Universidad San Francisco de Quito

Redes

Deber 1

Nombre: Mateo Ruiz

Codigo Banner: 00212195

NRC: 4005

1. Calculate the total time required to transfer a 1000-KB file in the following cases, assuming an RTT of 100 ms, a packet size of 1 KB data, and an initial 2×RTT of "handshaking" before data is sent:

Primero, procedemos a calcular el número de paquetes de datos necesarios:

$$1000KB = 1000 \times 8 = 8000Kb$$

$$1KB = 1 \times 8 = 8Kb$$

Numero de paquetes: $\frac{8000}{8} = 1000$

2xRTT = 200ms

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

Para este caso:

$$Tiempo\ de\ envio\ de\ paquete = \frac{1000\ bytes \times 8\ bits}{1.5Mbps} = 0.00533\ segundos$$

$$Tiempo\ total = (1000\ paquetes) \times (0.00533\ seg) + 0.2\ segundos = 5.53\ seg$$

2. **The bandwidth is 1.5 Mbps, but after we finish sending each data packet we must wait one RTT before sending the next.**

$$Tiempo\ de\ envio\ de\ paquete = \frac{1000\ bytes \times 8\ bits}{1.5Mbps} = 0.00533\ segundos$$

$$TL = (1000\ paquetes) \times (0.00533\ seg) + 0.1\ segundos) + 0.2seg = 105.53\ seg$$

3. **The bandwidth is "infinite," meaning that we take transmit time to be zero, and up to 20 packets can be sent per RTT.**

   En este caso, dado a que se nos dice que el ancho de banda es infinito, no nos preocupamos del tiempo para enviar cada paquete, y como podemos enviar 20 paquetes por cada 100ms entonces:

$$Tiempo\ total = \frac{1000\ paquetes}{20\ paquetes/RTT} = 50RTT$$

$$T = 50RTT \times 0.1s = 5s + 0.2s = 5.2seg$$

4. **The bandwidth is infinite, and during the first RTT we can send one packet, during the second RTT we can send two packets, during the third we can send four, and so on.**

En este caso, dado a que es creciente el envio de paquetes por RTT y va 1 +2 + 4 y asi sucesivamente hasta ser 2^n podemos decir que la ecuación es:

$$S(n) = 2^{n+1} - 1\ paquetes$$

En este caso cuando n = 9 podemos ver que ya sean enviado los 1000 paquetes requeridos, por tanto, el ultimo lote:

$$9RTT \times RTT = 9 \times 0.1 = 0.9$$

$$T = 0.9 + 0.2 = 1.1\ seg$$

2. One property of addresses is that they are unique; if two nodes had the same address it would be impossible to distinguish between them. What other properties might be useful for network addresses to have? Can you think of any situations in which network addresses might not be unique?

En el remote caso de que dos nodos compartan las mismas direcciones en la red, esto ocasionaría un conflicto donde ninguno de los dos nodos pueda comunicarse correctamente, por tanto no seria imposible distinguirlos pues serían los únicos dispositivos que no se comunican adecuadamente.

Una propiedad de las direcciones web es que son prácticas, es decir que pueden ser recordadas y escritas por un ser humano.

Puede existir el caso en el que se puede duplicar la dirección en diferentes redes aisladas, por tanto, no habría ningún problema puesto a que no existe comunicación entre ellas. De lo contrario habría un conflicto

3. For each of the following operations on a remote file server, discuss whether they are more likely to be delay sensitive or bandwidth sensitive:

1. Open a file

Es más sensible retraso que al bando de ancha porque la cantidad de datos usada es pequeña durante el proceso.

2. Read the contents of a file

Sensible a la banda ancha porque leer un archivo depende de la cantidad de datos intercambiados y la velocidad de transferencia.

3. List the contents of a directory

Mas sensible al retraso ay que listar los contenidos de un directorio ya dependería del tiempo de respuesta del servidor, sin embargo, si el directorio es demasiado grande podría ser sensible para la banda ancha.

4. Display the attributes of a file

Es más sensible al retraso ya que al no depender de la cantidad de datos, solo dependería del tiempo de respuesta del servidor.

