

Boletín de Exercicios

- Supoñamos unha mensaxe de 10^7 bits que se quere enviar entre dous hosts entre os que hai 2 routers intermedios. Supoñamos que os enlaces son de 2 Mbps (millóns de bits por segundo). Obter o tempo de transmisión nos dous casos seguintes:

- A mensaxe sen segmentar
- Segmentada en 2500 paquetes de 4000 bits cada unha

Considerar que os routers son dispositivos de almacenamento e reenvío. Utilizar diagramas de tempo.

Vamos ter en conta so o tempo de almacenamento e reenvío. Sabendo que temos enlaces de 2 Mbps, os tempos necesarios para almacenar e reenviar a mensaxe completa e un dos paquetes son, respectivamente,

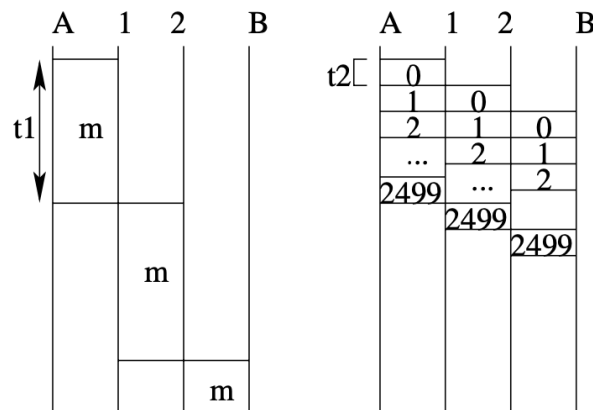
$$t_1 = \frac{10^7 \text{ bits}}{2 \text{ Mbps}} = 5 \text{ sg} \quad (1)$$

$$t_2 = \frac{4000 \text{ bits}}{2 \text{ Mbps}} = 2 \text{ msg} \quad (2)$$

Tendo en conta o esquema da seguinte figura, os retardos de transmisión nos dous casos, son:

$$t_{\text{mensaxe}} = 3t_1 = 15 \text{ sg} \quad (3)$$

$$t_{\text{paquetes}} = 2500t_2 + 2t_2 = 5 + 0,004 = 5,004 \text{ sg} \quad (4)$$



Se supoñemos que aos paquetes se lles engade unha cabeceira de 40 bytes (a cabeceira IP é de 20 bytes de tamaño mínimo e 60 bytes de tamaño máximo), cada paquete terá 4320 bits e o resultado será:

$$t_2 = \frac{4320 \text{ bits}}{2 \text{ Mbps}} = 2,16 \text{ msg} \quad (5)$$

$$t_{\text{paquetes}} = 2500t_2 + 2t_2 = 5,400 + 0,00432 = 5,40432 \text{ sg} \quad (6)$$

2. Supoñamos un paquete de lonxitude L que ten o seu orixe no host A e que viaxa a través de tres enlaces, conectados mediante dous dispositivos de conmutación de paquetes, ata un host destino B. Supoñamos que d_i , v_i e R_i son a lonxitude, a velocidade de propagación e a velocidade de transmisión do enlace i , para $i = 1, 2, 3$, respectivamente. Cada dispositivo de conmutación de paquetes retarda cada paquete un tempo t_{proc} .

- Supoñendo que non se produce retardo nas colas, cal é o retardo total de extremo a extremo en función dos parámetros d_i , v_i , R_i (con $i = 1, 2, 3$) e L .
- Supoñamos que a longtitude do paquete é de 1.500 bytes, a velocidade de propagación dos enlaces é $2,5 \cdot 10^8$ m/s, a velocidade de transmisión nos tres enlaces é de 2 Mbps, o retardo de procesamento en cada conmutador de paquetes é de 3 milisegundos, a lonxitude do primeiro enlace é de 5.000 Km, a do segundo 4.000 Km e a do último 1.000 Km. Para estes valores, cal é o retardo total de extremo a extremo?
- Debuxa a situación do apartado anterior nun diagrama de tempo.

a)

$$t_{\text{extr-extr}} = \frac{L}{R_1} + \frac{L}{R_2} + \frac{L}{R_3} + \frac{d_1}{v_1} + \frac{d_2}{v_2} + \frac{d_3}{v_3} + t_{\text{proc}} + t_{\text{proc}}$$

b)

$$t_{\text{tran}_1} = \frac{L}{R_1} = \frac{1500 \cdot 8}{2 \cdot 10^6} = 6 \text{ ms} = t_{\text{tran}_2} = t_{\text{tran}_3}$$

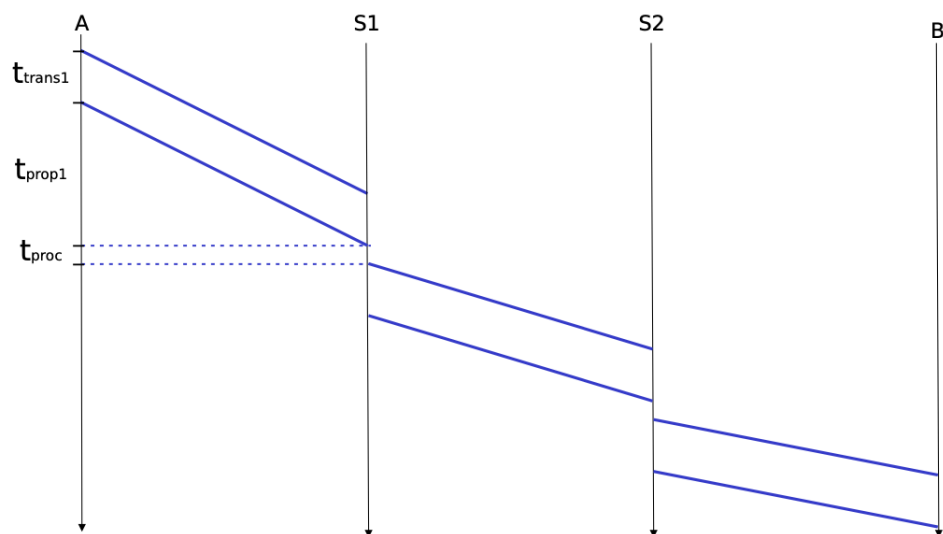
$$t_{\text{prop}_1} = \frac{d_1}{v_1} = \frac{5000 \cdot 10^3}{2,5 \cdot 10^8} = 20 \text{ ms}$$

$$t_{\text{prop}_2} = \frac{d_2}{v_2} = \frac{4000 \cdot 10^3}{2,5 \cdot 10^8} = 16 \text{ ms}$$

$$t_{\text{prop}_3} = \frac{d_3}{v_3} = \frac{1000 \cdot 10^3}{2,5 \cdot 10^8} = 4 \text{ ms}$$

$$t_{\text{extr-extr}} = 6 + 6 + 6 + 20 + 16 + 4 + 3 + 3 = 64 \text{ ms}$$

c) Diagrama de tempo.



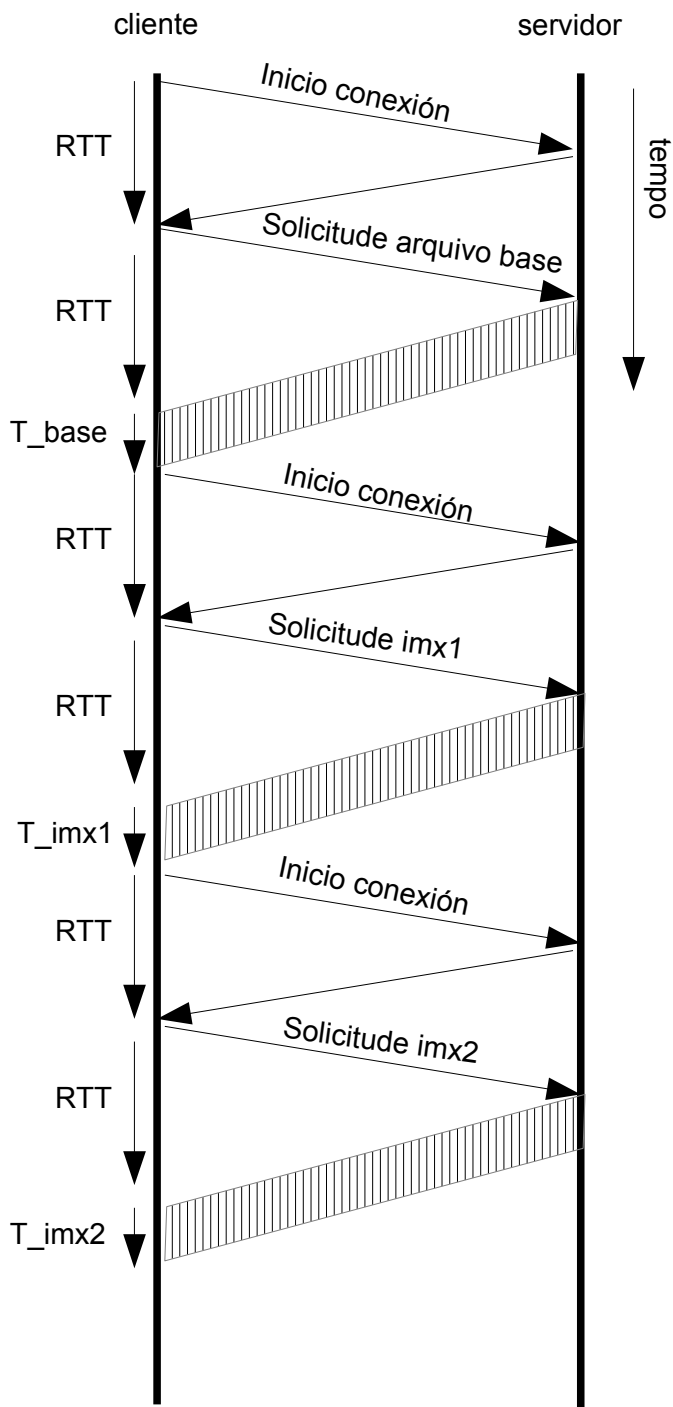
3. Mostra e calcula, utilizando diagramas de liñas de tempo, o tempo de transferencia dunha páxina web con dúas imaxes (documento base e dous obxectos) nos diferentes tipos de conexións que permite o protocolo HTTP. Supón que esta páxina é primeira que se solicita ao servidor web.
- a) Conexións non persistentes
 - Serie
 - Paralelo
 - b) Conexións persistentes
 - Sen entubamento
 - Con entubamento

