CN HW3

B05611047 鍾宜樺

1. How to execute your program (3%)

A. 準備 (如下面截圖)：

  a. 將檔案全部弄到一個資料夾裡面 (server.cpp, agent.c, client.cpp, Makefile, tmp.mpg)

  b. 輸入 Command line: make

  c. 接著先跑一次 server 執行檔，讓 server 的資料夾: server_f 出現

    ./server local local 8887 8888 8889

  d. 把要播放的影片放到 server_f 的資料夾裡面

Step a.
Step b.

```
cnlab@cnlab-VirtualBox:~/cnhw3_final$ ls
agent.c  client.cpp  Makefile  server.cpp  shw_player.cpp  tmp.mpg
cnlab@cnlab-VirtualBox:~/cnhw3_final$ make
g++ -std=c++11 server.cpp -o server `pkg-config --cflags --libs opencv` -pthread
server.cpp: In function 'int main(int, char**)':
server.cpp:406:29: warning: ISO C++ forbids converting a string constant to 'char*' [-Wwrite-strings]
         setIP(ip[1], "local"); //agent IP
                     ^
g++ client.cpp -o client  `pkg-config --cflags --libs opencv` -pthread
client.cpp: In function 'int main(int, char**)':
client.cpp:312:29: warning: deprecated conversion from string constant to 'char*' [-Wwrite-strings]
         setIP(ip[1], "local"); //agent IP
                     ^
g++ agent.c -o agent `pkg-config --cflags --libs opencv` -pthread
agent.c: In function 'int main(int, char**)':
agent.c:72:29: warning: deprecated conversion from string constant to 'char*' [-Wwrite-strings]
         setIP(ip[1], "local");
                     ^
make: *** No rule to make target 'shw', needed by 'all'.  Stop.
cnlab@cnlab-VirtualBox:~/cnhw3_final$ ./server local local 8887 8888 8889
playing: tmp.mpg
cnlab@cnlab-VirtualBox:~/cnhw3_final$ ls
agent  agent.c  client  client.cpp  Makefile  server  server.cpp  server_f  shw_player.cpp  tmp.mpg
cnlab@cnlab-VirtualBox:~/cnhw3_final$ mv tmp.mpg server_f/
cnlab@cnlab-VirtualBox:~/cnhw3_final$ ls
agent  agent.c  client  client.cpp  Makefile  server  server.cpp  server_f  shw_player.cpp
cnlab@cnlab-VirtualBox:~/cnhw3_final$
```

Step c.
Step d.

B. 播放影片：

  a. 先下 agent 的 command line，如下圖：

    ./agent local local 8887 8888 8889 0.1   //(0.1 是 lost rate)

```
cnlab@cnlab-VirtualBox:~/cnhw3_final$ ./agent local local 8887 8888 8889 0.1
可以開始測囉^Q^
sender info: ip = 127.0.0.1 port = 8887 and receiver info: ip = 127.0.0.1 port =
 8889
agent info: ip = 127.0.0.1 port = 8888
get     data    #1
fwd     data    #1,     loss rate = 0.0000
```

  b. 再下 client 的 command line (這裡有**兩步驟要輸入！**)，如下圖：

    b-1 ./client local local 8887 8888 8889

    b-2 play tmp.mpg   //(tmp.mpg 是 filename)，截圖中是 plplay 是突然當掉了

```
cnlab@cnlab-VirtualBox:~/cnhw3_final$ ./client local local 8887 8888 8889
Giving instruction: play <filename>
plplay tmp.mpg
recv    data    #1
send    ack     #1
recv    data    #2
send    ack     #2
```

  c. 最後下 server 的 command line，如下圖：

    ./server local local 8887 8888 8889

```
cnlab@cnlab-VirtualBox:~/cnhw3_final$ ./server local local 8887 8888 8889
playing: tmp.mpg
Finded out the file, file name: tmp.mpg
send    data    #1,     winSize = 1
recv    ack     #1
```

2. Explain your program structure (3% * 3)

(including 3 flow charts for sender, agent and receiver)

如下三張 flow charts，順序為：server -> agent -> receiver

| Agent |

(Start)
↓

① 讀進 sender IP, receiver IP... (port ...)
↓

② create UDP socket for agent: agent socket = socket(PF_INET, SOCK_DGRAM, 0)
↓

③
3-1 Configure setting in sender struct.
3-2 Configure setting in agent struct.
　　3-2-a) binding: agent socket binding: Bind a socket to the
　　　　　agent address with a bind. 用來接 receiver, sender 的 pkt.
3-3 Configure setting in receiver struct
↓

④ 接 receiver 送過來的 file name 後傳給 sender.
↓

⑤ 開始接來自 sender ‖ receiver 的 pkt. (playing video 的)
　　... 由 ip 和 port from judge 是來自 sender ‖ receiver.

直到接到
fin

　　5-a 若是來自 sender 的 pkt, 則 random drop 沒有 drop 則 "send to" receiver
　　　　(recvfrom)　　　　　　　　　　　　　並 count loss rate.
　　5-b 來自 receiver 的 ack 則不可 drop. 傳給 sender.
　　　　(recvfrom)　　　　　　　　　　　　(sendto)
↓接到 fin.

⑥ 一直到接到 sender 的 fin segment 後傳到 receiver 那.
↓

⑦ 直到收到 receiver 的 fin ack segment 後將 fin ack
傳給 sender 後
↓

⑧ 結束程式.

Client.

Start

① 創建 client 的 file : client_f.

② 讀進 sender IP 、 recv IP, sender port .....

③. create UDP socket for receiver : receiver socket= socket(PF_INET, SOCK_DGRAM,0)

④ Configure setting in sender struct.

④-1 Configure setting in agent struct.

④-2 Configure setting in receiver struct.

→ 這部份的詳細敘述
請看 sender's flow chart 的
④-1 ～ ④-3

④-3-a binding : Receiver socket binding : Bind a socket to the receiver address with a bind., 用來接 agent 的 pkt.

⑤ Giving instruction : eg : play tmp.mpg.

⑥ 用 sender 將 filename 傳給 agent, 再傳給 receiver.

⑦ 開始 wait for pkt from agent 等待接 video's pkt, 用 "recvfrom" & imgsz

⑧ 接到 pkt 後要 "send ack" (for the seqNumber 是 pkt 的 ack) → 照順序排
→ 用 sendto.
→ 將 pkt data 存入播放 video 的 imshow 中
1→3→4
則 3.4 丟掉

直到 video
播完前
(imgsz=0前

⑨-1 如果有 pkt loss 的話要傳上一個已收到的 pkt's seqNumber 的 ack 回去,
讓 sender 知道 receiver 端現接到 哪裡.

⑨-2 如果 w/o pkt loss, 則 judge 若 buffer 已滿 (32 個 segments) 則要進行 flush
的動作 清空 buffer.

↓影片播完. get imgsz=0.

⑩ 等待 fin segment 後, "sendto" 傳 fin ack, 斬 agent 傳回給 server.

⑪ 程式 結束