

【資訊處理】

《資料通訊》

試題評析

今年的高考資訊處理人員的資料通訊，試題十分平易，沒有特別艱澀的題目，考生要取得高分並不難。較關鍵的是第二題與第八題，其中第二題必須了解 16-QAM 的 baud rate 與 bit rate 有 1:4 的關係，就能簡單算出答案；第八題的第 3 小題，則是考 slow start 的觀念，尚未達到最大窗口一半前，會有加倍增大窗口的作用，之後則以線性式增加增大窗口。預估今年考生，要拿到六、七十分應不致於太難，程度較好的考生應該可以拿到八十分以上。

一、類比信號要經過什麼程序才能轉換成數位信號？相較於直接傳輸類比訊號，將類比訊號數位化後再進行數位傳輸有那些好處？（10分）

答：

(一)類比訊號要轉換成爲數位訊號，需要經過取樣(sampling)的程序，也就是以固定的週期進行取樣，每次取樣就可以產生一組數位資料。例如：PCM(pulse code modulation)編碼，就是每 $125\mu\text{sec}$ 取樣一次，每秒取樣 8000 次，每次取樣可得到 7bits 或 8bits 的資料，位元傳輸率爲 64Kbps。

(二)數位傳輸的優點：

- 1.訊號只允許兩種兩壓值，可以在適當的距離加裝中繼器，即可將訊號還原成與原先相同，所以可連接任意個中繼器；而類比訊號使用訊號放大器卻會有變形失真的現象，而且會累積而愈益嚴重。
- 2.可以將聲音、影像、數據等資料整合在一起，使線路有較佳的利用率。
- 3.數位訊號的中繼器成本較低。
- 4.數位訊號的維護較容易，因爲數位訊號比較容易偵測到其問題的出處。

二、假設 BPSK (binary phase shift keying) 的頻寬使用效率是 1 bit/s/Hz，所以使用 BPSK 傳輸 1 kbps 的資料需要 1 kHz 的頻寬。

(一)使用 16-QAM (quadrature amplitude modulation) 傳輸 1 kbps 的資料需要多少 Hz 的頻寬？（4分）

(二)在什麼情形下我們會考慮採用 BPSK 而不用 16-QAM？在什麼情形下我們會考慮採用 16-QAM 而不用 BPSK？（6分）

答：

(一)使用 16-QAM 時，是以 16 種不同的振幅與相位組合，來傳送 4bits 的資料，也就是說每個訊號可以傳送 4bits 的資料，因此所需的最低頻寬 = $1\text{kbps} \times \frac{1\text{baud}}{4\text{bits}} = 250\text{Hz}$ 即可。

(二)在干擾較嚴重的情形，可以採用 BPSK，因爲 BPSK 是以相位來判定傳輸的資料內容，較不會受干擾影響；而在頻寬較低且沒有干擾的通道上，傳輸大量資料，需要較高位傳輸率時，可以採用 16-QAM。

三、請說明如何利用直接序列展頻 (direct sequence spread spectrum) 技術達到多重存取的目的。（10分）

答：

DSSS 就是每個傳輸者使用一組 PN sequence (Pseudorandom number sequence)，將其資料位元乘上 PN sequence 以產生傳送的 chips，然後將其在共用通道上傳送，接收者必須事先知道傳送者使用的這組 PN sequence，以便能夠在通道上辨認出訊息。不同的資料傳輸可以使用不同的 PN sequence，同時在一個通道上傳送，但各自的接收者可以辨認出各自的資料，以此方式來達到通道的多重存取之目的。

四、考慮使用一個CRC碼做資料訊框 (frame) 傳輸的錯誤偵測。假設CRC碼的生成多項式 (generator polynomial) $g(X) = X^4 + X + 1$ ，原始訊息的長度為12位元，傳輸的訊框長度為16位元 (CRC碼放在後面)。

(一)如果原始訊息為100110011100，則編碼後之訊框為何？(5分)

(二)試舉出2個無法偵測出來且權重 (weight) 不同的錯誤型態。(5分)