Sobre o tempo de desconexión: Desde o punto de vista do protocolo HTTP pódese considerar o seguinte nas desconexións: cando o servidor remata de enviar todos os arquivos (ou 1 arquivo, en conexións non persistentes) manda a mensaxe de peche de conexión inmediatamente. Cando o cliente recibe o arquivo ao final recibe a mensaxe de desconexión e pecha a conexión. Desta forma a fase de desconexión queda oculta dentro do tempo de transferencia do arquivo e non se ten en conta. Esta é a forma en que está explicado no libro é moitos exercicios deste. Desde o punto de vista do protocolo TCP non é tan sinxelo. Pódese considerar que ao recibir o cliente o último segmento do arquivo envíase un ACK ao servidor para confirmar, e é entón o servidor é cando desconecta e envía esa mensaxe de desconexión (non o pode facer antes de ter todos os ACK). Despois o cliente responde a esa mensaxe co seu ACK de desconexión e inmediatamente (polo que non hai retardo, o tempo é despreziable), porque non precisa confirmación, pode abrir unha conexión nova (son dúas mensaxe pero se fan case ao mesmo tempo).

A diferenza entre as dúas opcións é considerar o protocolo TCP que vai por debaixo explicitamente ou non. No primeiro caso o tempo de desconexión se incúe no tempo de transferencia, no segundo é explícito. Como en realidade no segundo caso non se teñen en conta cantos segmento hai e os seus ACK, senón que se mete todo como tempo de transferencia ten sentido meter aí tamén a desconexión. Na práctica todo isto está moi optimizado e se superpoñen moitas cousas, polo que non ten moitísima importancia.

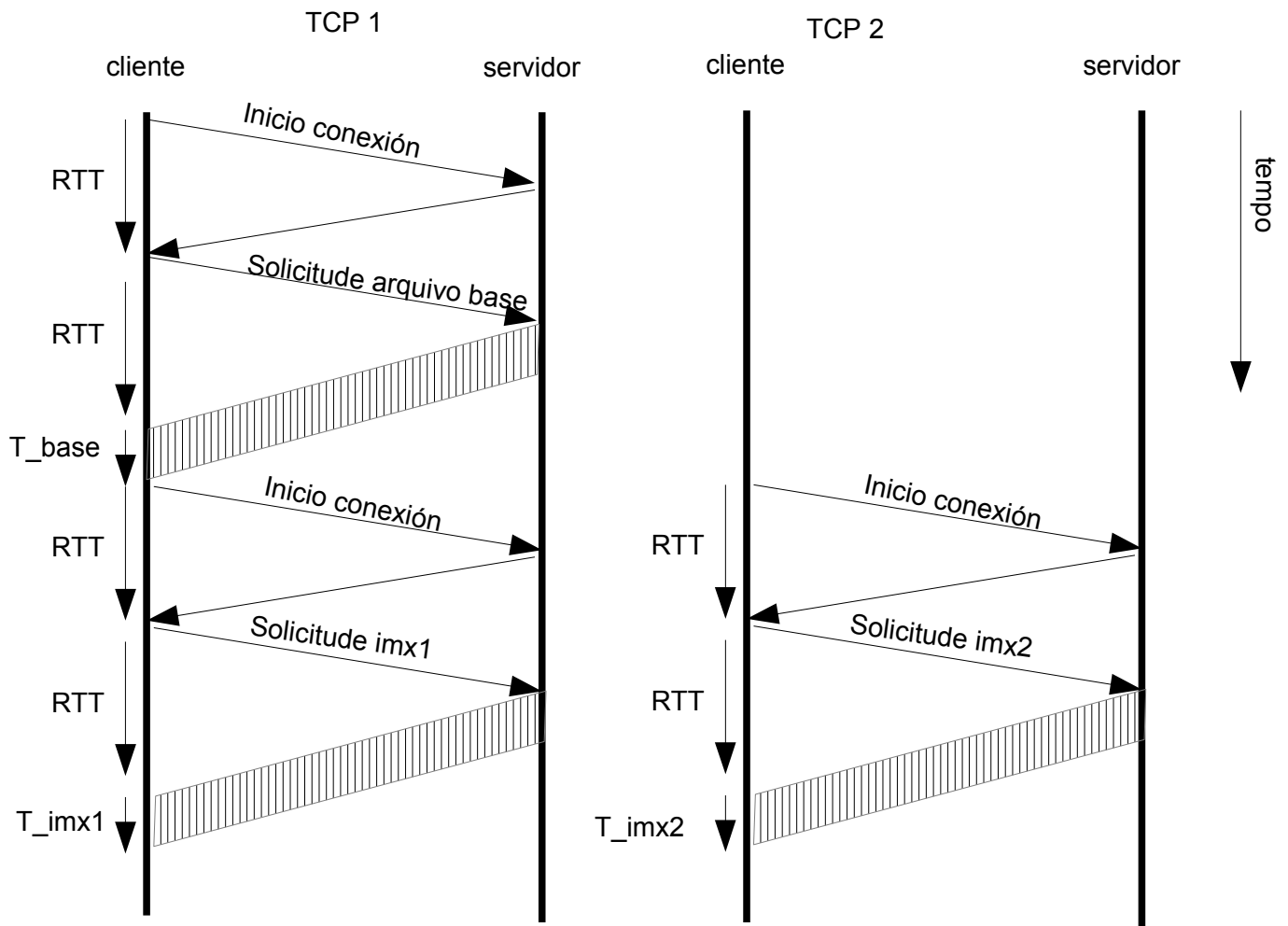
No exame e exercicios podedes usar o que queirades, sen dar explicacións.

Non Persistente - Serie



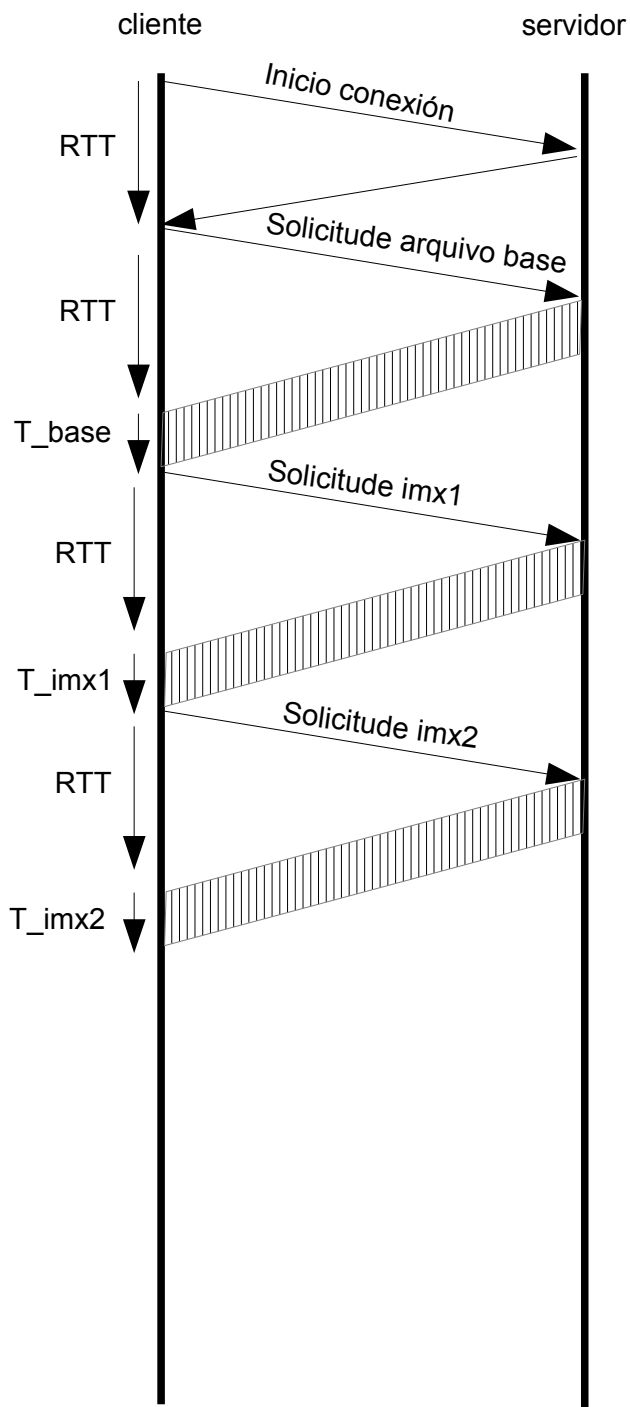
$$\text{Tempo total} = 6 \text{ RTT} + T_{\text{base}} + T_{\text{imx1}} + T_{\text{imx2}}$$

