

Design challenge report

Group 6

整份 code 可以分為三個部分：

1. 每個裝置建立自己的 ID 和 Address。
2. H 發出 flag message。
3. A~G 收到 flag message 並建立到 H 的 route table，五秒後開始傳送自己的 ID 跟 address 給 H。H 收到 A~G 的 message 後 print 出這份 message 的來源(i.e ID and address)。

每個裝置傳送的 message 有三種 (A 和 H 比較特別，只會傳送兩種 message)，不同種 message 有各自的用途，根據 array[0]區分不同種類的 message，故裝置收到一個新的 message，可以從 array[0]得知屬於哪一類，做出相對應的處理。

第一類：array[0] = 0，代表的是確認 ID 跟 address 的廣播封包，這個封包的長度是 10，array[1]存的是發送封包的人的 ID，array[2]~array[9]預設都是 0，當有 address 被使用到時，就把相應的欄位改為 1(address = 1,把 array[2]改成 1，address = 2，把 array[3]改為 1，依此類推)，如此一來，就可以讓聽到的裝置知道哪幾個 address 已被使用，哪幾個還沒有。

第二類：array[0] = 1，代表的是 A~G 回傳給 H 的單播封包，包含 ID 跟 address。array[1] = address，array[2] = ID。

第三類：array[0] = 2，代表的是 flag 封包，array[1]=ID，array[2] = address。ID 和 address 欄位放的是上一個裝置的 ID 和 address。

第一類和第三類訊息是用 broadcast 的方式，而第二類訊息是用 unicast 的方式避免 broadcast storm。

Step 1：

建立 ID、Address：一開始，當我們把板子接上電之後，會開始偵測有沒有收到 array[0] = 0 的封包，並且先預設一串長度為 10，內容都是 0 的 array。過了五秒鐘之後，如果這期間沒有收到任何封包，表示自己是第一台裝置，故把 ID 設為 1，並且隨機從 1~8 之間選一個當作自己的 address(Demo 時 address 為預設)；如果有收到封包，先判斷封包內容，其中，array[1]代表收到的 ID，array[2]~array[9]代表哪幾個 address 被用過了，到最後，僅留下所收到最大 ID 的資料，覆蓋到預設的 array 之中，自己的 ID 就是所收到最大 ID +1，

array[2]~array[9]之中剩餘還是 0 的位置，代表自己可以用的 address，再從中隨機挑選一個當作自己的 address(Demo 中 address 為預設)。之後用自己的 ID 覆蓋掉 array[1]，把自己 address 對應的欄位改成 1，之後 broadcast 出去。確認 ID 後，LED 燈會開始閃爍。當收到 array[0] = 0 且 array[1]比自己的 ID 大的時候就停止 broadcast (表示 id 在自己之後的裝置已經建立 ID 和 address，故可以停止廣播自己的資訊)，另外，在收到 array[0] = 2 的時候也停止傳輸(因為 H 不會 broadcast 第一類封包，若沒有後者條件 G 會一直 broadcast 自己的 ID 和 Address)。

當裝置看到自己的 ID=8(即 H)，表示所有裝置已建完 id 和 address，H 廣播 flag message。

Step 2：

傳送 flag：H 裝置 broadcast flag message (第二類)：array[0] = 2，array[1] = H's ID，array[2] = H's address，H broadcast 一秒後進入休眠 (一足夠長時間確保 G 會收到)。當 A~G 任一個收到此種類 message，會判斷 array[1]是否比自己的 ID 大，是的話，代表是從靠 H 的方向來的，把 ID 跟 address 記錄起來，作為到 H 的 relay node。array[1]比自己的 ID 小時，代表是從 A 的方向傳來的，忽略掉從 A 方向傳來的封包。

建立 relay node 的方法是，只聽第一個收到的 flag message，所以 relay node 可能是 one hop 或 two hop。之後建立一個 message，array[0] = 2，array[1]=ID，array[2] = address，broadcast 出去。Broadcast flag message 的時間，從收到的第一個 flag 開始，持續 broadcast 五秒鐘，五秒鐘後，進入 step 3。

