1. **ABR:根據目前網路的流量，調整傳輸量的一種服務，保證你有個最低的傳輸速度(是一種擁塞控制和流量管理的方法)**
2. **CBR:分配個固定的通道大小，而我所能用的傳輸量就是這麼多**
3. **SDN 這個網路技術→利用軟體來自動改變網路的架構，**

**(傳統的網路設備要做:data plane 和control plane)這個技術就是data plane、control plane分開工作，然後利用server來 集中式去做control plane的工作(這個server會用軟體決定要如何處理封包路徑)；而原本的網路設備不需要會那麼多的功能，它只要會data plane的工作就好**

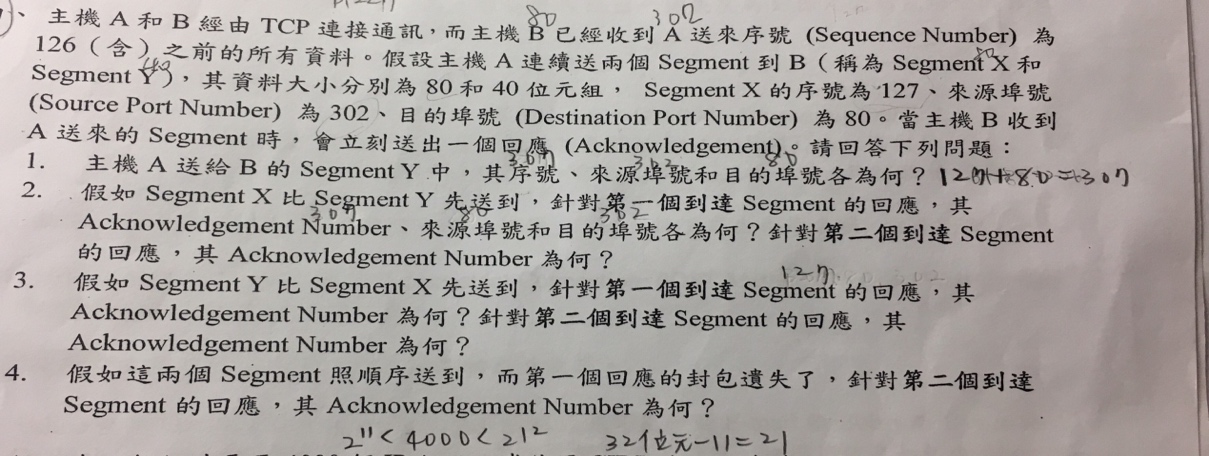
**Open flow 以軟體定義網路為基礎，一個新的網路架構**

1. **ICMP  :   偵測錯誤與回報的機制**
2. **traceroute:封包一開始TTL=0，然後慢慢遞增+1送出封包，當TTL為0時，回傳一個icmp封包(type =11)可以知道這條路徑上的路由器位址、狀況 ; 接著，直到目的端主機一開始設一個目的端沒有用的Port，送回目的端的資訊。**
3. **\*ping  :  向目的端主機發送echo request (type=8)  在等候  回應echo reply (type=0)   以測試和目的端主機是否有正常的過去和回來的通路。**
4. **EDC (Error-Detection and –Correction)是個統稱:錯誤偵測and修正，加在資料後面的欄位，用來偵測錯誤並修正的{位元}， ps. Parity check 或 CRC 就要用到這個**
5. **CSMA/CD 發送資料前，偵測通道是否閒置，如果通道繁忙則延緩傳輸，如果閒置則傳輸資料並持續偵測通道，為了盡快知道有沒有碰撞以減少浪費的時間；假如發生碰撞，則會等待一段隨機的時機 ( 投影片36頁的:*binary exponential backoff公式* ) 然後再次偵測通道、傳送。**
6. ***exponential backoff : n為連續發生碰撞次數(最大值為10)，r為隨機延遲時間，r 是 0,1,2,...2^n – 1，表示可能延遲的2^n – 1 個時間槽***
7. **AQM(佇列管理的演算)主動式佇列管理，簡言之 指在佇列滿、壅塞之前，就先把裡面的封包丟掉(避免發生壅塞)**
8. **AQM中有個叫RED(early random detection)演算法:**