Non Persistente - Paralelo

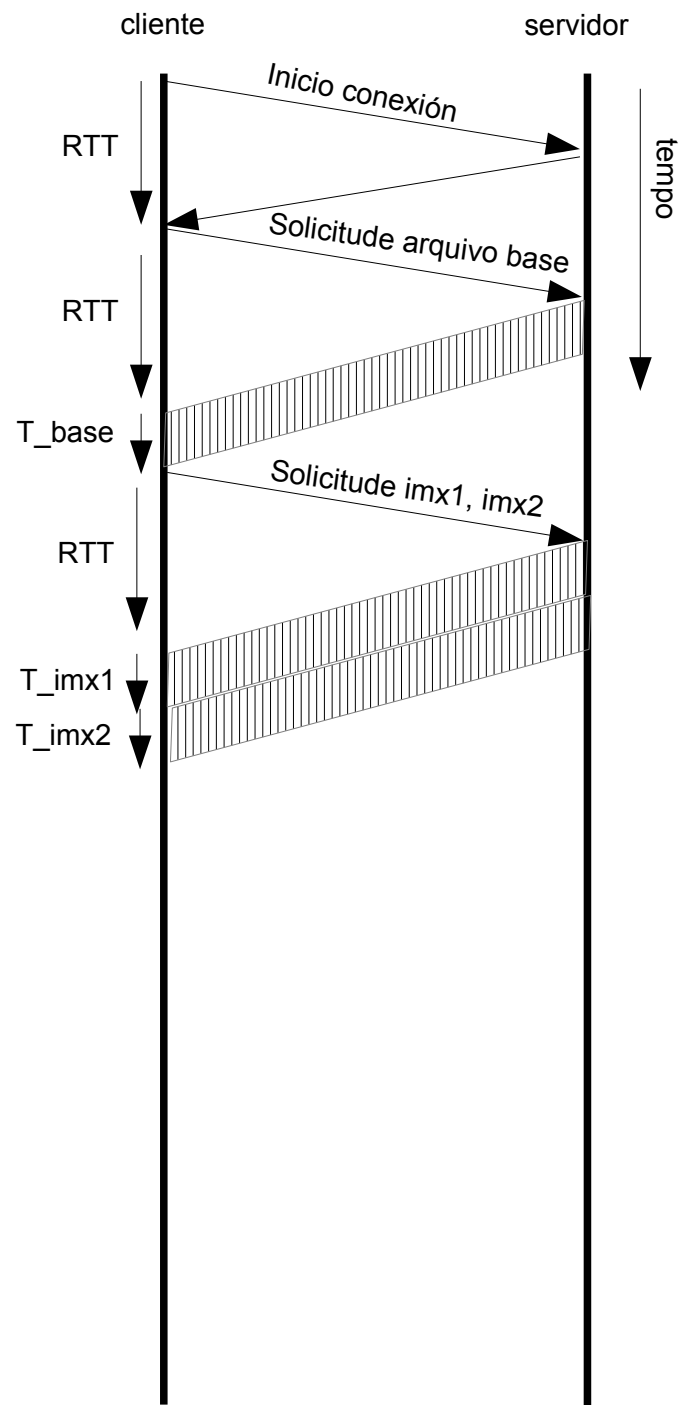


$$\text{Tempo total} = 4 \text{ RTT} + T_{\text{base}} + \max(T_{\text{imx1}} + T_{\text{imx2}})$$

