Sthefano Ulloa (215689)

03-03-2023

Tarea 1

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:
    1. The bandwidth is 1.5 Mbps, and data packets can be sent continuously.
    2. The bandwidth is 1.5 Mbps, but after we finish sending each data packet we must wait one RTT before sending the next.
    3. The bandwidth is "infinite," meaning that we take transmit time to be zero, and up to 20 packets can be sent per RTT.
    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.

Latency = Propagation + Transmit + Queue

Propagation = Distance / Speed of light

Transmit = Size / Bandwidth

Transfer time = RTT + Bandwidth^{-1} (Transfer size)

Throughput = $\frac{\text{Transfer size}}{\text{Transfer time}}$

Latency = Propagation + $\frac{\text{Size}}{\text{Bandwidth}}$

1) Latency = 50 ms + $\frac{1000 \text{ KB}}{1.5 \text{ Mbps}}$ + 2 (100 ms)

Si el RTT es 100, la propagación (a un solo lado) debería ser la mitad

$= 50 \cdot 10^{-3} + \frac{\cancel{10^6} \cdot 8}{1.5 \cdot \cancel{10^6}} + 2(100 \cdot 10^{-3})$

$= 0.05 + 5.33 + 2 \cdot 0.1$

$= 0.05 + 5.33 + 0.2$

$= 5.58$ s

2) 5.58 s + 999 RTT

+ 999 · 100 ms

+ 99.9 = 105.48 s

3) Como tenemos 1000 paquetes, y podemos enviar 20 por RTT

$\frac{1000}{20} = 50$ RTT

=> Latency = 50 RTT + 2RTT + 0.25 s

52 RTT + 0.25 s

52 · 0.1 s + 0.25 s = 5.45 s

4) $1 + 2 + 4 + \dots + 2^n = 2^{n+1} - 1$

$$2^{n+1} - 1 = 1000$$

$$2^{n+1} = 1001$$

$$(n+1) \ln(2) = \ln(1001)$$

$$n = \frac{\ln(1001)}{\ln(2)} - 1 = 8.96 \approx 9$$

En 9 RTTs podemos enviar 1023 paquetes

$$\text{Latency} = 9\,RTT + 2\,RTT + 0.25\,s$$

$$= 11\,RTT + 0.25\,s$$

$$= 11 \cdot 0.1\,s + 0.25\,s = 1{,}35\,s$$

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 caso de direcciones IP, una característica fundamental es que en una packet-switched network, se pueden rutear. Es decir, los paquetes deben poder llegar a los distintos hosts conectados a la red o subredes.
- Además son jerárquicas (hierarchy?) como 192.168.1.2

  2 → hosts
  1 → subred ?

- En el caso de las URLs deben tener cierta información que haga fácil entenderlas como dominio (Organización), sección, archivo, etc (usfq / redes / index.html)
- Las IP no son únicas entre LANs. Por ejemplo router de la oficina tiene dispositivos con 192.168.1.25 y router de la casa tiene un dispositivo con la misma IP.
- Si no se tiene una IP privada, múltiples instalaciones de internet podrían compartir la misma IP pública.
- En una VPN la ip pública de todos los dispositivos podría ser la misma

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
   2. Read the contents of a file
   3. List the contents of a directory
   4. Display the attributes of a file

1) (Asumiendo open como la system call de ese nombre)
Abrir un archivo genera un file descriptor que apunta al archivo. El tamaño de este es pequeño. Por lo que más bien lo significativo sería el delay hasta que el servidor reciba la orden.

2) A diferencia que lo anterior, leer el archivo dependería más del bandwidth ya que el delay generalmente afecta en milisegundos y el especialmente si el archivo es grande el transmit depende inversamente del bandwidth

$\text{transfer} = \frac{\text{size}}{\text{bandwidth}} + RTT \Rightarrow$ bandwidth pequeño, transfer time grande.
bandwidth grande, transfer time pequeño.

Es decir depende más del bandwidth

3) Similar al open, la data transmitida es pequeña por lo que depende más del delay

4) Los atributos o metadata es pequeño. Depende más del delay

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?

$$\# \text{Paquetes} = \frac{10^6}{\text{packet size}}$$

$$\text{Overhead} = 100\text{ B} \cdot \# \text{Paquetes}$$

$$\text{Perdida} = \text{Packet size}$$

$$\text{Total} = \text{Overhead} + \text{Perdida} = 100\text{ B} \cdot \frac{10^6}{\text{packet size}} + \text{Packet size}$$

Packet size	total
1000 B	101 000
5000 B	25 000
10 000 B	20 000 ← tamaño óptimo
20 000 B	25 000

5. Suppose we want to transmit the message 11001001 and protect it from errors using the CRC polynomial $x^3 + 1$

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?

