sec

B → A

tiempos de inserción:

- t8_ins = L / R = 1000 * 8 bits / 10 mbps = 8e-04 sec
- t6_ins = L / R = 1000 * 8 bits / 50 mbps = 0.00016 sec
- t3_ins = L / R = 1000 * 8 bits / 200 mbps = 4e-05 sec
- t1_ins = L / R = 1000 * 8 bits / 10 mbps = 8e-04 sec

tiempos de propagación:

- t8_prop = D / C = 0.05 km / 1.7e5 km/s = 2.94e-07 sec
- t6_prop = D / C = 10 km / 2e5 km/s = 5e-05 sec
- t3_prop = D / C = 4 km / 2e5 km/s = 2e-05 sec
- t1_prop = D / C = 0.1 km / 1.7e5 km/s = 5.88e-07 sec

Luego haciendo la sumatoria de las componentes podemos obtener la latencia por tramo y luego sumando los tramos obtendremos el RTT total. Por lo tanto los resultados serían:

camino ida 0.0017861764705882355 sec
camino vuelta 0.001870882 sec
RTT 0.0036570584705882354 sec

aprox 3.66 ms

ICMP

Responda Verdadero o Falso. *Justifique la respuesta*.

a. La herramienta ping utiliza UDP en su implementación.

Falso, no utiliza capa de transporte, por lo tanto no utiliza UDP. Ping utiliza ICMP que básicamente es un datagrama IP que en la parte de data en lugar de ir el paquete de la capa de transporte va el ICMP.

b. ICMP es un protocolo de transporte.

Falso, es un protocolo que funciona en la capa de red.

c. ICMP se elimina en IPv6

Falso, en IPV6 se utiliza ICMP, un ejemplo sería el mensaje de cuando el datagrama supera el MTU y se debe dropear, se envía un ICMP para informar el error.

d. El tráfico ICMP no se puede filtrar debido a que ICMP es un protocolo de diagnóstico y reporte de errores.

Falso, ICMP está contenido dentro del datagrama IP, si un firewall tiene que dropear paquetes por su dirección destino IP, tranquilamente lo podría hacer en el caso de ICMP, recordando al punto A, el mensaje ICMP viaja en la parte de datos del datagrama IP.

TCP/IP

Dada una conexión TCP **recién establecida** entre dos host para la cual el RTT es de *250 ms* y el MSS es de *1 KB*. Se desea transmitir un archivo de *20 KB*.
La red por la cual están conectados está congestionada, y se sabe que de enviar una ráfaga de **7 o más paquetes** , se pierden **todos** los paquetes de la ráfaga enviados por el cliente.

Otros datos:
- Initial Window = 1 KB
- Loss Window = 1 KB
- SS Threshold = 32 KB
- RWND = 10 KB
- Timeout equivale a 2 RTT

a. Calcular el tiempo que se tardará en enviar la totalidad del archivo.
Justificar el desarrollo de la transmisión utilizando la tabla propuesta.

Se tardan 10 RTT = 2,5 segundos en enviar la totalidad del archivo. La justificación se encuentra en la tabla y en el esquema temporal

b. ¿Qué sucedería en caso de que, cuando se produce la pérdida, lo que se pierde son los paquetes que responde el servidor en lugar de los enviados por el cliente?
Justificar el desarrollo de la transmisión utilizando la tabla propuesta.

En este caso, al no perderse los paquetes, se tardan 2 RTT menos (ver justificacion en tabla) => tiempo empleado = 8 RTT = 2 segundos

Punto a)

RTT	CWND	RWND	FlightSize	Recv Bytes	SSTH	Comments
1	1	10	1	1	32	Inicio en SS, llegan todos los paquetes por lo que duplico la ventana
2	2	10	2	3	32	Sigo en SS, llega todo bien, nuevamente duplico la ventana
3	4	10	4	7	32	Sigo en SS, llega todo bien, nuevamente duplico la ventana
4	8	10	8	7	32	Por consigna se pierde toda la rafaga.
5	RTO					Al terminar este ciclo se dispara el RTO
6	1	10	1	8	4	Seteo loss window, sigo en SS y SSTRESH = CWND / 2
7	2	10	2	10	4	Todo ok, sigo en SS, duplico ventana

8	4	10	4	14	4	Como CWND >= SSTHRESH paso a congestion avoidance
9	5	10	5	19	4	Llega todo OK, aumento ventana en 1
10	6	10	1	20	4	Fin de la transmision

Punto b)

RTT	CWND	RWND	FlightSize	Recv Bytes	SSTH	Comments
1	1	10	1	1	32	Inicio en SS, llegan todos los paquetes por lo que duplico la ventana
2	2	10	2	3	32	Sigo en SS, llega todo bien, nuevamente duplico la ventana
3	4	10	4	7	32	Sigo en SS, llega todo bien, nuevamente duplico la ventana
4	8	10	8	15	32	Por consigna se pierden todos los ACK
5	RTO					Al terminar este ciclo se dispara el RTO
6	1	10	1	15	4	Seteo loss window, sigo en SS y SSTRESH = CWND / 2. Al enviar el primer paquete de la ventana anterior, recibe el ACK15 por lo tanto se entera de que recibió la ventana anterior ok
7	2	10	2	17	4	Todo ok, sigo en SS, duplico ventana
8	4	10	3	20	4	Como CWND >= SSTHRESH paso a congestion avoidance

Responder Verdadero o Falso. *Justificar la respuesta.*

a. Los routers implementan todas las capas del modelo TCP/IP excepto la capa de aplicación.

Falso, los routers implementan hasta capa de red.

b. El protocolo de la capa de red proporciona una comunicación lógica entre procesos que se ejecutan en diferentes hosts mientras que un protocolo de capa de transporte proporciona comunicación lógica entre hosts.

Falso, la capa de red no conoce sobre procesos que se ejecutan en los hosts, de esto se encarga la capa de transporte utilizando el puerto de los hosts, que el puerto es el que se asocia a un proceso.

c. Un protocolo de capa de transporte puede ofrecer un servicio de entrega confiable aún cuando el protocolo de red subyacente no lo proporcione a nivel de capa de red.

Verdadero, así funciona en la realidad, ya que internet es best effort y no garantiza la entrega de paquetes.

d. Un protocolo de capa de transporte puede ofrecer garantías de delay aún cuando el protocolo de red subyacente no lo proporcione a nivel de capa de red.