Persistente – Sen entubamento



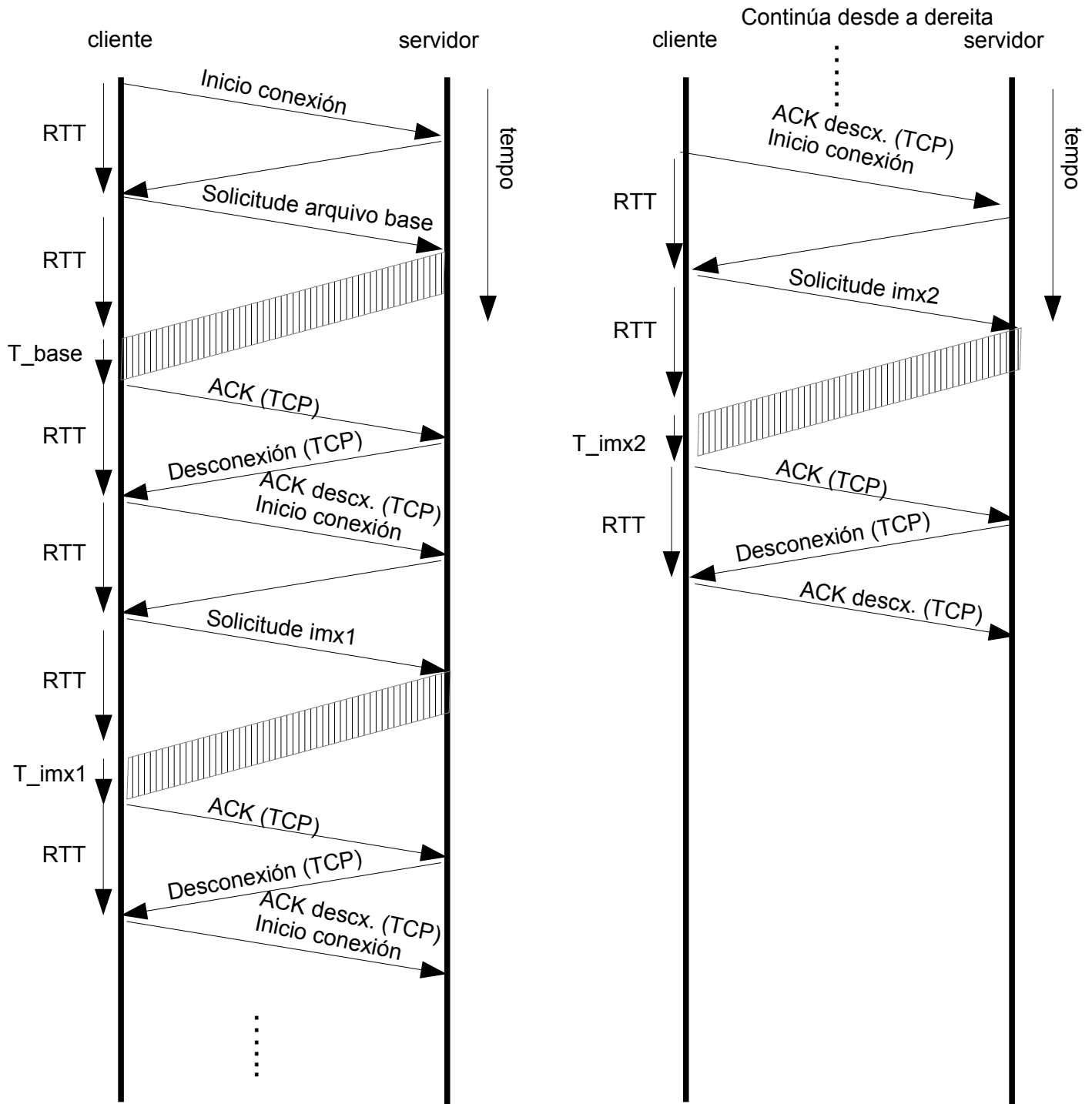
Persistente – Con entubamento



Tempo total (sen ent.) = 4 RTT + T_base + T_imx1 + T_imx2

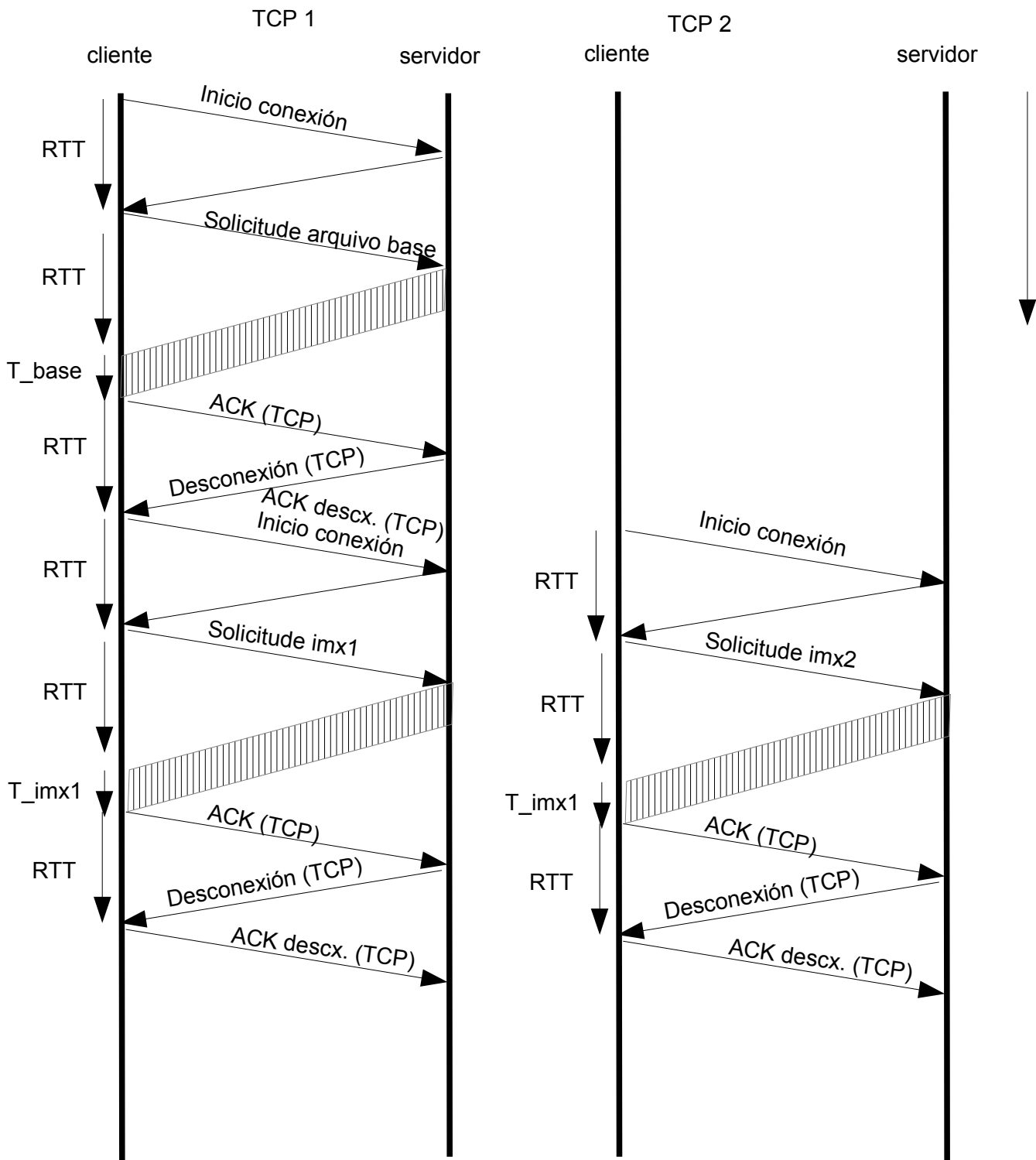
Tempo total (con ent.) = 3 RTT + T_base + T_imx1 + T_imx2

Non Persistente – Serie (tendo en conta desconexión TCP explícita)



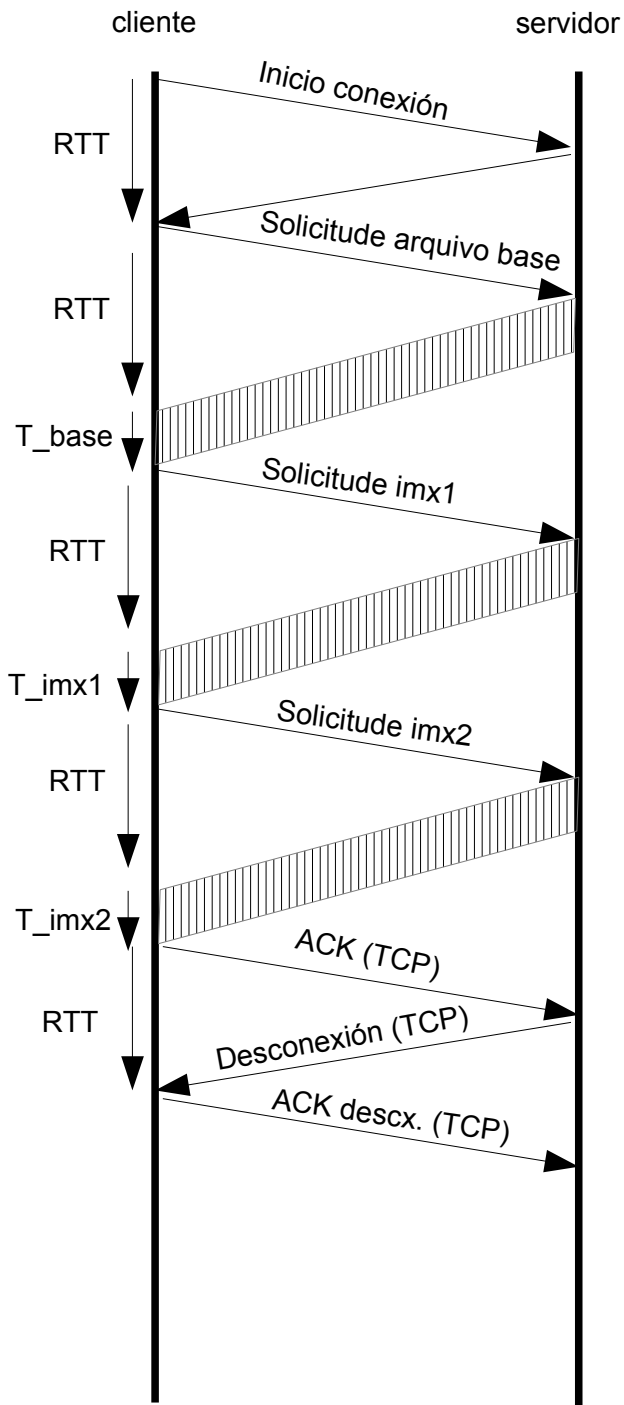
$$\text{Tempo total} = 9 \text{ RTT} + T_{\text{base}} + T_{\text{imx1}} + T_{\text{imx2}}$$

Non Persistente – Paralelo (tendo en conta desconexión TCP explícita)

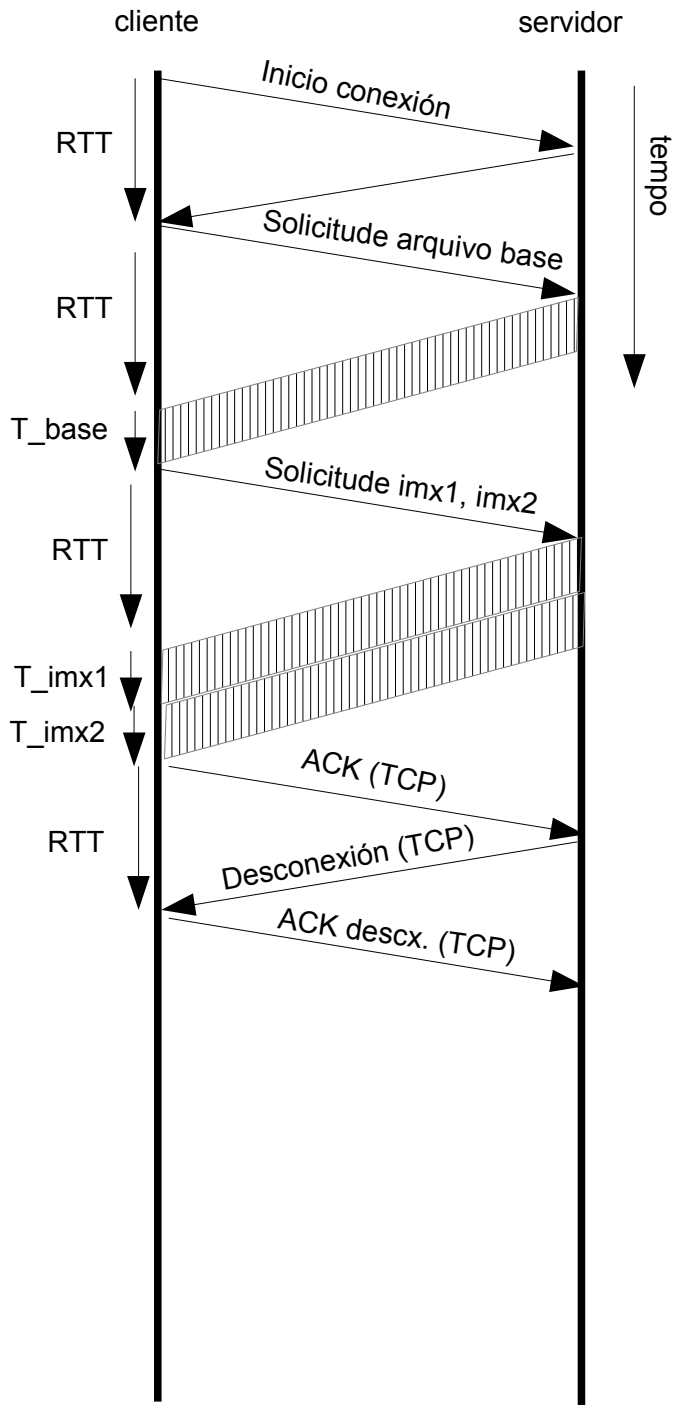


$$\text{Tempo total} = 6 \text{ RTT} + T_{base} + \max(T_{imx1} + T_{imx2})$$

Persistente – Sen entubamento
(tendo en conta desconexión TCP explícita)



Persistente – Con entubamiento
(tendo en conta desconexión TCP explícita)



