UDP (dest IP address , dest port)

TCP (source IP address , source port , dest IP address , dest port)

2. Sender: 將內容轉為16bit 整數,計算整數的和後放入UDP checksum 發送

Reciever: 驗證內容的和與checksum 是否相符

3. 1110 0110 0110 0110

1101 0101 0101 0101

Sum 1 1011 1011 1011 1011

checksum 0100 0100 0100 0100 1

4. 串流媒體、VoIP 語音、網路遊戲

5. rdt2.0

Rtd\_send() --> rtd\_rcv() (檢查封包是否有誤)

如果無誤回傳 ACKs eceiver explicitly tells sender that pkt received OK

如果有誤回傳 NAKs receiver explicitly tells sender that pkt had errors

如果發送端收到 NAKs 會再重傳一次封包

6. 因為只需要判斷封包是否要重傳對於簡單的停止等待協議

只需要one-bit sequence number即可判斷

7. sequence number 為傳送封包的序號,可以讓接收端與傳送端知道

目前是傳送到第幾個封包

8. rdt3.0

rdt\_send() and start timer --> rtd\_rcv()

如果在timer 時限內有收到ACKs or NAKs = 沒有發生packet loss

如果在timer 時限內沒有收到回覆就會在重新發送一次封包

9. timeout 設太短：導致 sender 一直重送

timeout 設太長：封包已經掉了，但 sender 卻很晚才發現

10. Pipeline protocol : 把原來只送一次封包收一次封包 , 改為一次可以複數傳遞與接收

11.

訊息重送

Go-back-N 只要發生錯誤,就要把windows裡的封包全部重送

Selective repeat 只需要重送發生錯誤的封包

Window buffer

Go-back-N 在recieve 端沒有 window buffer

Selective repeat 在recieve 端有 window buffer

Number of timers

Go-back-N 傳送時只要一個timer

Selective repeat 傳送時windows 有多少封包就要多少timer

12. 2w-1

13. estimated RTT + 4\*DevRTT

14.

Delay ACK. Wait up 500ms for next segment. If no next segment, send ACK

15.

在同一筆資料收到超過3個ACK 明顯表示這區段已遺失,

所以會立刻重送遺失區段 , 而非等到timeout 才重送

16 sender 會依照 reciever 的 buffer 來控制傳送流量,不會讓reicever一次

接收太多封包

17. GBN retransmission 窗格內封包全部傳送 receiver window = 1

TCP tetransmission 僅重傳有掉的封包 , receiver window = N

18. Sequence number: 表示這個資料段中的第一個資料位元組,被當成sender端發送

資料的counter

Acknowledge number: reciever端所期望接收到下一個封包的序號

=sequence number+1

19. (1.) variable delay

(2.) retransmitted messages (e.g.req\_conn(x)) due to message loss

(3.) message reordering

(4.) can’t see other site

20.

1.client 傳送 FIN

2. server 回傳 ACK

3.server 回傳 FIN (server 準備關閉)

4.client 傳送 ACK (server 已經關閉, client 等待 2倍 max segment time 然後關閉)

21. initially cwnd 設定為1

--> 每次RTT後 cwnd x 2

--> done by incrementing cwnd for every ACK received

22. sender 端加速傳送速度的方法

Additive increase:每次RTT後 cwnd+1 直到出現loss

Mutiplicative decrease : 在偵測到loss 後把cwnd/2

1.

Data plane: 將數據從路由器的輸入移動到適當的路由器輸出

Control plane: 決定數據從route source 端到destination 端

Determine the route taken by packets from source to destination

2. 路由表中的每個表項都指定了一個網絡，所以一個目的地址可能與多個表項匹配。最明確

的一個表項即子網掩碼最長的一個

3.

1. memory

透過記憶體進行交換:封包來到輸入埠，然後該封包會從標頭中取出目的位址

，查詢轉送表，找出適當的輸出埠，然後將該封包複製到該輸出埠的緩衝區

2. bus

透過匯流排進行交換:輸入埠會透過共用匯流排直接將封包傳輸到輸出埠，而無需

要繞送處理器介入。匯流排是共用的，所以匯流排在同一時間內只能傳輸一個封包。

3. cross bar

透過互連網路進行交換:將n格輸入埠連接到n個輸出埠的互連網路。抵達輸入埠的

封包會沿著連接到該輸入埠的水平匯流排移動，直到抵達前往其目標輸出埠的

垂直匯流排為止。

4. 當 switch fabric 比 input line 速度慢時會發生

5. 前面的封包出現問題導致後面封包無法順利傳送

6. 當封包超過MTU時 會進行IP fragment

Header 相關訊息有 ,

Length 代表有多少byte , ID , fragflag 代表是否為最後一個fragment ,offset 為 定位紀錄

7. Time to live : 代表封包存活時間 每經過一個router 就會減1,當TTL直為0時就會被丟棄

Upper layer : 表示傳輸層協議 ex 6 TCP , 17 UDP , 1 ICMP

8. 是一個用於給使用者分配IP位置以及網路上有效地路由IP封包的對IP位址進行歸類的方法

9. 先將 datagram 存到buffer 排隊後經由 link layer 傳送到 router 輸出端

Scheduling 方法

1. FIFO 先進先出

2. priority scheduling: 會先傳送優先度較高的datagram

3. Round Robin: 每個類別平均分配傳送

4. Weighted Fair Queuing : 會算出給每個類別傳送的百分比來做輸送

10.Network part :指定分配的唯一編號

Host part : 分配IP的一部份,在此網路為識別機器的編號

11. NAT translation table會儲存 WAN side addr 與 LAN side addr 可透過查表

來知道這段訊息是由哪個device 傳出去或是誰傳給哪個device

12.可以減少網路中IP數量的方法,主要方法是透過選定多個router 合併起來成一個來表示

13. 同一台 router 會持有數量不等的同一個字首的 ip 去分配給連進來的主機，所以透過字首去找變可最快找到接近該台主機的 router，而要 longest 的原因就是因為每台 router 分配到的字首長度不一樣，所以找最長長度的字首才能確定哪台才是我真的想要的