答：

(一)1001100111000010 加底線部份為 CRC 碼。

(二)1011111111000010 加底線部份的三個bits錯誤時，無法察覺有錯誤。

1111001111000010 加底線部份的四個bits錯誤時，無法察覺有錯誤。

五、在媒體接取控制 (MAC) 機制設計中：

(一)請敘述隨機接取 (random access) 控制機制與固定指定 (fixed assignment) 控制機制之定義與其差異。(5分)

(二)請說明當網路負載很重 (heavily loaded) 時，應該使用以上那種控制機制比較有效率？為什麼？反之，當網路負載很輕 (lightly loaded) 時，應該使用以上那種控制機制比較有效率？為什麼？(5分)

答：

(一)所謂隨機接取是指沒有主控機制來分配時間，由各個存取裝置自行決定要不要發送；而固定指定則是以循環方式，將通道頻寬分配給通道上的各個裝置，如TDM。

(二)當負載很重時，採用固定指定的方式會較有效率，因為這樣可以完全充分地運用通道的頻寬；而當負載很輕時，採用隨機接取會較佳，因為碰撞機率低，可以讓要發送的裝置可以較在較早的時間發送資料，延遲時間較短。

六、請回答以下問題：

(一)某公司取得一個B類IP網路151.112.0.0。今欲將其切割成13個子網路，應如何設定該網路之子網路遮罩 (subnet mask)，才能符合此要求？請列舉其中第三個子網路IP地址之範圍及此子網路廣播地址。(5分)

(二)假設你要替一個WWW伺服器選擇有效的自動錯誤控制機制 (ARQ)，但又不希望增加該伺服器太多處理負擔，請問應該用那一種ARQ機制較為適合？請說明理由。(5分)

答：

(一)子網路遮罩為 255.255.240.0

第三個子網路 IP 地址範圍為 151.112.48.1~151.11.63.254

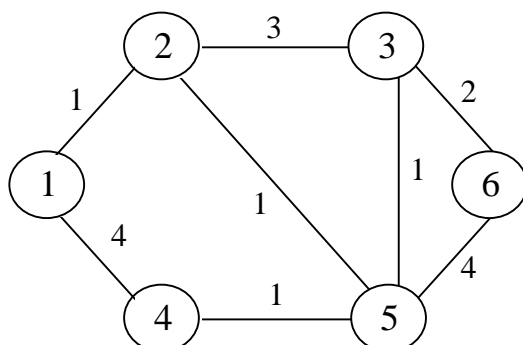
子網路廣播地址為 151.112.63.255

(二)使用 selective repeat 的 ARQ 做法，因為只要重傳有錯的封包，可以減輕伺服器的負擔。

七、在封包繞境 (routing path) 演算機制中：

(一)請敘述最短路徑 (shortest-path routing) 演算法與最低成本 (least-cost routing) 演算法之關係與其差異。(8分)