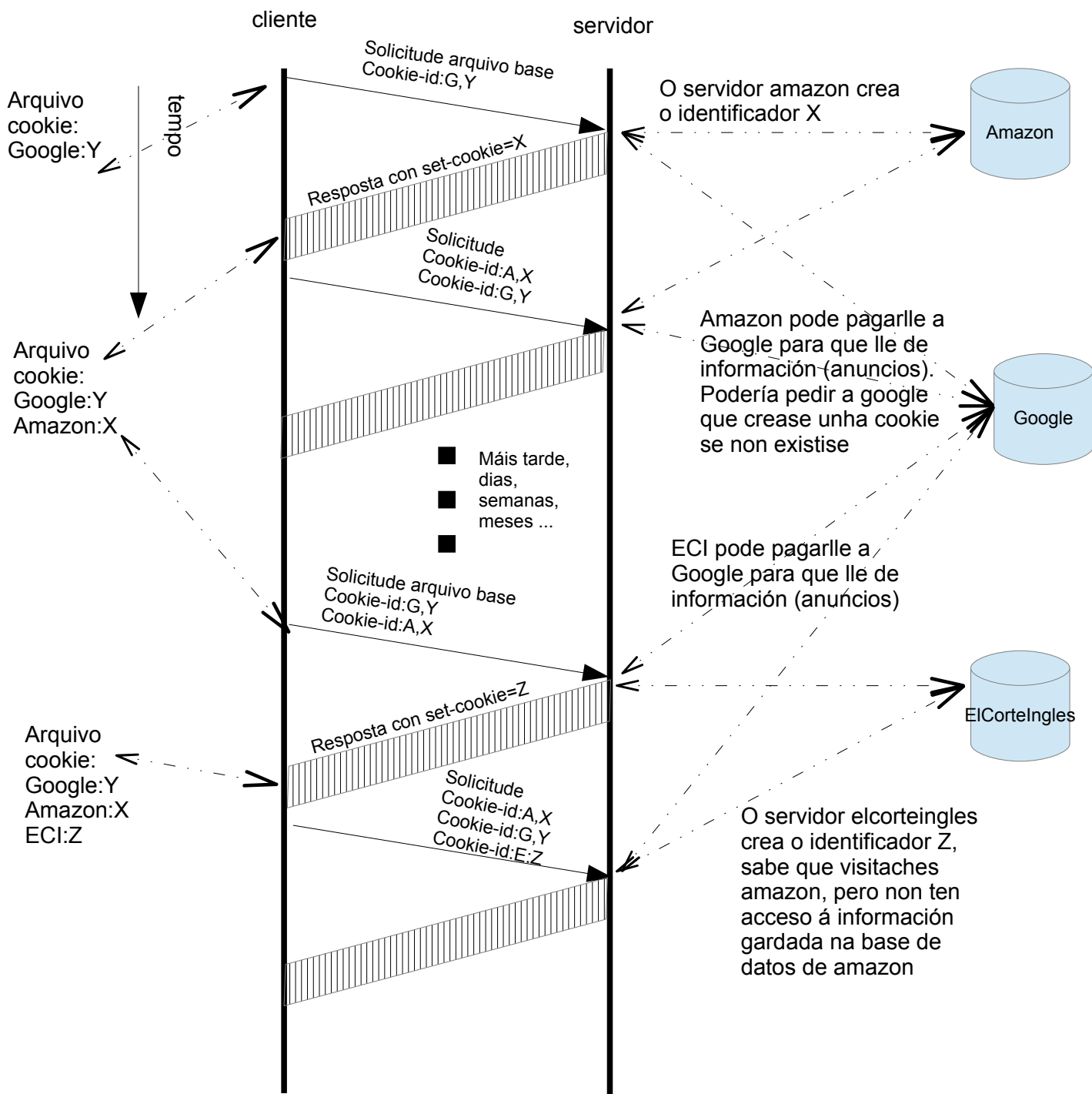
Tempo total (sen ent.) = $5 \text{ RTT} + T_{base} + T_{imx1} + T_{imx2}$

Tempo total (con ent.) = $4 \text{ RTT} + T_{base} + T_{imx1} + T_{imx2}$

4. Explica en que consiste o mecanismo das cookies no protocolo HTTP e para que se utiliza. Mostra mediante un exemplo o seu funcionamento utilizando un diagrama de liña de tempo.

O protocolo HTTP non ten estado, cada petición dun obxecto non ten en conta os anteriores. Se queremos comprar algo, por exemplo, ao entrar na páxina do carro non recorda o produto que eliximos na páxina anterior. Para dotar a HTTP de estado, e poder seguir o que o usuario fai nas páxinas, creáronse as cookies. Cando se entra nunha páxina o servidor pode escribir un número identificativo nun arquivo no computador do usuario, usando a opción *set-cookie* na resposta HTTP. Desta maneira o usuario garda nun arquivo o nome da páxina visitada e o número. Cando o usuario volva a facer unha petición ao servidor enviaralle o arquivo de cookies (en principio manda todas as cookies, pódese engadir unha opción á cookie para que so se envíe se se conecta a unha páxina determinada). O servidor web pode usar este número do arquivo de cookies para gardar nunha base de datos información sobre un usuario concreto, desta maneira sabe o que fai na páxina. Hoxe en día este mecanismo é algo máis complexo, existen 3 tipos principais de cookies ao menos: as de sesión, que se borran ao pechar o navegador, que serven para identificar usuarios, carros da compra, funcionalidades esenciais das páxinas; as de primeiros, que son as que usa cada páxina/servidor/empresa para recordar aos seus usuarios entre sesións; e as de terceiros, que son cookies de páxinas/servidores/empresas que non estamos a visitar directamente pero que venden os seus servizos a outras páxinas/servidores/empresas de modo que se poida rastrexar a actividade dun usuario entre páxinas de distintas empresas, para publicidade, estatísticas, etc.

Cookies (omitense os inicios de conexión)



5. Tes que diseñar un protocolo de capa de transporte que usa xanela deslizante de forma similar a como o fai TCP. Este protocolo vai ser usado sobre unha rede a 100 Mbps. O retardo da rede é de 100 ms, e o tempo de vida máximo dun segmento (MSL) é de 60 segundos. Se queremos aproveitar a capacidade da rede e que non se superpoñan os números de secuencia, cantos bits, como mínimo, deberías utilizar para os campos *Número de secuencia* (SN) e *Xanela otorgada* (AW) da cabeceira do teu protocolo?

Solución para o SN:

Para calcular o tamaño do campo SN, temos que ter en conta que os números de secuencia non deben dar a volta en menos do MSL (*Maximum Segment Lifetime*), que é 60 s. En 60 s, a 100 Mbps transmitense 6.000 Mbits, que son $750 \cdot 10^6$ bytes. Para representar ese número de bytes, necesitamos polo menos 30 bits, xa que

$$2^{29} = 536.870.912 < 750 \cdot 10^6 < 2^{30} = 1.073.741.824, \implies \text{SN de 30 bits.}$$

Solución para a AW:

Se queremos aproveitar ao máximo a capacidade da rede, a xanela otorgada debe se-lo suficientemente grande como para poder enviar información mentras non chegue o primeiro ACK. Desta forma, durante un RTT poderíamos enviar os bytes:

$$N\text{Bytes} = RTT \times BW = 2 \cdot 100 \text{ ms} \cdot 100 \text{ Mbps} = 20 \text{ Mbits} = 2.500.000 \text{ bytes.}$$

Para indicar unha fiestra dese tamaño necesitamos 22 bits, xa que

$$2^{21} = 2.097.152 < 2.500.000 < 2^{22} = 4.194.304, \implies \text{AW de 22 bits.}$$

Hai que verificar que o rango de números de secuencia sexa o dobre do tamaño da xanela, que neste caso se cumpre sobradamente.

Outra solución:

A xanela otorgada debe se-lo suficientemente grande como para poder encher a capacidade da rede. O produto retardo \times ancho de banda é:

$$R \times BW = 100 \text{ ms} \cdot 100 \text{ Mbps} = 10 \text{ Mbits} = 1.250.000 \text{ bytes.}$$