Step 3：

回傳自己的 ID 跟 address 給 H：step2 結束後，每個裝置都會有自己到 H 的 relay node。把自己的 ID、address 放入一個 array 中，array[0] = 1，array[1] = address，array[2] = ID，並且 unicast 給 relay node，最後到達 H。H 有兩個 array，h_addr 跟 h_t，分別紀錄每個裝置的 address 和收到的時間。每次接收到 array[0] = 1 的訊息時，就判斷他的 ID，把 address 放入 h_addr[ID-1]，接收到的時間(從 H 醒來開始記)放入 h_t[ID-1]之中，當 h_addr[ID-1]跟 h_t[ID-1]裡面已經有值，代表已經收過，就忽略掉這個封包。

討論

讓每個裝置知道到 H 的 relay node，在步驟 3 時，並不是用廣播或 one-hop

的傳輸方式。有可能建立出： $A \rightarrow C \rightarrow D \rightarrow F \rightarrow H$ ， $B \rightarrow D \rightarrow E \rightarrow G \rightarrow H$ 等 two hop 的路徑，可以加快傳輸速度，也可以避免過多封包在裝置間重覆傳送。

一開始我們是想建立 two-hop relay node 到 H，即每個裝置到 H 的最短路徑，但 two-hop 相對 one-hop，雖然路徑變短了，但封包的遺失率變高，需要重傳更多次，於是我們索性不管，決定順其自然，取決於當時的傳輸環境。

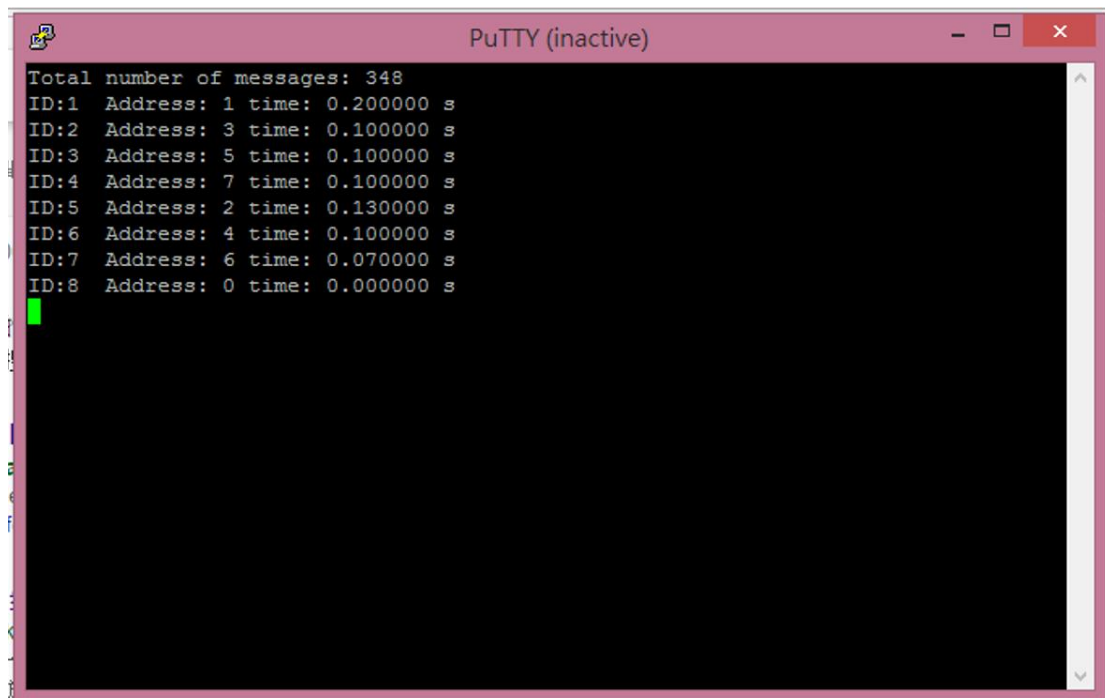
所以我們建立 relay node 的方式是只聽第一個收到的 flag message，不管是來自 one-hop 或 two-hop，若是 one-hop，表示當時環境品質可能很差，two-hop 傳輸會有很高的封包遺失，反之若環境品質高，則有高機率建立 two-hop 的 relay node。

在接收 flag 的時候可以有改進的空間，因為我們僅用收到的第一個 flag 來做事，但是也許可以讓裝置聽一段時間，由這段時間內收到的封包數以及 ID 的距離來做運算，最後決定出最佳的路徑。EX：B 在這期間收到了來自 C 的 flag 有 1000 個，收到 F 的 flag 有 500 個，收到 G 的 flag 有 10 個，也許最後就選擇留下 F 而不是 C。