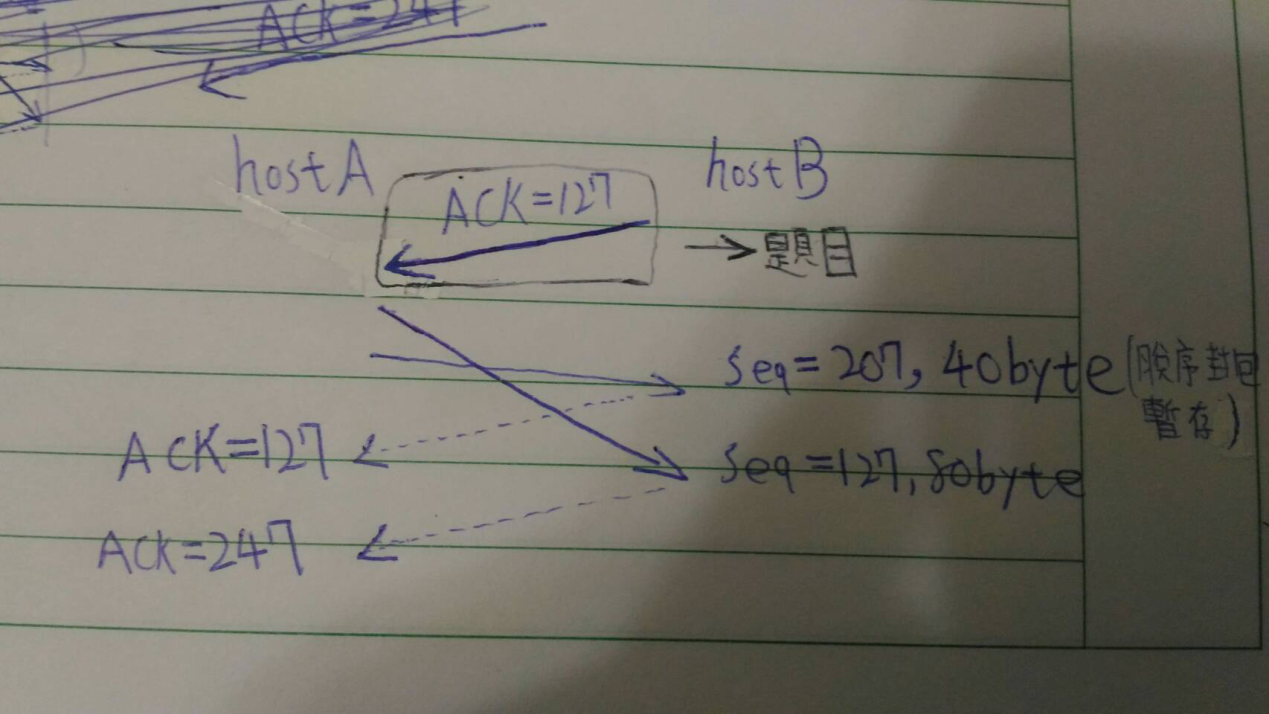
**當 平均佇列長 <最小佇列門檻時 讓封包送入佇列**

**當 平均佇列長 >最大佇列門檻時 佇列中的封包丟掉，傳送端因此也可得知可能發生壅塞而少傳資料**

**在這兩個門檻值之間我到底是要留還是要丟封包，會由某個機率決定**

12.

的第3小題



1. **NAT是為了解決IPv4位址不足，而有虛擬IP，能將經過NAT的封包來源端的IP和port轉換為NAT自己的IP和port，並將資訊(原本舊的IP和port)存在NAT的table裡，封包傳回時，也能將新的IP和port轉回成相對應的虛擬IP和原本的port**
2. **CIDR是為了避免IP位址浪費，以IPv4為例，能隨意分割32位元的ip位址,分為2部分網路的位址和主機的位址，根據使用需求能夠有效率地去分配主機位址的數量**

**a.b.c.d/x 的x表示網路位址的位元，後面的位元為主機位址的位元**

1. **classful  address  :  以IPv4為例，將32位元的ip切割成4個8bits的網路  ，命名A、B、C、D網路，A網路的前8位元為網路位址，後24位元為主機位址，由於網路ip位址的長度被限制，可能某個子網路不需要這麼多個主機位址，造成IP位址的浪費**
2. **多工 : 在來源端執行，將各個socket的資料和header 分封在一起成segment ，然後再丟給網路層**
3. **網路位址中能使用的主機數量要-2，因為那是網路的位址(頭) 和 廣播的位址(尾)**
4. **\*MSS : 傳輸層的封包(segment)的最大值**
5. **\*MTU :  資料連結層所能傳輸封包的最大長度**
6. **\*longest prefix matching :  當封包的IP位址符合fowarding table內2個以上的目的端位址範圍，會選擇對應最長址首的連接介面**
7. **anycast :  加入一個群組並發送資料，但只有群組裡的最近某個節點會收到 ps.(unicastㄧ對ㄧ傳輸,broadcast送給全部，multicast一對多，就是說我們先加入一個multicast群組、然後將資料送給這個群組裡的所有節點)**
8. **TCP   :  可靠性傳輸、資料串流、握手協定、全雙工、流量控制、point to point**

