《計算機網路》

試題評析	第一題：公開金鑰匙加密所衍生的問題與解決。 第二題：自動重傳滑動窗協定的觀念。 第三題：乙太網路的二進位指數倒退演算法。 第四題：距離向量的計數至無窮大問題。 第五題：TCP壅塞控制中的快速重傳作法。
考點命中	第一題：《高點資通安全補充講義》第一回，王致強編撰，第 19 頁。 第二題：《高點資通網路講義》第二回，王致強編撰，第 30~31 頁。 第三題：《高點資通網路講義》第二回，王致強編撰，第 63~64 頁。 第四題：《高點資通網路講義》第四回，王致強編撰，第 16~17 頁。 第五題：《高點資通網路講義》第五回，王致強編撰，第 30 頁。

一、在網路中傳輸加密資訊，可透過接收方的公開金鑰加密，請說明可能衍生之問題，並說明解決的方法。（20 分）

【擬答】

(一)公開金鑰匙的優點：

1.不易被破解。

2.沒有鑰匙保管等問題。

(二)公開金鑰匙的缺點：

1.加解密速度較慢。

2.長時間使用固定的一組 private/public key，仍然有風險。

(三)解決方法：採用如 SSL 對每一個 session，在通訊前先傳送一組 shared key，使用對稱型加密法加解密，可以提高處理的速度，而且每個 session 都使用一組不同的 shared key 當 client 與 server 進行建立連線的 handshaking 時，由 server 先產生一組 public/private keys，並將 public key 傳送給 client，接著就用這組 key 相互繼續協商後續的安全性(如：加密法，身份驗證等等)，此時 client 用 public key 加密要傳給 server 的訊息，而 server 則以 private key 來加密傳給 client 的訊息，以使 client 能確認 server 的身份。

二、自動回覆請求（Automatic Repeat Request）可確保傳輸資料的正確性，請分別說明 Go-Back-N 與 Selective Repeat 兩種作法。（20 分）

【擬答】

(一)Go-back-N

1.使用負面確認訊息(或稱爲 reject)，要求傳輸端由一個特定序號的訊框開始，重送該訊框及所有後續的訊框。

2.訊框 N+1 損壞時的處理：

(1)當接收端收到訊框 N+2 而沒有收到訊框 N+1 時，認定訊框 N+1 遭損壞，於是送出 NAK N+1，以要求重送訊框 N+1 及以後的訊框。

(2)傳送端收到 NAK N+1 後，重送訊框 N+1 及其以後的訊框。

(3)接收端在收到訊框 N+1 之前，若先收到的其他後續訊框皆會捨棄；直到收到訊框 N+1 之後再回復接收新的訊框，並傳回確認訊息。

(4)接收端每次送出負面確認時，必須啓動計時器以防止負面確認框的遺失。

3.正面確認框 ACK N 損壞時，傳輸端可以從 ACK N+1，來確認訊框 N 已正確送達。但是如果有太多連續的確認框遺失時，仍會有問題，因此可以在傳輸端使用計時器，來啓動重送，以避免這個問題。而接收端收到重送的訊框，便可以知道確認訊息已遺失，便可以重新送出確認框，或以負面確認來要求對

方送出下一個希望收到的訊框。

4.背負式確認訊息(piggyback acknowledgement)：在全雙工的模式中，可以在訊框中加入一個確認序號欄位，以便做爲確認之用，這樣可以減少傳遞 ACK 的訊框數量。