Para indicar unha xanela dese tamaño necesitamos 21 bits, xa que

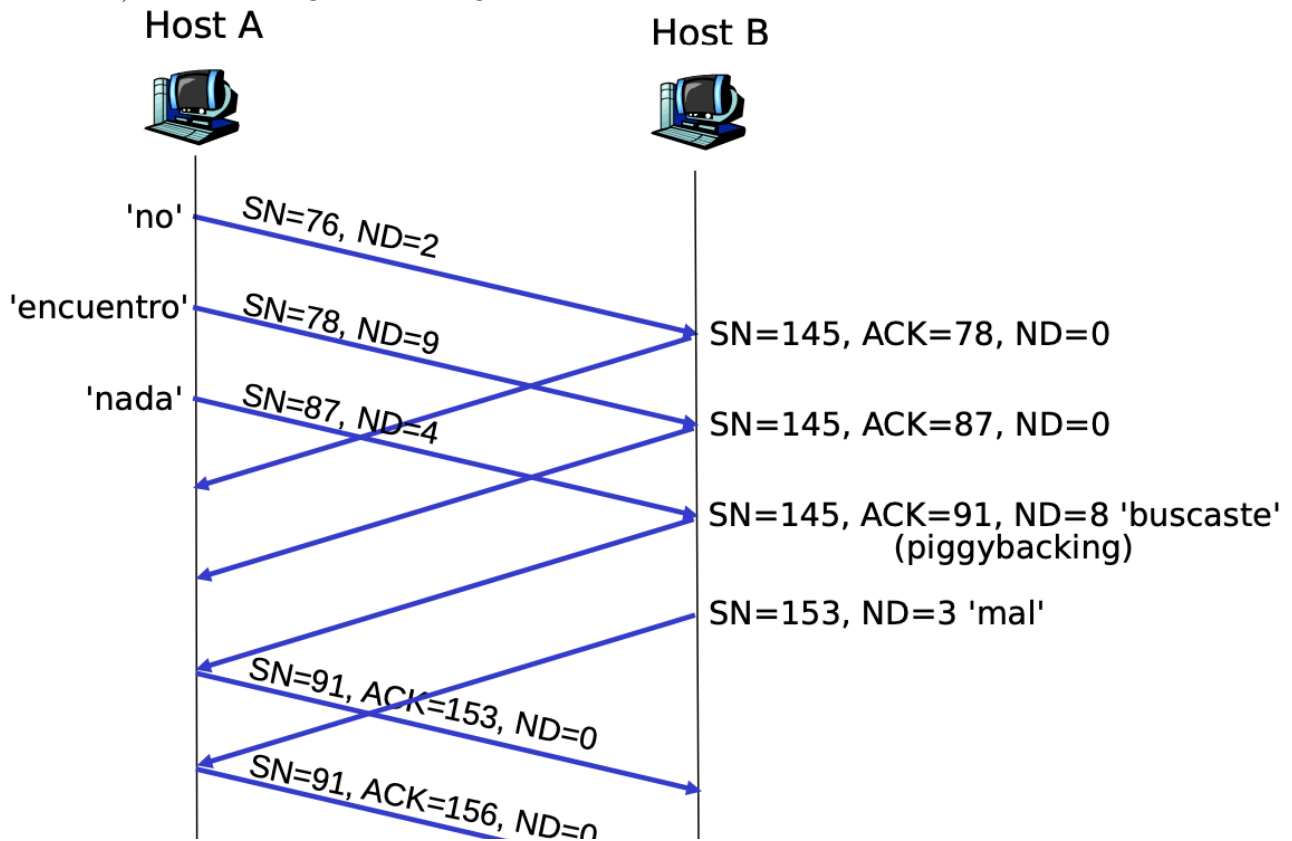
$$2^{20} = 1.048.576 < 1.250.000 < 2^{21} = 2.097.152, \implies \text{AW de 21 bits.}$$

Hai que verificar que o rango de números de secuencia sexa o dobre do tamaño da xanela, que neste caso se cumpre sobradamente.

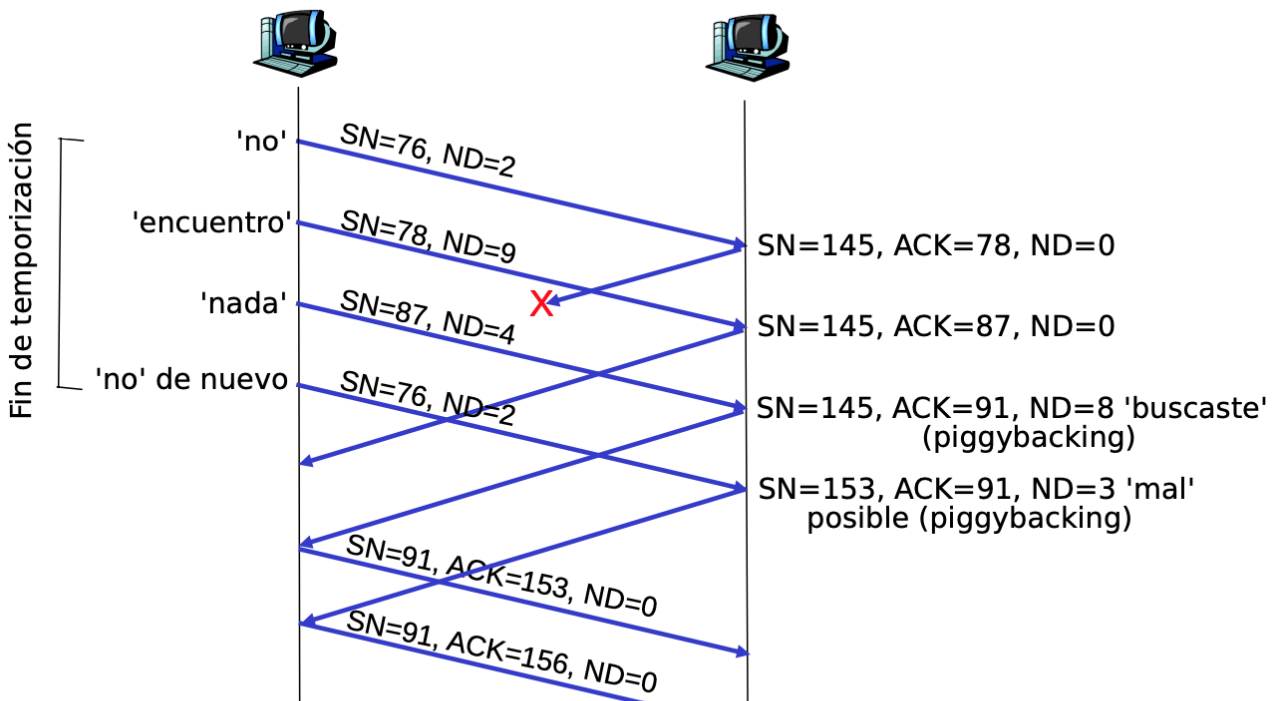
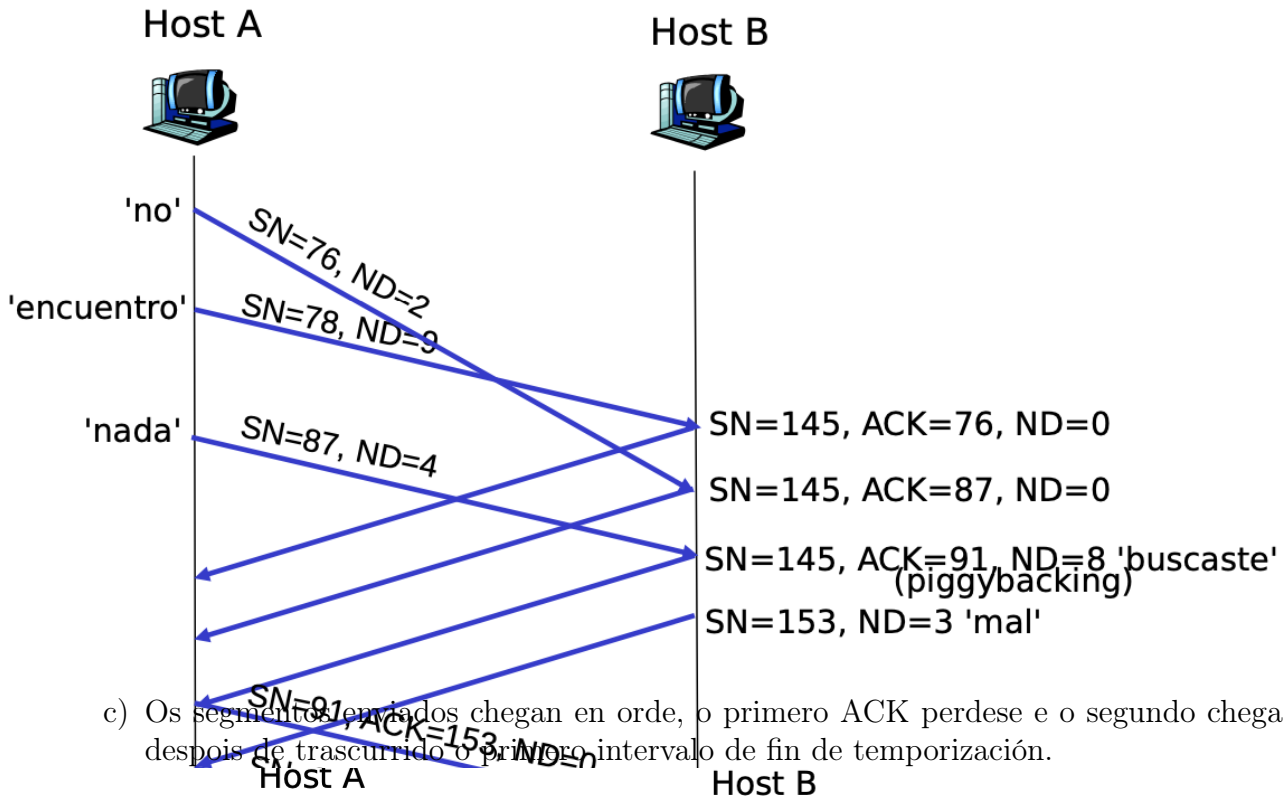
6. Supoñamos unha aplicación que consiste no envío e recepción de palabras entre dous hosts A e B usando TCP como protocolo de transporte e que a aplicación forza un PUSH con cada palabra. Un host envía un ACK cando recibe un segmento do outro. O host A utiliza o porto 5301 e o host B o 6666. Nun instante determinado o host A ten como número de secuencia o 76, o host B o 145 e a xanela otorgada é de 50 bytes. A continuación o host A envía as palabras 'no' 'encuentro' 'nada' e xusto despois o host B responde coas palabras 'buscaste' 'mal'. Debuxar un diagrama de tempo que mostre tódolos segmentos e ACKs enviados cos números de secuencia, números de ACK, portos orixe e destino e tamaño dos datos nos seguintes casos:

- a) Supoñendo que tódolos segmentos chegan en orde.

