

Deber 1

Gabriel Lora

Reader + Lab 4797

Pregunta 1

1. The bandwidth is 1.5 Mbps, and data packets can be sent continuously

Cada paquete tiene un tamaño de 1000 bytes. Por lo que el número de paquetes lo calculamos tal que $N^{\circ} \text{ packets} = \frac{(1000 \text{ KB})}{(1 \text{ Kb})} = 1000 \text{ paquetes}$

El tiempo requerido para enviar uno de los paquetes considerando el ancho de banda de 1500 kbps

El tamaño del paquete de 1 KB \rightarrow 8 Kbit. Tenemos que tiempo por envío: $\frac{8 \text{ kb}}{1500 \text{ kbps}} = 0,00533 \text{ s/paquete}$

Para todos los paquetes: $1000 \text{ paquetes} \times 0,00533 \text{ s/paquete} = 5,33 \text{ segundos}$

Tiempo inicial del handshaking: $2 \times \text{RTT} \approx 200 \text{ ms}$

Tiempo total: $200 \text{ ms} + 5,33 \text{ segundos} = 5,53 \text{ segundos}$

2. The bandwidth is 1.5 Mbps, but must wait 1 RTT sending each data packet

Tiempo para enviar un paquete \Rightarrow $\text{Packet size} = \frac{1 \text{ KB}}{\text{Bandwidth}} = \frac{1 \text{ KB}}{1.5 \text{ mbps}} = 0,008 \text{ segundos}$

Tiempo total del paquete: $2 \times \text{RTT} + \text{Tiempo de transmisión de paquetes} + 1 \times \text{RTT}$
 $= 2 \times 100 \text{ ms} + 0,008 \text{ s} + 100 \text{ ms} = 0,308 \text{ segundos}$

Tiempo total: $0,308 \text{ s} \cdot 1000 \text{ paquetes} = 308 \text{ segundos}$

3. Infinito bandwidth. we can transmit fine to be zero and up to 20 packets can be sent per RTT

Se enviarán 20 KB de paquetes en toda RTT. El tiempo para 20 paquetes es:

$$\begin{aligned} & 20 \text{ paquetes} \times \text{Packet trasmit time} + 2 \times \text{RTT} \\ & = 20 \times 0 + 2 \times 100 \text{ ms} = 200 \text{ ms} \end{aligned}$$

$$\text{Número total de RTTs} = \frac{\text{Total de paquetes}}{20} - \frac{1000 \text{ Paquetes}}{20} = 50 \text{ RTTs}$$

$$\text{Tiempo total} = (\text{Número de RTTs}) \times \text{RTT} + 2 \times \text{RTT} = 50 \times 100 \text{ ms} + 2 \times 100 \text{ ms} = 5200 \text{ ms} = 5.2 \text{ segundos}$$

④ Bandwidth infinito, during First RTT we can send 1 packet, during the Second two packets, and so on

El crecimiento del algoritmo tiene la forma $2^{(n-1)}$ y sabemos que 1000 tienen 1000 paquetes por lo que tenemos que $2^{n-1} = 1000$, despejando n tenemos que $= (n-1) \ln(2) = \ln(1000)$

$$\Rightarrow n = \frac{\ln(10)}{\ln(2)} + 1$$

$$n = 10.96 \approx n = 11$$

Por lo que se necesitan 11 RTT para transmitir el archivo.

El tiempo total sería de: $100 \text{ ms} \times (2^{11-1}) = 1,024 \text{ segundos}$

Pregunta 2

1) Otra propiedad considerando que las direcciones son únicas puede servir como localizadores ya que proveen de información sobre como los datos deben enrutarse. De igual manera lograr una jerarquía a nivel de direcciones.

2) Cuando las redes utilizan asignaciones dinámicas de direcciones, donde hay una probabilidad de que se asigne a dos dispositivos la misma dirección IP al mismo tiempo al solicitar una nueva dirección IP al mismo tiempo

Pregunta 3

1. Open a File: en este caso sería una operación sensible a los retrasos (Delay) ya que esta operación implica crear una conexión con el servidor de archivos y mantenerla hasta que el sistema responda y abra el archivo, mas no lo lea. Por lo que el uso de ancho de banda en esta operación es bajo.

