

1.

a.

- i. Three-way handshake是TCP用來建立client跟server間建立可靠連接的一個過程。三次握手確保雙方都準備好發送與接收數據，以避免不同步或其他意外的發生。

第一步 - Client發起連接請求：

Client向Server發送連接請求，內容包含用於建立連接的SYN number，以及Initial Sequence Number。

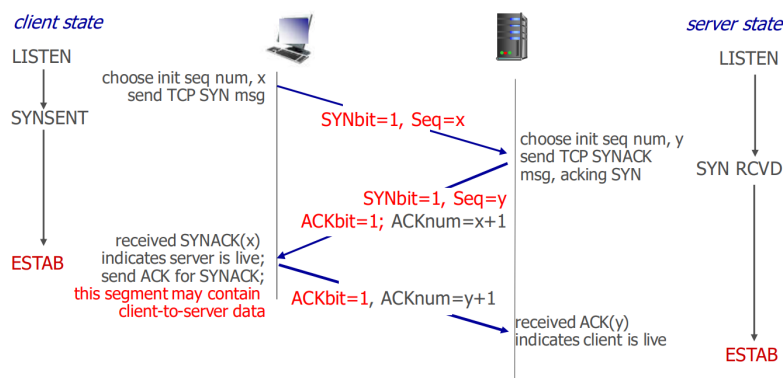
第二步 - Server確認連接請求：

Server確認收到Client的SYN後，向Client發送SYN及ACK以確認收到連接請求。

第三步 - Client確認連接：

Client在收到Server的SYN和ACK後，Client同樣會發送ACK給Server，用於確認連接建立。

Server在接收到Ack後，連接即建立成功。



b.

- i. (5%) persistent connections 是使用同一個TCP連接來傳送和接收多個HTTP請求/應答的方法，使用該方法可以避免為每一個新的請求/應答打開一個新的連接。
- ii. (5%) HTTP1.0為non-persistent，每發送一次請求就需要多開一組TCP connection.
- c. (5%) 允許一次傳送好幾個封包，無須等待前一個封包的回應，即可傳送下一個封包

(5%) Out Of Order Responses: 如果Server不是按照請求的順序回覆，Client可能需要重新排序回覆以確保是正確的data。

Head-of-Line Blocking: 如果其中一個請求延遲，可能會導致後面的請求被block，直到該請求得到回覆。

相容性問題: 不是所有Server都支援pipeline。

d.

i. (3%)Go-Back-N:

使用一組timer, 當timeout之後, 將未返回ACK之後的packet都進行重送

ii. (3%)Selective-Repeat:

只須對timeout的該packet進行重送即可

iii. (4%)Selective-Repeat比較有效率 + 舉例

2. 令bandwidth = R(bits/s)

500 bytes = 500 \* 8 bits

transmission delay =  $500 * 8 / R$  (s)

propagation delay =  $250km / 2 * 10^8(m/s) = 25000m / 2 * 10^8(m/s)$

$25000m / 2 * 10^8 > 1/4 * 500 * 8 / R$

$\Rightarrow R > 8 * 10^5(bits/s)$  or  $1*10^5 bytes/s$

only propagation delay 5pt

3. (3%) receiver window用來幫助TCP的flow control

Sender根據receiver window的大小來控制可以接收的數據量, 同時根據流量狀況動態調整window大小。

(3%)congestion window用來幫助TCP的congestion control

congestion window用於控制Sender傳輸資料的速率, Sender會根據congestion window的大小來動態調整傳輸的數據量, 以避免網路擁塞。

(4%)舉例:.....

4. (3%)SMTP 是一種用於將電子郵件從client端發送到server或在servers之間發送的protocol。

(3%)IMAP主要用於訪問和管理server上的電子郵件, 它允許client端從server檢索電子郵件。

**(4%)How do these protocols work together?**

- Use SMTP to send the email from email client to email provider's SMTP server.
- The SMTP server forwards the email to the recipient's email provider's SMTP server.
- The recipient's email provider stores the email using IMAP.
- The recipient uses IMAP to access and read the email.

5. socket.recv(): 不提供有關資料來源(the sender's address or port)的資訊

socket.recvfrom(): 提供有關資料來源(including the sender's address and port)的資訊

**When to use one over the other:**

- socket.recv():  
使用TCP socket時(連接導向)  
不需要有關寄件者address和port的資訊時
- socket.recvfrom():  
使用UDP socket時(非連接)  
需要有關寄件者address和port的資訊時

6. 0x9DA8

0x4A7F=0100 1010 0111 1111

0x65D1=0110 0101 1101 0001

0x8392=1000 0011 1001 0010

0x2E74=0010 1110 0111 0100

一個hex to bin 0.5

全部相加, 有進位則加到和的個位, 再取一補數: 1001 1101 1010 1000 => 0x9DA8

進位沒補至最低位 -2

沒有算補數 -5

7. A. start\_timer  
B. start\_timer  
C. stop\_timer  
D. start\_timer  
E. stop\_timer  
F. timeout  
G. timeout  
H. udt\_send(sndpkt)  
I. start\_timer

一個一分