- b) Supoñendo que o segundo segmento que envía A chega antes que o primeiro.
- c) Supoñamos que os segmentos enviados chegan en orde, o primeiro ACK perdese e o segundo chega despois de traspasado o primeiro intervalo de fin de temporización.
- a) Todos os segmentos chegan en orde.



- b) O segundo segmento que envía A chega antes que o primeiro.



7. Consideremos unha única conexión TCP (Reno, con recuperación rápida) que emplea un enlace a 10 Mbps, que é o único entre os hosts emisor e receptor. Supoñamos que se quere enviar un arquivo de gran tamaño e que o buffer de recepción é moito maior que a xanela de congestión. O tamaño de segmento TCP é de 1.500 bytes, o RTT é de 100 milisegundos e nunca se perde un segmento por expiración do temporizador.

- a) Cal é o tamaño máximo de xanela (en segmentos) que esta conexión TCP pode alcanzar?

Sexa W o tamaño máximo de fiestra en segmentos. Entón,

$$\frac{W \times \text{MSS}}{\text{RTT}} = 10 \text{ Mbps}$$

xa que se descartarán paquetes se a tasa máxima de envío supera a capacidade do enlace. Entón, temos que

$$\frac{W \times 1500 \times 8}{0,1} = 10^7 \text{ bps} \implies W = 83,3 \Rightarrow 84 \text{ segmentos}$$

- b) Cales son o tamaño medio de xanela (en segmentos) e a taxa de transferencia media (en bps) desta conexión?

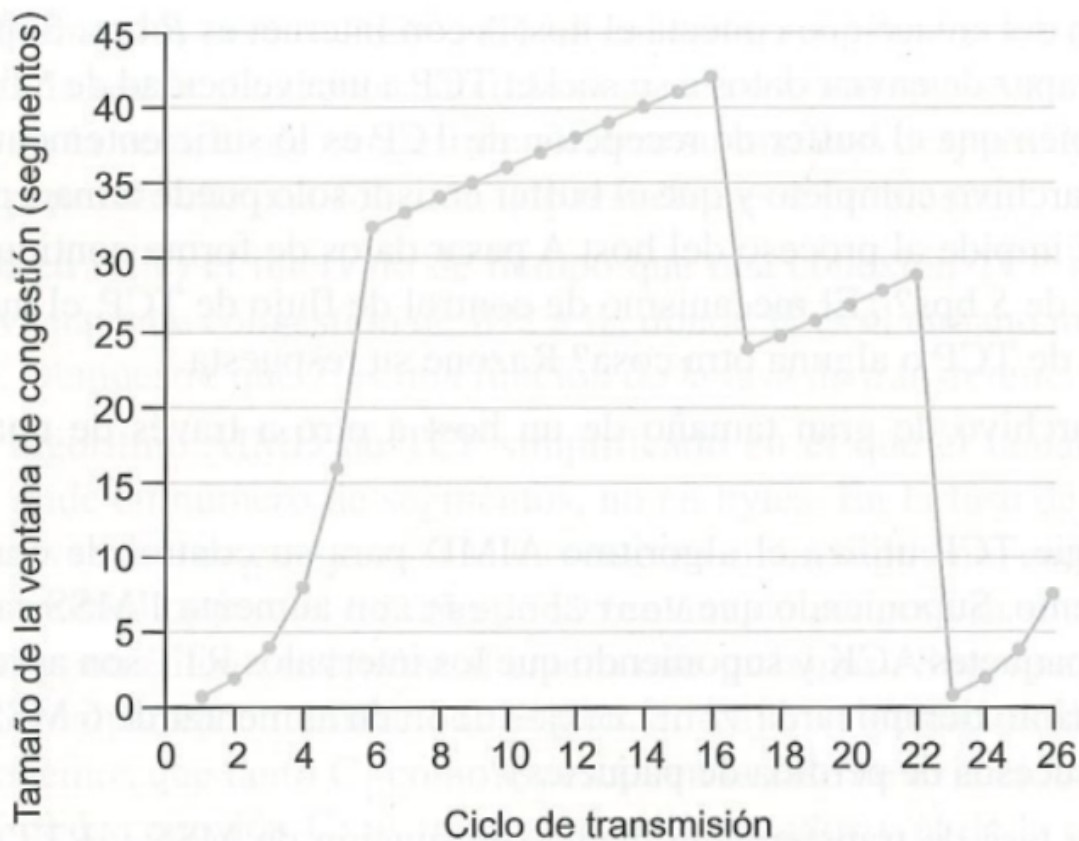
Como a fiestra de conexión W varía desde $W/2$ a W , o tamaño medio da xanela de conexión é $0,75W = 63$ segmentos. A taxa de transferencia media é

$$\frac{63 \times 1500 \times 8}{0,1} = 7,56 \text{ Mbps}$$

- c) Asumendo que a xanela de conexión está no seu valor máximo, canto tarda esta conexión TCP en alcanzar de novo o seu tamaño de xanela máximo despois de recuperarse dunha perda de paquete?

Cando se perde un paquete (3 ACKs duplicados) a ventana pasa a $W/2 + 3$. Para chegar ata W , a razón de 1 segmento por RTT, necesítanse $(W/2 - 3)$ RTTs. Polo tanto,

$$\left(W - \left(\frac{W}{2} + 3\right)\right) \cdot \text{RTT} = \left(\frac{W}{2} - 3\right) \cdot \text{RTT} = \left(\frac{84}{2} - 3\right) \times 0,1 = 3,9 \text{ segundos}$$



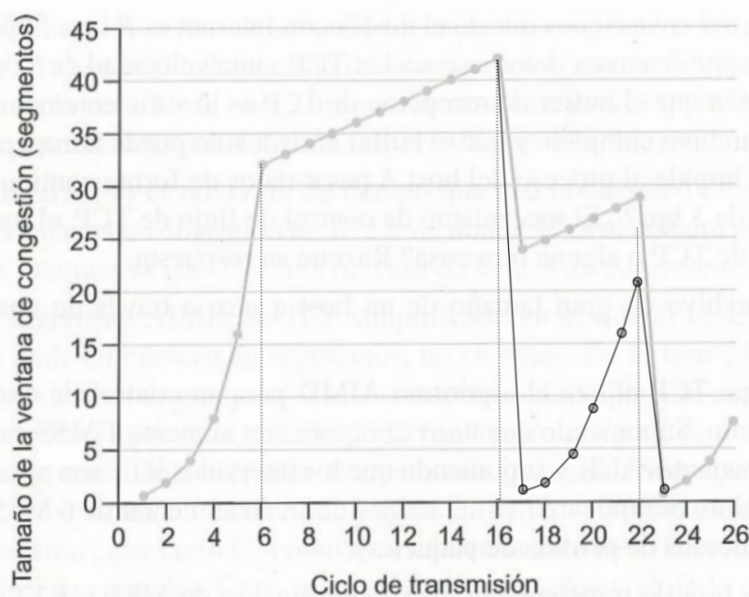
8. Na figura mostrase o comportamento de TCP Reno. Responder ás seguintes preguntas, razoando brevemente as respostas.

- Identifica os intervalos de tempo nos que TCP opera en inicio lento. Inicio lento nos intervalos [1, 6] e [23, 26].
- Identifica os intervalos de tempo nos que TCP opera en AIMD (evitación da congestión). AIMD nos intervalos [6, 16] e [17, 22], aínda que en realidade é [6, 22].
- Ao final do ciclo 16, detectase perda de segmento mediante 3 ACKs duplicados ou mediante un fin de temporización? 3ACKs duplicados. Se fose un fin de temporización a CW sería 1 e seguiría con inicio lento.
- Ao final do ciclo 22, detectase perda de segmento mediante 3 ACKs duplicados ou mediante un fin de temporización? Un fin de temporización e por iso a CW pasa a 1 e segue con inicio lento.
- Cal é o valor inicial de **umbral**? É 32, xa que termina a fase de inicio lento e pasase a AIMD.
- Cal é o valor de **umbral** transcurridos 18 ciclos? O **umbral** fíxase á metade da xanela de congestión cando se detecta a perda do paquete. No ciclo 16 a CW é 42, polo tanto, no ciclo 18 **umbral** é 21 (no 17 pasa a ser 21). A CW é 24 (21 + 3).
- Cal é o valor de **umbral** transcurridos 24 ciclos? O **umbral** fíxase á metade da xanela de congestión cando se detecta a perda do paquete. No ciclo 22 a CW é 29, polo tanto, no ciclo 24 **umbral** é 14 (no 23 pasa a ser 14).
- Durante que ciclo envíase o segmento 100? Durante o primeiro ciclo envíase o segmento 1; no segundo os segmentos 2 e 3; no terceiro os segmentos 4-7; no cuarto os segmentos 8-15; no quinto os segmentos 16-31; no sexto os segmentos 32-63; no sétimo envíanse os segmentos 64-96 (so 33, xa que está en AIMD); no octavo envíanse

34 segmentos, do 97 en adiante. Polo tanto, o segmento 100 envíase durante o ciclo 8.