2. Read The contents of a file: esta operación requiere de más ancho de banda ya que es una transferencia de información entre el servidor de archivos y el usuario, por lo que requiere de una baja latencia de red para completar la operación.

3. List the contents of a directory: Si son muchos archivos, es una operación sensible a los retrasos (Delay), y al finalizar listarlos, más no abrirlos no requiere de una gran cantidad de ancho de banda para poder completar la operación.

4. Display the attributes of a file: Mostrar la información de un fichero. Una operación sensible a los retrasos (Delay) ya que lo contenido de datos que son transferidos son pequeños, por lo que más importa es la rapidez con la que se muevan dichos datos.

Pregunta 4 Dados enviados: 1×10^6 bytes . Consideremos que el overhead es de 100 bytes . Data size

El overhead total sería: $100 \text{ bytes} + \text{Datos enviados} + \text{loss bytes}$

$$\rightarrow \text{Total} = 100 \text{ bytes} + \left(\frac{1 \times 10^6 \text{ bytes}}{\text{Data size}} \right) + \text{Data size}$$

Data size	Total
1000	101000
5000	25000
10,000	20000
20,000	25000

, l óptimo es de tamaño 10,000 bytes ya que tiene menor perdida total

Preguntas

(A) función polinomial $x^3 + 1$ tiene un valor de $K = 3$, nuestro generador para la función es 1001.

Proceso de división polinomial larga para este mensaje 11.

$$M(x) (1 \cdot x^7) + (1 \cdot x^6) + 0 + 0 + (1 \cdot x^3) + 0 + 0 + (1 \cdot x^0)$$

$$M(x) = x^7 + x^6 + x^3 + 1,$$

Lo multiplicamos por x^K , donde $K=3$ tal que = nuestro mensaje 01 11001001000

Proceso de división:

1001	1	1	0	0	1	0	0	1	0	0
1	0	0	1	1						
	1	0	1	1						
		1	0	0						
			1	0						
				1	1	0				
					1	0	1			
						1	0	1		
							1	0	1	
								1	1	
									1	1

Tenemos como residuo 11 por lo que restar el mensaje de manera que el mensaje sea: 11001001011
donde los 3 últimos dígitos son los CRC

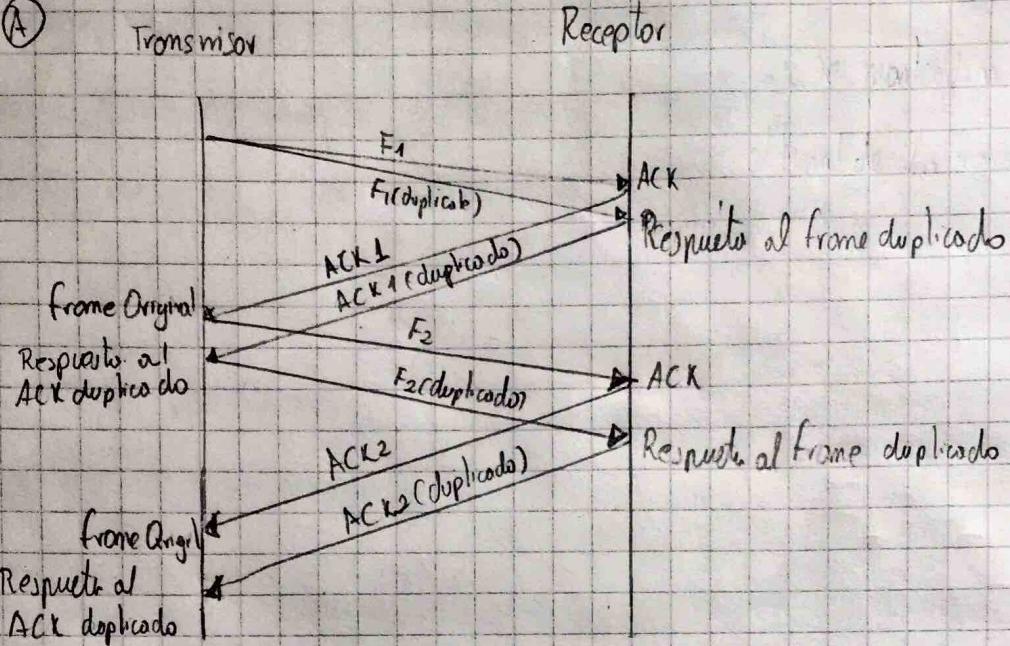
(B) Ahora la señal es: 01001001011

Añadiendo tenemos que = 1001 1010010001011
 $\downarrow 1001$
 0, 0 1 0 1 1
 1 0 0 1
 1 0