**UDP :  不可靠傳輸、無流量控制、無壅塞控制、半雙工、傳輸效率高**

1. **TCP是位元組串流，當他序號是X，傳輸資料長度為100時，下次的序號是X+100**
2. **主機B即使沒有資料要傳給A，也依舊會回送ack的訊息給主機A**
3. **rwnd 受到tcp流量控制所影響，cwnd受到tcp壅塞控制所影響**
4. **slow start  :  一開始在slow start狀態下cwnd為1MSS，然後MSS倍數成長，直到超過門檻ssthresh(若逾時或封包遺失ssthresh會設為當下MSS的一半)**
5. **\*AIMD  :  超過ssthresh時，進入AIMD狀態，cwnd每次+1MSS  線性成長，如果封包遺失(收到3次重複ack)則會將cwnd設為當下cwnd的一半**
6. **快速復原:是收到3個重複ack，不會等到逾時則直接重送。**
7. **\*reaction  to timeout :  如果是3次重複ack則cwnd被減半並且+3MSS(進入AIMD狀態)，ssthresh設為cwnd / 2) ；如果是逾時則cwnd被設為1(進入slow start狀態)ssthresh  設為  cwnd  / 2。**
8. **TCP  option  欄位:擴充tcp的功能**
9. **\*sIPv6和IPv4共同欄位:  source   address、destination   address、  version、(ipv4) type of service 和 (ipv6)priority用途相似，(IPv4)TTL對應(IPv6)hop limit用途相似**
10. \*IPv6:位址長度:128位元，不允許封包在路由器內切割，沒有option欄位，新增了flow label欄位(區分要以相同方式處理的封包)，沒有checksum，欄位長度固定
11. IPv4: 位址長度:32位元，允許封包在路由器內切割，有option欄位，沒有flow label欄位，有checksum，欄位長度不一定固定
12. Channel partitioning 分割成許多小塊(通道、時槽等等)並做分配，不發生碰撞 ex : FDMA 、TDMA 、CDMA
13. Random Acess : 讓節點自由競爭，使用全部的通道進行傳送，但可能碰撞(2 host同時發送的訊號混雜互相受到干擾)，所以要能偵測到碰撞並恢復。補充:TDMA是事先分配好誰在哪個時槽送資料 ex : CSMA/CD、CSMA/CA
14. Taking turns : 有個特殊的封包讓節點輪流存取， 不發生碰撞。Ex:token ring , polling
15. DHCP:完成管理位址的工作能自動去尋找網路中的DHCP伺服器，然後幫client 自動去取得一個臨時的IP
16. CRC: G為除數，D為被除數，r是被除數後面要部幾個0，(r = 根據除數的位元長度-1)，然後餘數(ex : 001)就是CRC，加上data(假如10010)，就會長這樣(10010001)
17. CDMA : 他可以讓甲和乙的資料混在一起，編碼一起送，而對方需要有特殊的code就可以在混雜的資料中分析出來他想要得到的資料。
18. 同位元偵測:我們會相講要偶數還是奇數運算，通常我們用偶同位，如果是要偶同位，要看1的個數，如果有1的個數為偶數最後的同位元你要放0，如果是奇數最後同位元要是1，如果1的個數不是偶數個就是代表錯誤。
19. 解多工 : 利用port 來將封包送到相對應的應用程式
20. DOCSIS:這是使用第四台 Cable Modem 上網時，其鏈結層所使用的多重存取機制。
21. Destination-based forwarding : 比對目的端的IP address，去決定轉送到適當的link
22. Generalized forwarding: 例如SDN，根據封包標頭區裡面的值(ex:MAC、IP位址等等)去比對，決定動作(轉送)
23. Data plane : 用硬體實作，依據封包的header欄位，查表決定抵達input port的封包該轉送到路由器得哪個output port
24. Control plane : 用軟體實作，決定封包從來源端到目的端的路徑
25. Match + action : 去比對欄位然後執行相對應的動作

Match的欄位有 : 來源端或目的端的MAC、IP位址，TCP/UDP的port

Action的動作有 : 轉送、丟棄封包、修改來源端的IP address port(ex : NAT)

1. Thread : 一個Process底下，有許多自己的分身，而thread server就是可以處理多個socket裡的資訊
2. single bit indicating congestion :IP header 裡的如果ECN=11代表壅塞，則封包到達目的端時在TCP header裡ECE=1回傳給傳送端、ABR
3. 排程