另外在步驟 2 的時候，我們並沒有加入 H 的時間資訊，所以每個裝置計時的方式是由收到 flag 開始+5 秒，會有傳輸時間上的誤差。比方說，由 H 傳出的 flag 到達 A 需要經過 t_1 秒，A 傳自己的資訊到 H 需要 t_2 秒。那 H 是在 $0 + 5$ 秒鐘時醒來，但是 A 卻會在 $0 + t_1 + 5$ 時才判斷 H 應該醒來了，所以開始傳輸，H 接收到 A 所傳的資料變成了 $0 + t_1 + 5 + t_2$ 秒，會多了 t_1 這個誤差。

Demo 結果

Demo 時一共試了兩次，由 terminal 來讀取資料時失敗了，因為我們印出的東西有點多，所以後來改用 putty，如下圖。由圖可以看到，到把資料印出來(收到所有 A~G 的資料)之前，一共收到了 348 個封包。收到的時間如圖內所示，我們所用的最小單位是 0.01s，H 本身不記，所以 ID:8 都是 0。ID:1 時間最久，ID:7 最快，中間差不多但是 ID:5 比其他的久一點，大致上符合我們的猜想。



```
Total number of messages: 348
ID:1 Address: 1 time: 0.200000 s
ID:2 Address: 3 time: 0.100000 s
ID:3 Address: 5 time: 0.100000 s
ID:4 Address: 7 time: 0.100000 s
ID:5 Address: 2 time: 0.130000 s
ID:6 Address: 4 time: 0.100000 s
ID:7 Address: 6 time: 0.070000 s
ID:8 Address: 0 time: 0.000000 s
```

從 packet sniffer 來看所傳送的 data：

先檢查傳輸 flag 時的封包，從 P.nbr 5165 ~P.nbr 6358 之間。

這是依序聽到的封包，可以看到每一個 B~G 都有收到

Flag 並轉傳，而 A 並不傳 flag，僅收取 flag 而已。

mac payload(02 addr ID)
02 08 08
02 07 06
02 02 03
02 06 04
02 05 02
02 03 05
02 04 07

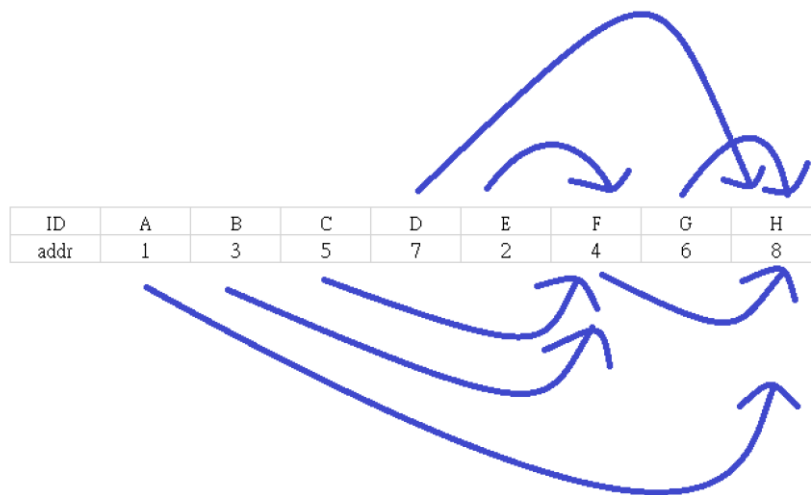
再來檢查 A~G 傳送封自己的 ID、addr 給 H 的動作：

從 P.nbr 6358 開始，出現了第一個 mac payload 開頭為 01 的 data，檢視過後，

可以建出一個表：

source addr	desti addr	mac payload(01 addr ID)
2	4	01 02 05
4	8	01 02 05
3	4	01 03 02
4	8	01 03 02
5	4	01 05 03
4	8	01 05 03
4	8	01 04 06
7	8	01 07 04
6	8	01 06 07
1	8	01 01 01

可以知道這次 demo 建出的 routing 為：G → H; D → H; A → H; B,C,E → F → H。畫成圖可以變成：



有一點令我們困惑，是否每個裝置最遠只能聽到 **two-hop**，因為從擷取的封包來看，顯示 A 的 **relay node** 是 H，代表 A 可以聽到 H 廣播的 **flag message**，我們認為其實 A 還是有機會聽到 H 的 **message**，只是能量很低，故可能幾百個封包有機會能收到一個正確的 **message**。