

國立清華大學試卷

| 記 分 | | | |
|-----|----|------------|-----------------|
| 1 | -0 | 2 | -0 |
| 3 | -0 | 4 | -0 |
| 5 | -0 | 6 | -0 |
| 7 | -0 | 8 | -0 |
| 9 | -0 | 10 | -0 |
| 11 | -0 | 12 | -0 |
| 13 | -0 | 14 | -0 ✓ |
| 15 | -0 | 16 | -0 |
| 17 | -0 | 18 | |
| 19 | | 20 | |
| 總 分 | | 100 109 | |

所 系 資工系

科 目 計網概

學 號 [REDACTED]

姓 名 [REDACTED]

日 期 12/6

1. ① 不用建 connection : 減少建 connection 的 delay
② server 跟 client 不需要存 connection 的 state \Rightarrow simple
③ header 比較小, 可以送比較多的資料
④ 沒有 congestion control : 可以用 available 的最高速率送
有些 (ex. 網路電話) application 可以容許資料掉一些且對速率很敏感 \Rightarrow 就會採用 UDP

2. ① 假如是已做好偵測 error 的 mechanism,
只需加上 countdown timer, 當 timeout 就自動重送 packet

② 假如是還無法偵測 error 的 mechanism
要加上 countdown timer 及 sequence number

* timeout 會自動重送

* 收立端可以用 sequence number 判斷 packet 是否 duplicate

3. (a) 如果 timeout, 把所有還沒收到 ack 的 packet 全部重送 (只有一個 timer 對 seq # 最小的 unacked packet)

(b) 每個 packet 有自己的 timer, 當 timeout(N) 把 packet N 重送且 restart 該 timer (即各別重送 unacked packet)

\Rightarrow receiver 端要加 buffer, 先把 un-order 的 packet buffer 起來, 等 gap 補齊送給 application layer

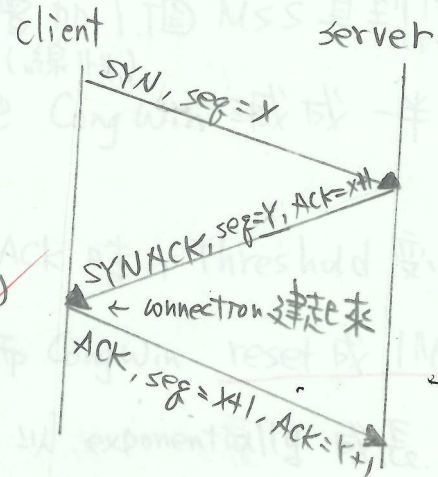
4. ① TCP 只有一個 timer 系統，還沒收到 ACK 最老 (oldest) 的 packet 用；而 SR 是每個 unacked packet 都有自己的 timer
- ② TCP 有 slow start 及 Congestion avoidance 會改變 window size
- ③ TCP 有 fast retransmission，當收到 3 個 duplicate ACK 時就會重送，但 SR 只有 timeout 才會重送

5.

Step 1: client 發出 SYN 給 server ($s=1$)
裡面含有 client 的第一個 seg # ($=x$)

Step 2: server 發出 SYNACK 給 client ($s=1, A=1$)
裡面含有 server 第一個 seg # ($=y$)
及下一個想收的 byte # ($x+1$) (ACK)

Step 3: client 發出 ACK 給 server ($A=1$)
裡面含有 seg # (data 的第一個 byte)
及 ACK (下一個想收的 byte)
有可能已含有要傳送的数据



6. ① delay 造成重送 \Rightarrow 即在網路裡可能有多個相同 packet 的 copy
- ② 太過擁塞造成 packet 掉了，還要再送一次 \Rightarrow 浪費頻寬 (buffer 滿了)
- ③ 如果 packet 掉了，前面所有用來傳這個的頻寬都浪費了 (含沿路經過 router 的頻寬)
7. (a) 當發生 packet loss 或 delay 的時候
- (b) 透過調整 window size 的大小 (CongWin) 而改變速率

8. (a) 當收到一個 ACK 就把 Cong Win 加 1 個 MSS

⇒ 一個 RTT, Cong Win 會變 = $\frac{MSS}{CongWin}$

(b) 收到一個 ACK, 會增加 $\frac{MSS}{CongWin}$ 個 MSS

(ex, CongWin = 8 → 收到一個 ACK 增加 $\frac{1}{8}$ 個 MSS)

⇒ 一個 RTT, Cong Win 增加一個 MSS

9. additive-increase: 一個 RTT, Cong Win 增加 1 個 MSS 直到 loss (線性)

multiplicative-decrease: 一旦發生 loss, 把 Cong Win 減成一半

10. TCP Tahoe: 當收到 3 個 duplicate ACK 時, Threshold 變成當時 Cong Win 的一半, 而 Cong Win reset 成 1 MSS 進入 SS state, Cong Win 以 exponentially 成長

TCP Reno: 當收到 3 個 duplicate ACK 時, Threshold 變成當時 Cong Win 的一半, 而 Cong Win 變成 Threshold (即 Cong Win 減半), 進入 CA state, Cong Win 以線性增加

