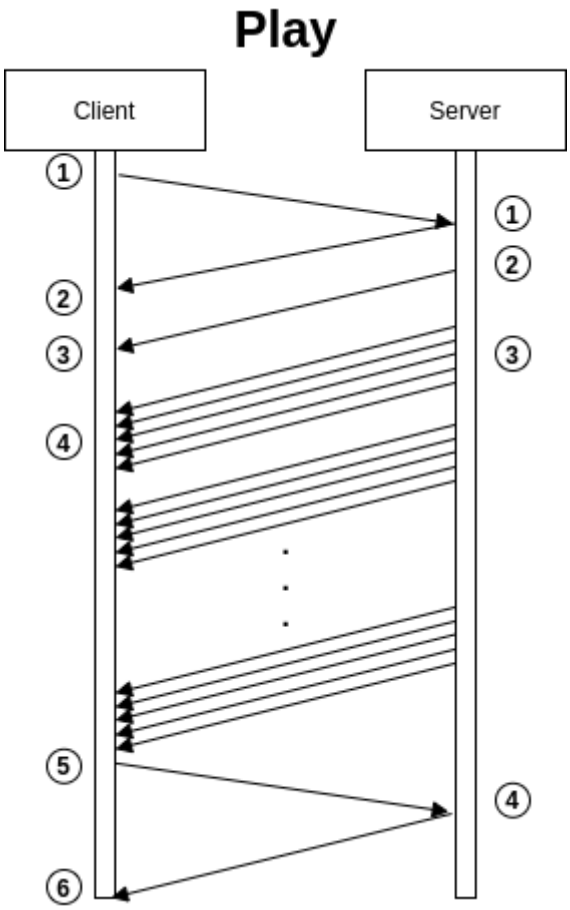
Computer Network HW2 Report

b07902408 資工三 李宥霆

1



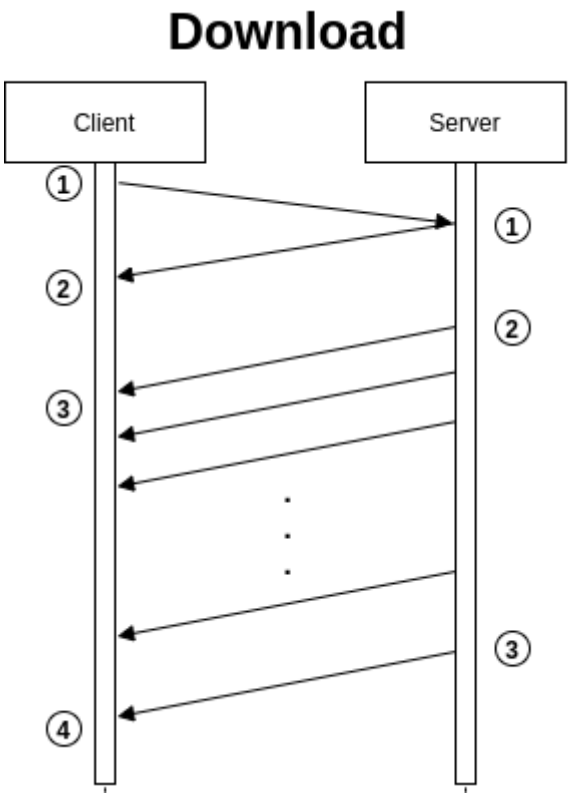
Client視角：

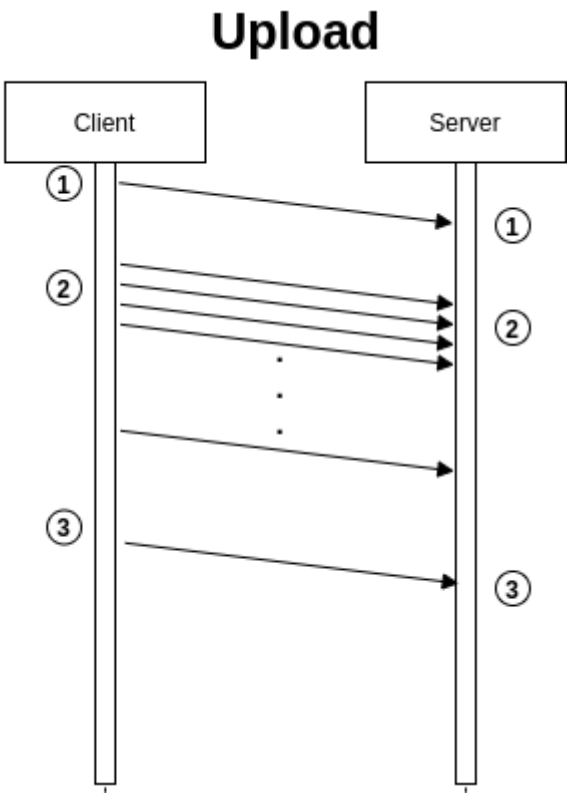
步驟	行為
1	收到指令，檢查指令沒有問題後送出PLAY_REQ，其中包含檔案名稱
2	如果Server端沒有該檔案或檔案格式不符，收到FILE_NOT_FOUND， 代表沒有檔案，回到等待指令的狀態；否則收到NONE，代表接受請求，繼續步驟3
3	收到WIDTH_HEIGHT，內含初始Mat物件所需要的寬跟高參數值
3.5	這時client會新增一個thread，負責收封包並放進buffer， buffer用C++的vector實做，在放進buffer前會上lock，防止Race condition
4	原本的thread從buffer拿多個封包，代表一個frame， 其中第一個封包為IMG_SIZE，代表frame大小，其餘封包為FRAME， 內含frame內容，重複步驟4直到收完所有frame
5	**(非必要)** client收到ESC，於是傳送ESC_REQ給server
6	client收到FINISH，代表串流結束，回到等待指令的狀態