1) 11001001 mod 1001 = 011

(Use una calculator.net)

=> 11001001011

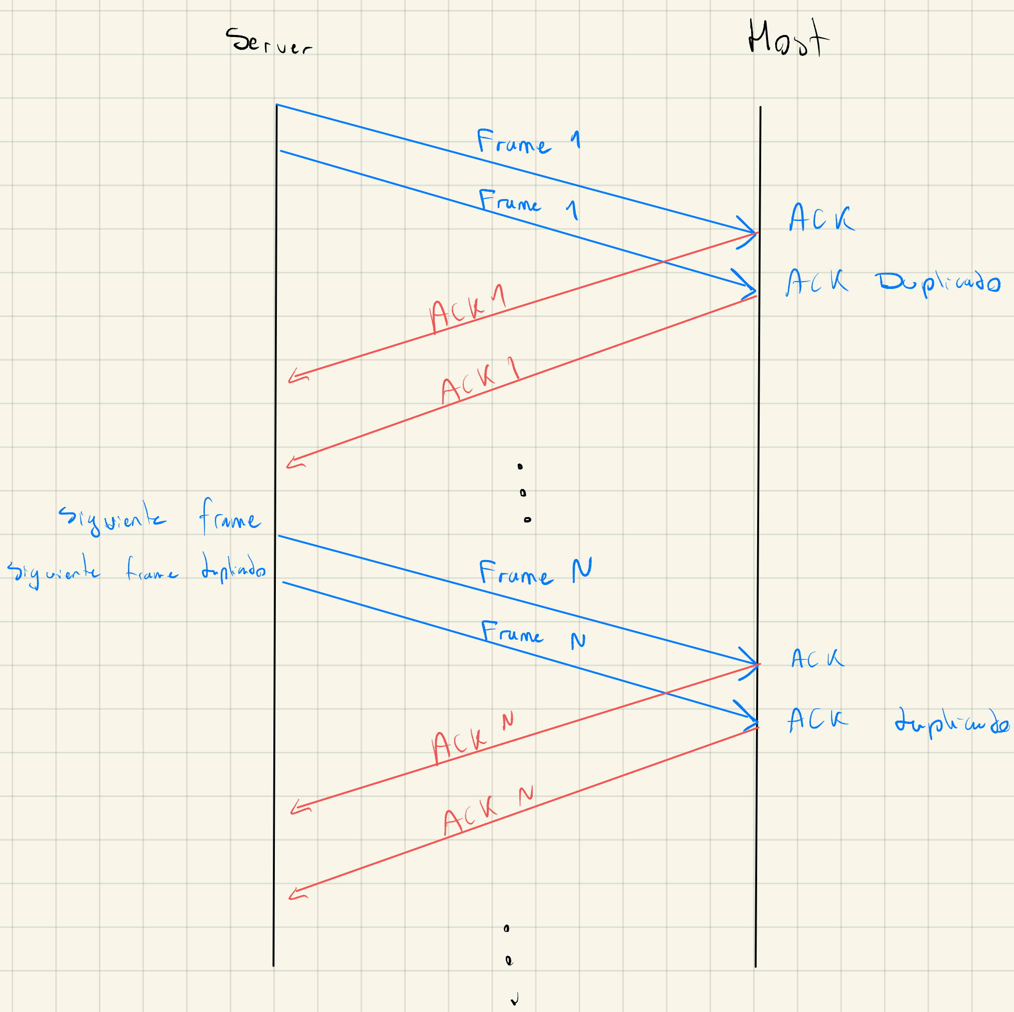
2) 01001001011 mod 1001 = 10 ≠ 0 => error

6. In stop-and-wait transmission, suppose that both sender and receiver retransmit their last frame immediately on receipt of adupliicate 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 theduplications continue? This situation is known as theSorcerer's Apprentice bug.

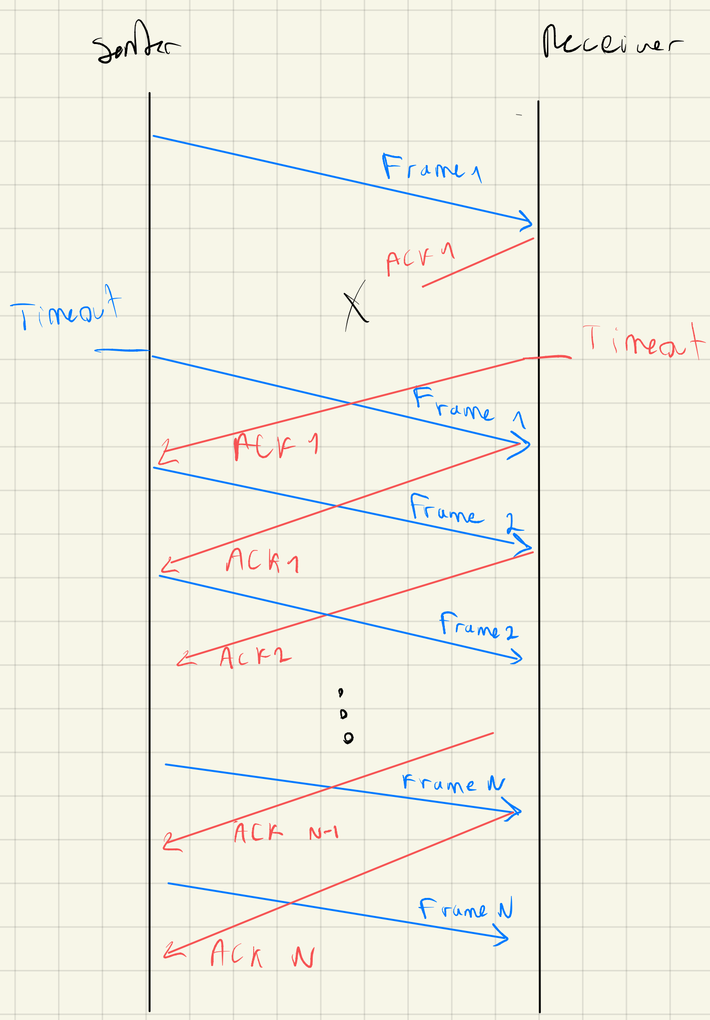
   2. Suppose that, like data, ACKs are retransmitted if there is no response within the timeout period. Suppose also that bothsides use the same timeout interval. Identify a reasonably likely scenario for triggering the Sorcerer's Apprentice bug.