4. Suppose that a certain communications protocol involves a per-packet overhead of 100 bytes for headers. We send 1 million bytes of data using this protocol; however, when one data byte is corrupted, the entire packet containing it is lost. Give the total number of overhead + loss bytes for packet data sizes of 1000, 5000, 10000, and 20000 bytes. Which of these sizes is optimal?

En este caso encontramos que el numero de paquetes equivale a:

$$Numero\ de\ paquetes\ =\ 1 \times 10^6 / Tamaño\ de\ datos$$

La sobrecarga estaría dada por:

$$Sobrecarga\ total = 100\ bytes\ \times Numero\ de\ paquetes = 100 \times \frac{1 \times 10^6}{Tamano} de\ datos$$

Y la perdida equivaldría al tamaño de datos y el tamaño total quedaría:

$$Tamano\ total = Sobrecargatotal + perdida$$
$$= 100 \times \frac{1 \times 10^6}{Tamano\ de\ datos} + Tamano\ de\ datos$$

Por tanto:

1000 bytes:

$$= \frac{1 \times 10^8}{1000} + 1000 = 101000$$

5000 bytes:

$$= \frac{1 \times 10^8}{5000} + 5000 = 25000$$

10000 bytes:

$$= \frac{1 \times 10^8}{10000} + 10000 = 20000$$

10000 bytes:

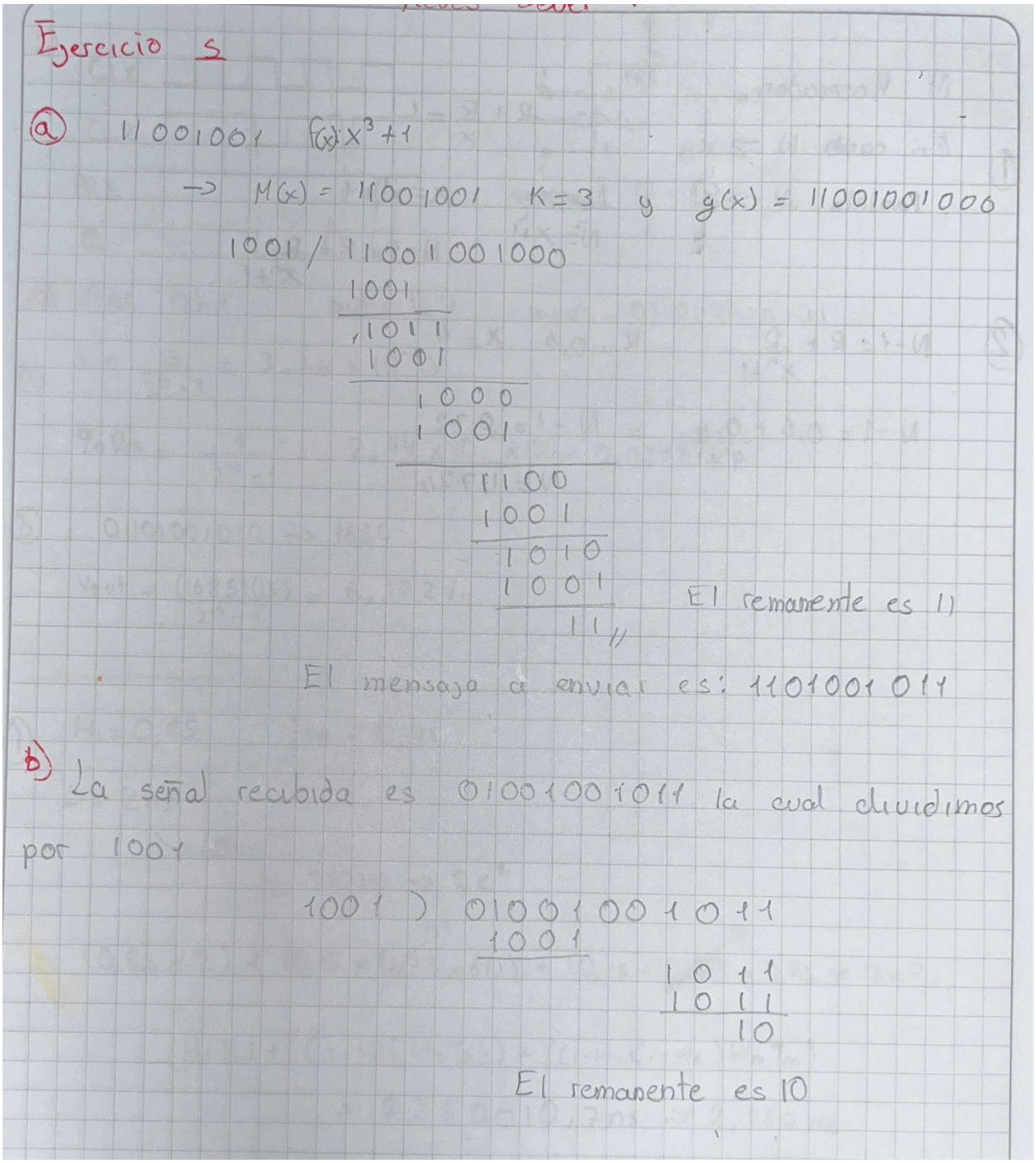
$$= \frac{1 \times 10^8}{20000} + 20000 = 25000$$