El residuo no es cero por lo que existe un error de bit durante la transmisión

Problema 6

(A)

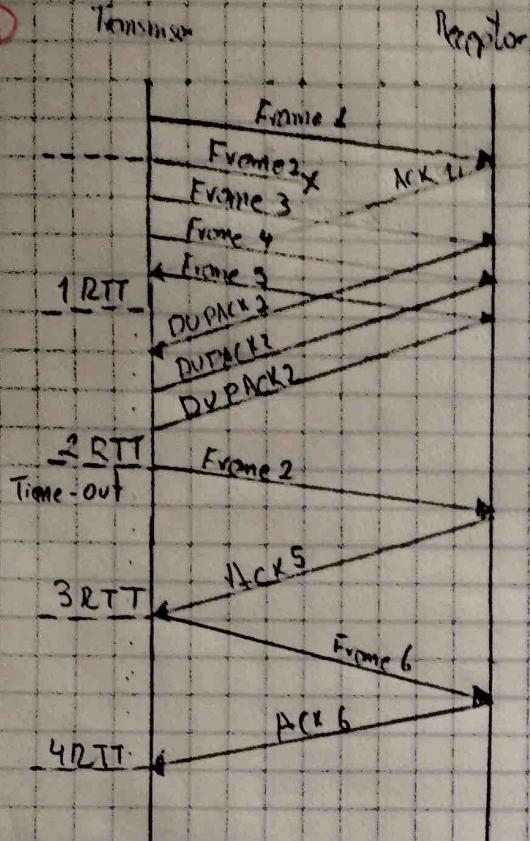


Como se puede ver los ACK duplicados retransmite inmediatamente los frames y seguirá así hasta que se termine la transmisión

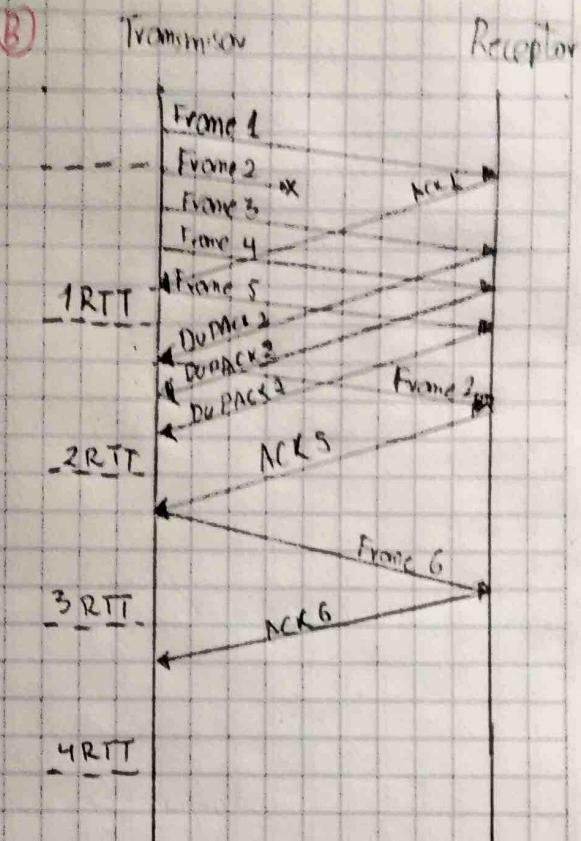
(B) Un caso donde un nodo A envía un paquete al nodo B, y al recibir este paquete el nodo B no envía de vuelta el ACK de respuesta debido a un retraso o error. Por lo que el nodo A al no recibir el ACK dentro del intervalo de tiempo de espera, vuelve a retransmitir el mismo paquete y nuevamente el nodo B no retorna de nuevo el ACK al nodo A. Por lo que el ciclo continua. Probando gestión de red y provocando el "Trigger sorceres"

Pregunta 7

(A) Transmisor



(B)



∴ Ciertamente el tiempo de transmisión se reduce porque el transmisor no tiene que esperar todo el time-out antes de volver a mandar el Frame 2 (esta vez lo hace apenas recibe el primer dupack)