11. (a) $155 - 35 = 120$

(b) first segment loss ⇒ Host B 想收 first segment

⇒ ACK number = 35

12. ① $\frac{980 - 20}{500 - 20} = 3 \Rightarrow 3 \text{ 個 fragments}$

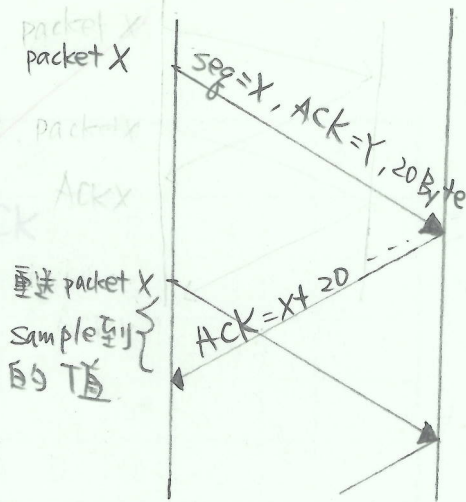
② $\frac{500 - 20}{8} = 60$

⇒ offset 依序為 0, 60, 120

13. 会有錯誤, 在还没收到第一次送的 packet 前就重送, 並且測量重送的 packet 的 RTT, 此時如果收到第一次送出 packet 的 ACK

會以為是第二次 packet 的 ACK

→ 量到錯的 RTT = 送出第二次 packet ~ 收到第一次 packet 的 ACK 的時間 → 並非 RTT



14.

(a)

prefix matching

link interface

link interface

11100000 00000000

11100000 00000001 00000000

11100000 00000001

otherwise

0

1

2

3

(b)

① 符合第 4 個 entry → interface 3

② 符合第 3 個 entry → interface 2

③ 符合第 2 個 entry → interface 1

how

15. (a) exponential 部份: $[1, 5], [20, 25]$

(b) 線性部份: $[5, 15], [16, 21]$

(c) CongWin 降一半 \Rightarrow triple duplicate ACK

(d) CongWin 降為 1 MSS \Rightarrow time out

(e)

| round | 數量 | 送的 packet |
|-------|----|-------------------------------------|
| 1 | 1 | 1 st |
| 2 | 2 | 2 nd ~ 3 rd |
| 3 | 4 | 4 th ~ 7 th |
| 4 | 8 | 8 th ~ 15 th |
| 5 | 16 | 16 th ~ 31 th |

\Rightarrow 故 25th 在 round 5 送出

16. ① Local 電腦有 (Local IP, Local Port) 要傳資料到 (dest IP, dest Port)
 \Rightarrow 先送到 IP 分享器

② 到了 IP 分享器, 把 Local IP 跟 Local Port 轉換成 (NAT IP, Port X)
送出給 (dest IP, dest Port) 並把對應的轉換記在 NAT translation table. (NAT assign)

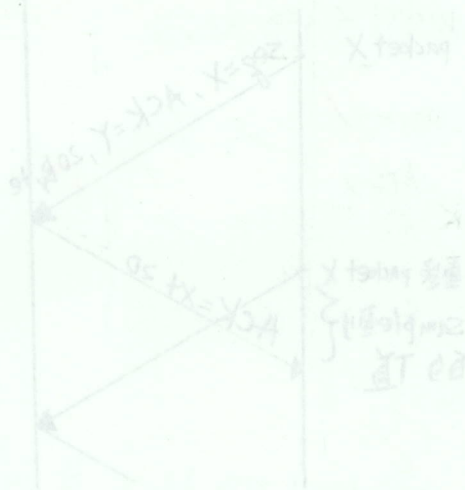
③ destination 收到後回給 (NAT IP, Port X), 送到 IP 分享器

④ IP 分享器收到後查表, 把 (NAT IP, Port X) 改成 (Local IP, Local Port) 送給 Local computer

17. ① 挑一個比較不可能被使用的 Port

② 送資料到 destination 的那個 Port, 且依序 TTL=1, TTL=2...
(每個 TTL 值做三次)

③ 當 TTL=k, 送到第 k 個 router, 因為 TTL 減為 0, router 會回報 ICMP 的 error message (TTL expired), 在 ICMP 裡有 router 的 IP, 透過發 DNS request 可以得到 router name, 並且有 timer 測量去、回時間 (左邊還有)



若 $X = -1$ 表示此包在發送前，發送者已收到

的 packet 且發送者，發送前已收到

若 $X = 0$ 表示此包在發送前，發送者已收到

的 packet 且發送者，發送前已收到

若 $X = 1$ 表示此包在發送前，發送者已收到

的 packet 且發送者，發送前已收到

TTL 非 0 表示此包在發送前，發送者已收到

link interface

prefix matching

0

00000000

00000111

1

00000000

10000000

00000111

2

10000000

00000111

3

00000111

① 若 $X = 0$ 表示此包在發送前，發送者已收到

② 若 $X = 1$ 表示此包在發送前，發送者已收到

③ 若 $X = 2$ 表示此包在發送前，發送者已收到

WON

- ④ 當 $TTL = n \rightarrow$ 抵達 destination computer, 因為選擇的 Port 沒有 process 用 (應該) \rightarrow 會回報 ICMP (dest port unreachable) Error message
- \rightarrow 從 Error 種類不同可知抵達 dest, trace route 結束, 且 ICMP 也有 destination 的 IP, 透過 DNS 及 timer 可得 name 及去、回時間