En este caso, podemos deducir que 10000 bytes es el mas optimo por ser el menor

5. Suppose we want to transmit the message 11001001 and protect it from errors using the CRC polynomial x3 + 1

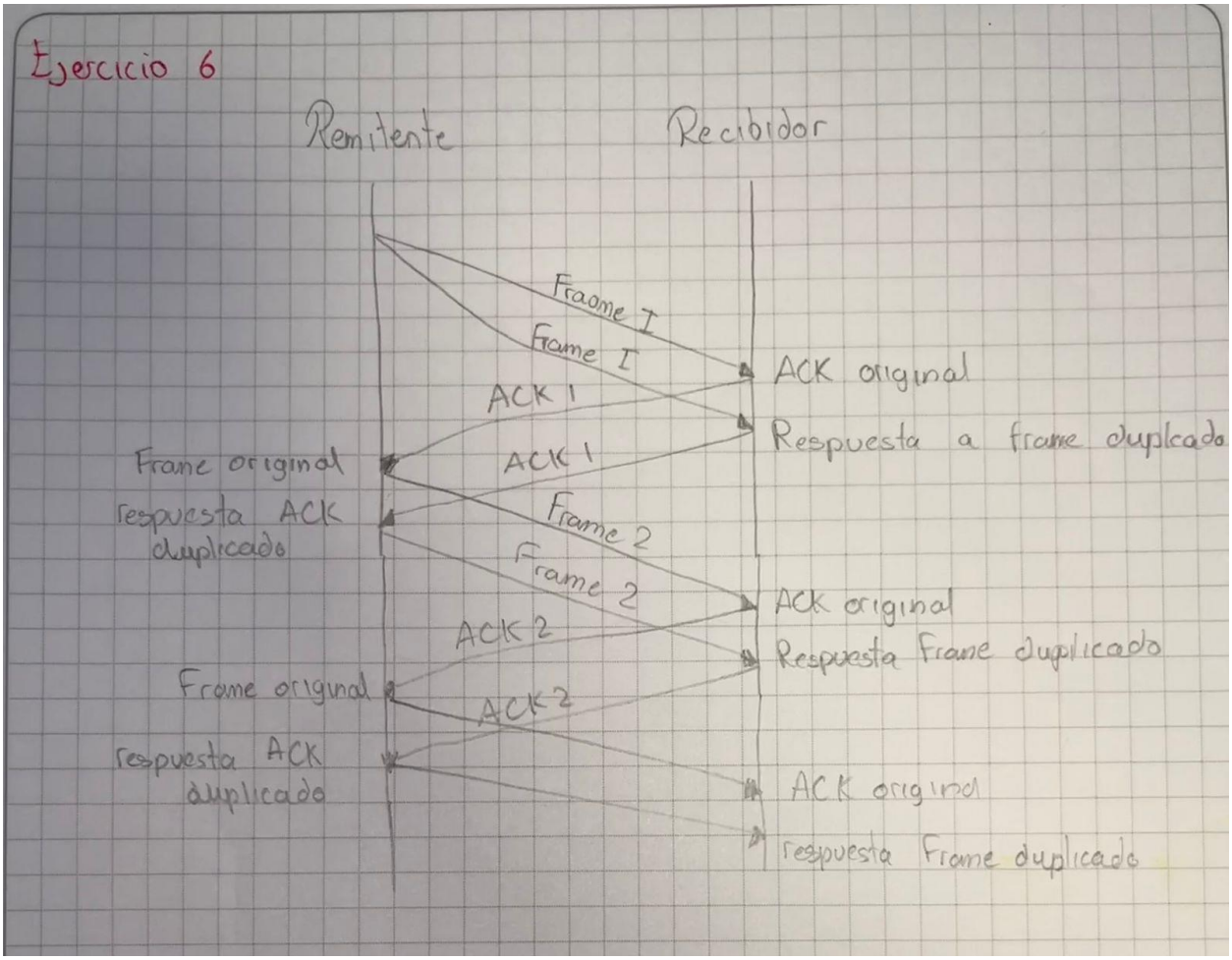
1. Use polynomial long division to determine the message that should be transmitted.