- i) Supoñendo que se detecta unha perda de paquete despois do ciclo 26 a causa da recepción de 3 ACKs duplicados, cales serán os valores da xanela de congestión e **umbral**? O **umbral** fíxase á metade da xanela de congestión cando se detecta a perda do paquete. A xanela de congestión fíxase a ese novo **umbral** + 3 MSS. No ciclo 26 a CW é 8, polo tanto, no ciclo 26 o **umbral** pasa a ser 4 e a CW 7 e segue con AIMD.
- j) Supón que se utiliza TCP Tahoe (non ten recuperación rápida) e que se recibiron 3 ACKs duplicados no ciclo 16. Cales serán os valores da xanela de congestión e **umbral** no ciclo 19? Como non tñe recuperación rápida, no ciclo 17 a CW pasa a valer 1 e o **umbral** pasa á metade da CW cando se detecta a perda do paquete, que é 21. A partir de aí pasase á fase de inicio lento, polo tanto, no ciclo 19 CW vale 4 e o **umbral** 21.
- k) Supón que se utiliza TCP Tahoe e que se produce un fin de temporización no ciclo 22. Cantos segmentos foron enviados desde o ciclo 17 ao 22, ambo-los dous incluídos? Como non ten recuperación rápida, no ciclo 17 a CW pasa a valer 1, polo tanto envíase 1 segmento no ciclo 17; 2 segmentos no 18; 4 segmentos no 19; 8 segmentos no 20; 16 segmentos no 21 e 21 segmentos no ciclo 22 (**umbral** vale 21 e despois pasárase a AIMD se non houberse perdas). En total 52 segmentos.

A figura mostra como sería con TCP Tahoe.



9. Cabeceiras IP:

- a) Describir cada un dos campos da cabeceira IPv4 e explicar para que serven.
- b) Describir cada un dos campos da cabeceira IPv6 e explicar para que serven.
- c) Comenta as principais diferenzas entre o datagrama IPv4 e IPv6.

Nesta pregunta hai que ter claro que campos ten as cabeiras. Non é necesario saber a orde ou o número de bits exactos que ocupan cada un, pero si é importante saber cales son, porque é o mesmo que saber como funciona o protocolo IP. Saber que existe o TTL, porque débese saber que os paquetes teñen un número de saltos limitado. Saber que hai unha suma de comprobación para evitar erros, e que hai que recalcularla. Saber que

hai flags de fragmentación porque existe fragmentación en IPv4. Saber que os paquetes se identifican e poden ter información sobre o tipo de servizo, etc. Saber que en IPv6 desaparece a segmentación e a comprobación de erros, engadense cabeceiras opcionais. En detalle as cabeiras as podedes atopar en: https://es.wikipedia.org/wiki/Cabecera_IP
https://es.wikipedia.org/wiki/IPv6#Paquete_IPv6

10. CIDR e agregación de rutas:

- a) Dispone do bloque de direccións 194.24.0.0/18, que clase de rede é? Supoñamos que a Universidade de A Coruña solicita 2048 direccións (equivalente a 8 redes de clase C). A continuación, a Universidade de Santiago de Compostela solicita 4096 direccións (equivalente a 16 redes de clase C). Finalmente, a Universidade de Vigo solicita 1024 direccións (equivalente a 4 redes de clase C). Realizar a asignación de direccións a cada unha das universidades.
- 1) Indica, para cada unha destas tres subredes, a máscara de subrede (en formato máscara e sufixo) así como a dirección base e de broadcast.
 - 2) Indica cales son as IPs que quedan libres. Podemos asignar estas IPs a unha única subrede?, por que? En caso afirmativo, indica a máscara, dirección base e dirección de broadcast desta subrede.
 - 3) Mostra as entradas coas que habería que actualizar un router que teña que reenviar paquetes a estas redes.

A UDC solicita 2048 direccións, 2^{11} . Polo tanto, asignaselle 194.24.0.0/21, onde $21 = 32 - 11$. A dirección base e a de broadcast son:

Base: 194.24.00000000.00000000 = 194.24.0.0

Broadcast: 194.24.00000111.11111111 = 194.24.7.255

- Máscara de subrede: /21, 255.255.248.0
- Dirección base: 194.24.0.0
- Dirección de broadcast: 194.24.7.255

A USC solicita 4096 direcciones, 2^{12} . Non se lle pode asignar 194.24.8.0/20, onde $20 = 32 - 12$, xa que implicaría que a dirección base e a de broadcast serían:

Base: 194.24.00000000.00000000 = 194.24.0.0

Broadcast: 194.24.00001111.11111111 = 194.24.15.255

e as primeiras direccións están asignadas á UDC. Polo tanto, asignaselle 194.24.16.0/20. A dirección base e a de broadcast son:

Base: 194.24.00010000.00000000 = 194.24.16.0

Broadcast: 194.24.00011111.11111111 = 194.24.31.255

- Máscara de subrede: /20, 255.255.240.0
- Dirección base: 194.24.16.0
- Dirección de broadcast: 194.24.31.255

A UVigo solicita 1024 direccións, 2^{10} . Hai varias opcións, pero se lle asignan as direcciones a continuación da UDC: 194.24.8.0/22, onde $22 = 32 - 10$. A dirección base e a de broadcast son:

Base: 194.24.00001000.00000000 = 194.24.8.0

Broadcast: 194.24.00001011.11111111 = 194.24.11.255

- Máscara de subrede: /22, 255.255.252.0
- Dirección base: 194.24.8.0
- Dirección de broadcast: 194.24.11.255

En resumo:

Univ	Dir. base	Dir. broadcast	Red
UDC	194.24.0.0	194.24.7.255	194.24.0.0/21
UVigo	194.24.8.0	194.24.11.255	194.24.8.0/22
Dispoñible	194.24.12.0	194.24.15.255	194.24.12.0/22
USC	194.24.16.0	194.24.31.255	194.24.16.0/20
Dispoñible	194.24.32.0	194.24.63.255	194.24.32.0/19

As direccións libres non se poden asignar á mesma rede porque non se poden asignar unha dirección base e unha máscara únicas. Ademais, o número de direccións libres non é unha potencia enteira de 2.

As táboas dos routers actualizaríanse coas tres entradas correspondentes:

Táboa de reenvío dun router			
destino	interface	gateway	métrica
...
194.24.0.0/21	int _i	*	i saltos
194.24.8.0/22	int _j	*	j saltos
194.24.16.0/20	int _k	*	k saltos
...
0.0.0.0/0 (por defecto)	int _x	*	x saltos

- b) Que ocorre nese router cando lle chega un paquete dirixido a 194.24.17.4?

O que se fai é buscar a coincidencia de prefixo, é dicir, realízase un AND bit a bit con cada unha das máscaras das entradas ata que se produza a coincidencia de prefixo. No exemplo, faise o AND bit a bit de 194.24.17.4 coa máscara da UDC, 255.255.248.0 ou /21:

```
194. 24.00010001.00000100 = 194.24.17.4
255.255.11111000.00000000 = 255.255.248.0
194. 24.00010000.00000000 = 194.24.16.0
```

Resultando 194.24.16.0. Como non coincide coa dirección base da entrada correspondente á UDC, entón non o envía pola interface int_i. Probase coa da UVigo, é dicir, o AND bit a bit de 194.24.17.4 coa máscara da UVigo, 255.255.252.0 ou /22:

```
194. 24.00010001.00000100 = 194.24.17.4
255.255.11111100.00000000 = 255.255.252.0
194. 24.00010000.00000000 = 194.24.16.0
```

Resultando 194.24.16.0. Como tampouco coincide coa dirección base da entrada correspondente á UVigo, entón non o envía pola interface int_j. Probase coa da USC, é dicir, o AND de 194.24.17.4 coa máscara da USC, 255.255.240.0 ou /20:

```
194. 24.00010001.00000100 = 194.24.17.4
255.255.11110000.00000000 = 255.255.240.0
194. 24.00010000.00000000 = 194.24.16.0
```

Resultando 194.24.16.0, que sí coincide coa dirección base correspondente á entrada da USC, polo tanto, se non se atopan máis coincidencias de prefixo, o envía por la interface int_k. No caso de que aparezan máis coincidencias, o paquete envíase pola interface que produza a **coincidencia co prefixo máis longo**. Se non se produce ningunha coincidencia, envíase pola interface que se indica na entrada por defecto.

- c) Agregación de rutas. Explicar en que consiste mediante un exemplo.

Supoñamos un router en Madrid, que so teña 4 interfaces de saída, unha para a zona Noroeste, outra para o Nordeste, outra para o Sur e outra para Canarias. As tres entradas poderían combinarse nunha entrada agregada, que sería 194.24.0.0/19 á interface Noroeste. Esta entrada reenvía tódolos paquetes dirixidos a calquera das tres universidades (e tamén ás direccións 194.24.12.0/22 sen asignar) pola interface do Noroeste. Se esas direccións (194.24.12.0/22) se asignasen a unha institución que non estivese na zona Noroeste, necesitaríase unha entrada adicional para esas direccións. Nesta situación, un paquete dirixido, por exemplo, a 194.24.13.147 reenviaríase pola interface da entrada que produza a coincidencia co prefixo máis longo, que, entre 194.24.0.0/19 e 194.24.12.0/22, resultaría esta última.

194. 24.00001101.10010011 = 194.24.13.147
255.255.11100000.00000000 = 255.255.224.0
194. 24.00000000.00000000 = 194.24.0.0
194. 24.00001101.10010011 = 194.24.13.147
255.255.11111100.00000000 = 255.255.252.0
194. 24.00001100.00000000 = 194.24.12.0