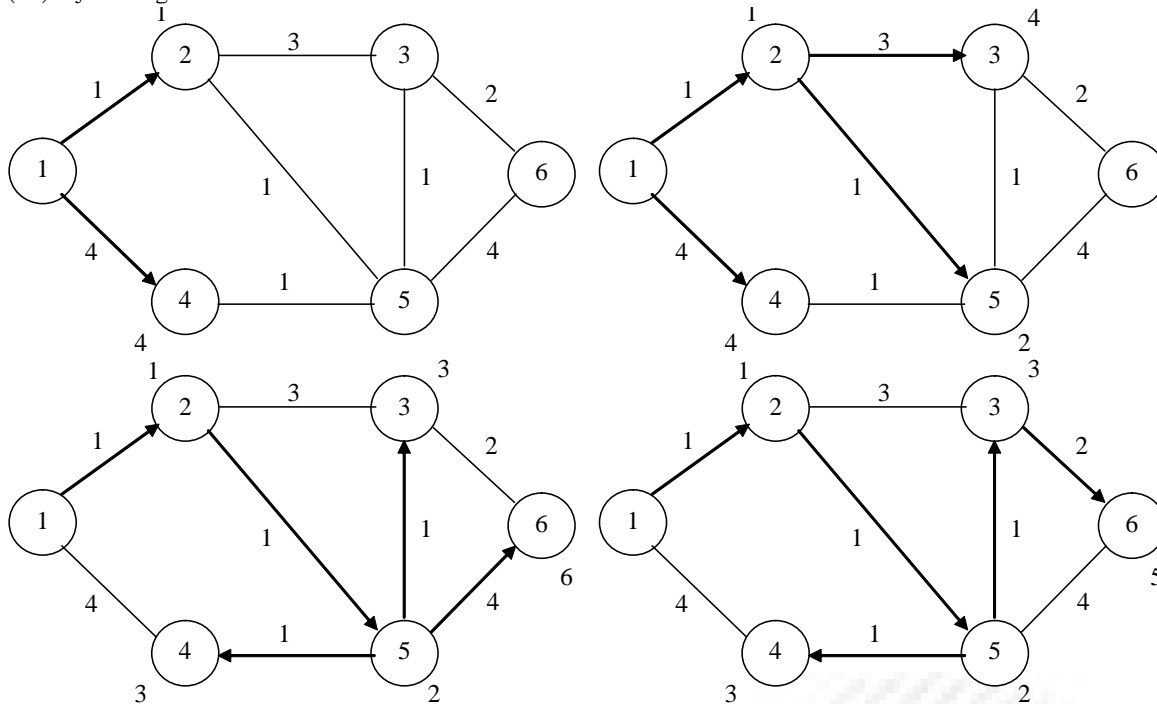
(二)請利用Dijkstra或Bellman-Ford演算法，逐步計算出下圖中從節點1到網路上其餘所有節點的最短路徑。其中，每個連線 (link) 上的數字代表此連線兩端節點的距離。(7分)



答：

(一)最低成本路由是選擇花費成本最低的路徑；而最短路徑則通常是選擇延遲時間最短的路由，兩者都可以使用像 Dijkstra's algorithm 來計算，其中的不同是使用不同的基準(metric)，最低成本路由使用費用做基準，最短路徑使用延遲時間。

(二)Dijkstra algorithm



八、請回答以下問題：

(一)假設一個CSMA/CD共享式乙太網路中，有三部主機A、B及C因同時送出其訊框而發生碰撞 (collision)。其中A及B為第一次發生碰撞，C為第二次發生碰撞。三部主機將停止其訊框傳送，且採用二進指數退回機制 (binary exponential back off)，分別等候一段隨機時槽 (slot time) 後再重傳。請問在隨後的等候時槽中，A、B及C三部主機又會因同時送

出訊框再發生碰撞的機率為多少？（5分）

(二)乙太網路中規定最小訊框 (minimum frame size) 為64位元組，其作用為何？（5分）

(三)假設一個FTP client正和一個FTP server建立連線，並以緩啟動 (slow start) 做為壅塞控制 (congestion control)，且FTP client端的接收緩衝空間 (window size) 大於50 MSS (maximum segment size)。試問FTP server需要經過多少往返時間 (round trip time, RTT) 後，才能在不必要等候client端的回應下，同時傳送超過41個MSS？（5分）

答：

$$(一) \frac{1}{2} \times \frac{1}{2} \times \frac{1}{4} + \frac{1}{2} \times \frac{1}{2} \times \frac{1}{4} = \frac{1}{8}$$

(二)為了使碰撞發生時，發送端可以確定能夠偵測到碰撞。

(三)經過 第一個RTT 之後，可以同時發送 2 個 MSS

經過 第二個RTT 之後，可以同時發送 4 個 MSS

經過 第三個RTT 之後，可以同時發送 8 個 MSS

經過 第四個RTT 之後，可以同時發送 16 個 MSS

經過 第五個RTT 之後，可以同時發送 32 個 MSS

(已超過50MSS的一半，改用加法遞增)

經過 第六個RTT 之後，可以同時發送 33 個 MSS

.....

經過 第十三個RTT 之後，可以同時發送 40 個 MSS

經過 第十四個RTT 之後，可以同時發送 41 個 MSS

九、在網路安全中，什麼叫做資料完整性 (data integrity)？我們可以用什麼方法驗證資料完整性？（10分）

答：

(一)資料完整性是指接收的資料與最初發送的完全一樣，沒有錯誤也沒有被篡改。

(二)可以使用文件摘要來保證資料的完整性，例如：MD5 或 SHA-1 等方式來產生文件摘要。