2. Suppose the leftmost bit gets inverted in transit. What is the result of the receiver's CRC calculation?



6. In stop-and-wait transmission, suppose that both sender and receiver retransmit their last frame immediately on receipt of aduplicate ACK or data frame; such a strategy is superficially reasonable because receipt of such a duplicate is most likely tomean the other side has experienced a timeout.

1. Draw a timeline showing what will happen if the first data frame is somehow duplicated, but no frame is lost. How long will the duplications continue? This situation is known as theSorcerer's Apprentice bug.

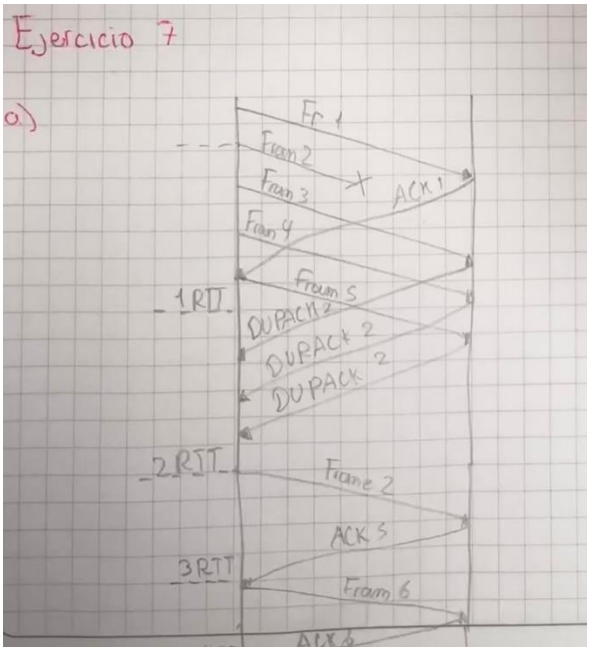


2. Suppose that, like data, ACKs are retransmitted if there is no response within the timeout period. Suppose also that bothsides use the same timeout interval. Identify a reasonably likely scenario for triggering the Sorcerer's Apprentice bug.

Por este bug, el ACK anterior y el marco de datos duplicado deben cruzarse en algún lugar de la red. En este caso tanto como el remitente y recibidor retransmiten al mismo tiempo, lo suficientemente sincronizadas como para cruzarse. Sin embargo, con el bug del aprendiz del brujo puede seguir ejecutándose de manera confiable en las condiciones adecuadas.

7. Draw a timeline diagram for the sliding window algorithm with SWS = RWS = 4 frames for the following two situations. Assume the receiver sends a duplicate acknowledgement if it does not receive the expected frame. For example, it sends DUPACK[2] when it expects to see FRAME[2] but receives FRAME[3] instead. Also, the receiver sends a cumulative acknowledgment after it receives all the outstanding frames. For example, it sends ACK[5] when it receives the lost frame FRAME[2] after it already received FRAME[3], FRAME[4], and FRAME[5]. Use a timeout interval of about 2 × RTT.

1. Frame 2 is lost. Retransmission takes place upon timeout (as usual).



2. Frame 2 is lost. Retransmission takes place either upon receipt of the first DUPACK or upon timeout. Does this scheme reduce the transaction time? Note that some end-to-end protocols (e.g., variants of TCP) use a similar scheme for fast retransmission.