1)



Basicamente se van a enviar frames duplicados hasta el final de la transmisión

2)

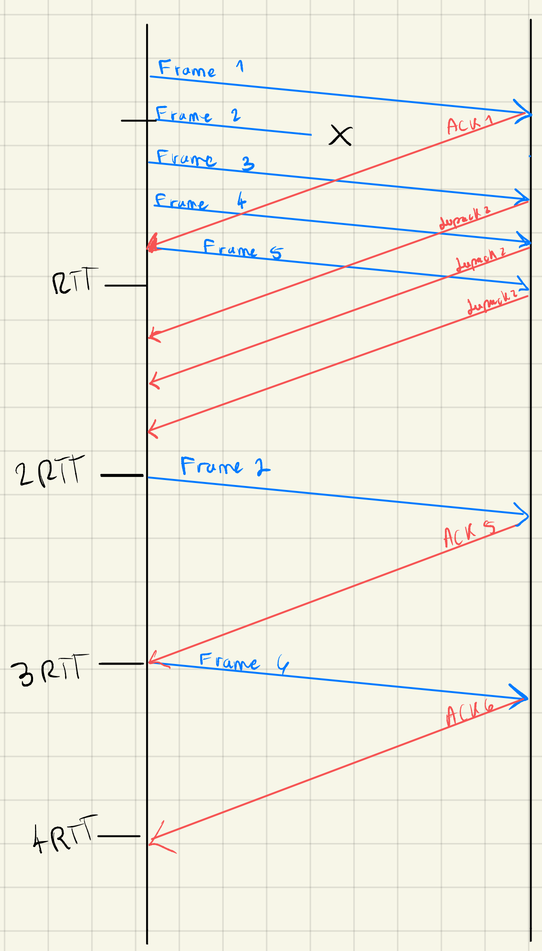


Si un acknoletgment se pierde y el sender y receiver de alguna forma se sincronizan para volver a enviar el frame y el ACK, se empezaran a enviar frames y ACKs duplicados (Sorcerer's appretince bug)

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.

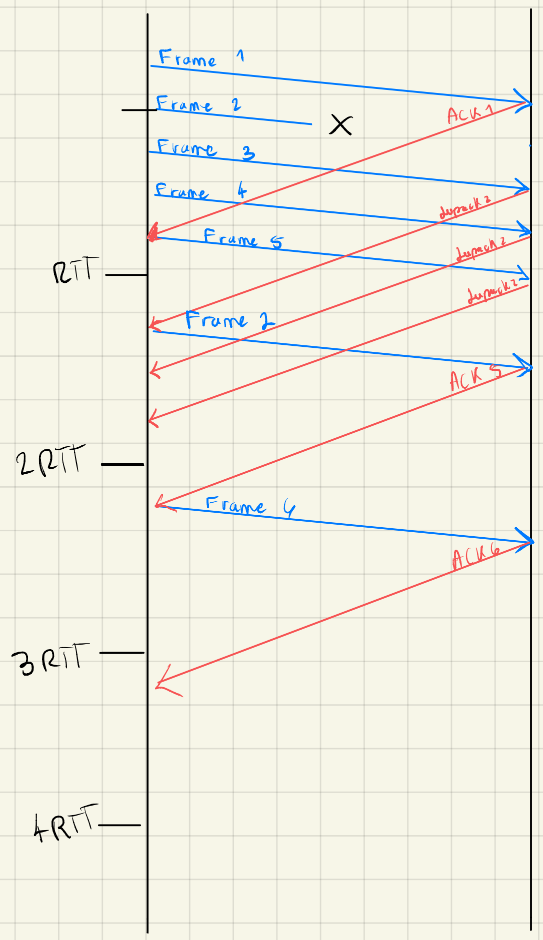
    1. Frame 2 is lost. Retransmission takes place upon timeout (as usual).
    2. Frame 2 is lost. Retransmission takes place either upon receipt of the first DUPACK or upon timeout. Does this scheme reduce the transaction time? Note that some end-to-end protocols (e.g., variants of TCP) use a similar scheme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

1)



Caso 1 = 4 RTT

2)



Caso 2 ≈ 3 RTT

Mejora con respecto al caso 1