(二)選擇性重送

1.分爲內隱式與外顯式兩種，前者只做正面確認(ACK)；後者(稱爲：選擇性拒絕)除了正面確認之外，還加上負面確認(NAK)。

2.當訊框 N+1 損壞時的處理：

(1)接收端對每個收到的訊框皆送須回一個確認訊息，例如，傳送端收到 ACK N, ACK N+2,…。

(2)傳輸端收到 ACK N+2 時，即發現缺少了 ACK N+1，表示訊框 N+1 發生問題，於是進入重送狀態。

(3)傳送端先暫停其他訊框的傳送，重新傳送出訊框 N+1，然後再恢復正常的訊框傳送。

3.當確認框 ACK N+1 損壞(接收端實際上已正確收到訊框 N+1)時的處理：

(1)傳輸端收到 ACK N+2 時，發現缺少了 ACK N+1，於是重送訊框 N+1。

(2)接收端收到重覆的訊框 N+1，捨棄之，並重送一次確認框 ACK N+1 給傳輸端。

三、請說明 Ethernet 媒體存取控制（Media Access Control）的詳細作法，並說明其後退（Backoff）時間的產生方式。（20 分）

【擬答】

Ethernet 的二進位指數倒退演算法：

1.在 IEEE 802.3 中，採用 slotted 1- persistent CSMA/CD 方式來發送訊框，屬於競爭型的共用通道。

2.採用 Binary Exponential Back-off Algorithm 來處理碰撞的問題。

3.第一次碰撞時，station 會選擇等候 0 或 1 個 slot time。第二次又發生碰撞時，則 station 會選擇等候 0～3 個 slot time。依此類推，第 k 次碰撞時，會選擇等候 0～2k-1 個 slot time。

4.超過 10 次碰撞之後，固定在 0～1023 time slot 之間隨機取一個亂數，不再增加，以免等候時間太長。

5.到達 16 次碰撞之後，就自行放棄發送此一訊框，後續問題可以交給上層(通常是 Transport layer)去處理。

6.當碰撞的 stations 個數較少時，可以避免等候太多的 time slots；而碰撞的 stations 較多時，也可以逐步調整到使碰撞機率較少的狀況，因此效果不錯。

7.確認訊息(Acknowledgement)

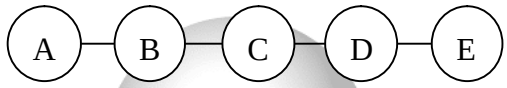
接收端在收到正確的 frame 之後，必須送回一個確認訊息給傳送端。可將接收完成之後第一個 contention slot 保留給接收端做爲傳送確認訊息之用，以減少 ACK 與其他訊框碰撞的機率。

四、請說明在距離向量（Distance Vector）路由演算法中的 Count-to-infinity 問題，並詳細說明如何避免該問題發生時產生封包迴路的問題。（20 分）

【擬答】

(一)無窮大的計數(Count-to-Infinity Problem)

1.Distance vector routing 具有一項特性，就是對於較佳的路徑，可以很快地傳播週知；但是，對於不好的狀況，卻可能需要較多的時間來傳播給所有路由器知道。例：在下面的例子中，在 A 與 B 間斷線時，每個 router 會一步一步往 ∞ 逼近，但卻無法迅速到達 ∞。



1	2	3	4
3	2	3	4
3	4	3	4
5	4	5	4
5	6	5	6
7	6	7	6
7	8	7	8

(二)常用解決方法有二

1.將網路中最長的一條路徑長度再加一定義當成 ∞，當路徑長度增加到達此值時，即爲無窮大。

2.水平切割法(Spilt Horizon)：基本方式與 distance vector 相同，唯一的改變是在與相鄰 routers 交換資訊時，若 router A 到 router C 的最佳路徑是經由 router B 到 router C，意即 A 往 C 的路徑是由 B 學習而得到的，則當 A 定期要傳 routing table 給 B 時，不能再將此一往 C 的路徑，又傳回給 B。

五、請說明 TCP 快速重傳（Fast Retransmit）的作法，並說明使用 TCP 快速重傳的理由。(20 分)

【擬答】

(一)快速重傳的作法

1.TCP 重覆收到 3 個相同序號的 ACK 時，就會重傳區段，不用等到計時器逾時。並進行壅塞避免的處理，即將窗口減半。

2.計時器逾時時間(RTT)的調適性公式，其中$0 \le \alpha \le 1$

$$RTT(t) = \alpha \times RTT(t-1) + (1-\alpha) \times Sample(t-1)$$

(二)快速重傳的理由：當傳輸端收到重覆的 ACK 時，可能是區段遺失了，於是立刻重傳，而不用等到計時器逾時才重傳。