Server視角：

步驟	行為
1	收到PLAY_REQ，其中包含檔案名稱，於是尋找該檔案是否存在。 是則回傳NONE，繼續步驟2；否則回傳FILE_NOT_FOUND，回到等待狀態
2	傳送WIDTH_HEIGHT，內含初始Mat物件所需要的寬跟高參數值
3	一次傳送一組封包，為一個frame，其中第一個封包為IMG_SIZE， 代表frame大小，其餘封包為FRAME，內含frame內容
4	收到ESC_REQ或送完frame，傳送FINISH，代表串流結束，回到等待狀態

2





Download Client視角：

步驟	行為
1	收到指令，檢查指令沒有問題後送出DOWN_REQ，其中包含檔案名稱
2	如果Server端沒有該檔案或檔案格式不符，收到FILE_NOT_FOUND， 代表沒有檔案，回到等待指令的狀態；否則收到NONE，代表接受請求，繼續步驟3
3	連續收到DATA,，內含被切塊的檔案
4	client收到FINISH，代表傳送完成，回到等待指令的狀態

Download Server視角：

步驟	行為
1	收到DOWN_REQ，其中包含檔案名稱，於是尋找該檔案是否存在。 是則回傳NONE，繼續步驟2；否則回傳FILE_NOT_FOUND，回到等待狀態
2	把檔案切塊，並一塊一塊傳送給client
3	server送出FINISH，代表傳送完成，回到等待狀態

Upload Client視角：

步驟	行為
1	收到指令，檢查指令沒有問題後送出UP_REQ，其中包含檔案名稱
2	把檔案切塊，並一塊一塊傳送給server
3	client送出FINISH，代表傳送完成，回到等待指令的狀態

Upload Server視角：

步驟	行為
1	收到UP_REQ，其中包含檔案名稱
2	連續收到DATA,，內含被切塊的檔案
3	server收到FINISH，代表傳送完成，回到等待狀態

3

SIGPIPE: 當socket的其中一方關閉了socket，另一方仍舊繼續往socket寫進資料時，系統就會傳送一個SIGPIPE給process，要他不要再繼續寫了。在我的程式中，如果client想要斷開連接，他會先傳送一個EXIT的封包給server，接著server端就會關閉socket並釋放資源，此時client端就可以自由關閉，而server端也不會因為繼續寫資料而收到SIGPIPE。

4

blocking I/O: 是指當I/O還沒做完時，process會處於被function call block住的狀態，等到I/O做完之後，function call才會return。
synchronized I/O: 是指處理I/O的thread是main thread，也就是不另外開一個thread來做I/O的處理。
舉個例子來說：
recv()這個function call有blocking的模式與non-block的模式，一般來說，他是blocking的，也就是要等到socket的buffer裡面有資料的時候才會把資料搬回來並return；而在參數中增加MSG_DONTWAIT這個參數時，模式變成non-blocking，呼叫recv()時不管buffer裡面有沒有資料都會馬上return，如果buffer裡面剛好有資料時會順便把裡面的資料搬回來。如果是在main thread裡面直接呼叫recv()這個function時，做的就是synchronized的I/O，除非我們自己新增一個thread，並在該thread裡面呼叫recv()這個function，這時做的就會是asynchronized的I/O。
總的來說，blocking跟non-blocking通常指的是function call本身的行為模式；而synchronized與asynchronized，則是指處理這個function call時是不是平行處理，可不可以在做I/O的同時處理其他事情。