Project IV: SOCKS4 Server

Prof. I-Chen Wu

Network Programming

Abstract

- SOCKS 4 Protocol
- SOCKS 4a Protocol (extension)
- In this project, you are asked to implement the **SOCKS 4 firewall protocol** in the application layer of the OSI model.

- SOCKS is similar to a proxy (i.e. intermediary-program) that acts as both server and client for the purpose of making request on behalf of other clients. Because the SOCKS protocol is independent of application protocols, it can be used for many different services: telnet, ftp, www, etc.

- There are two types of the SOCKS operations (i.e. **CONNECT** and **BIND**). You have to implement both of them.

Requirements

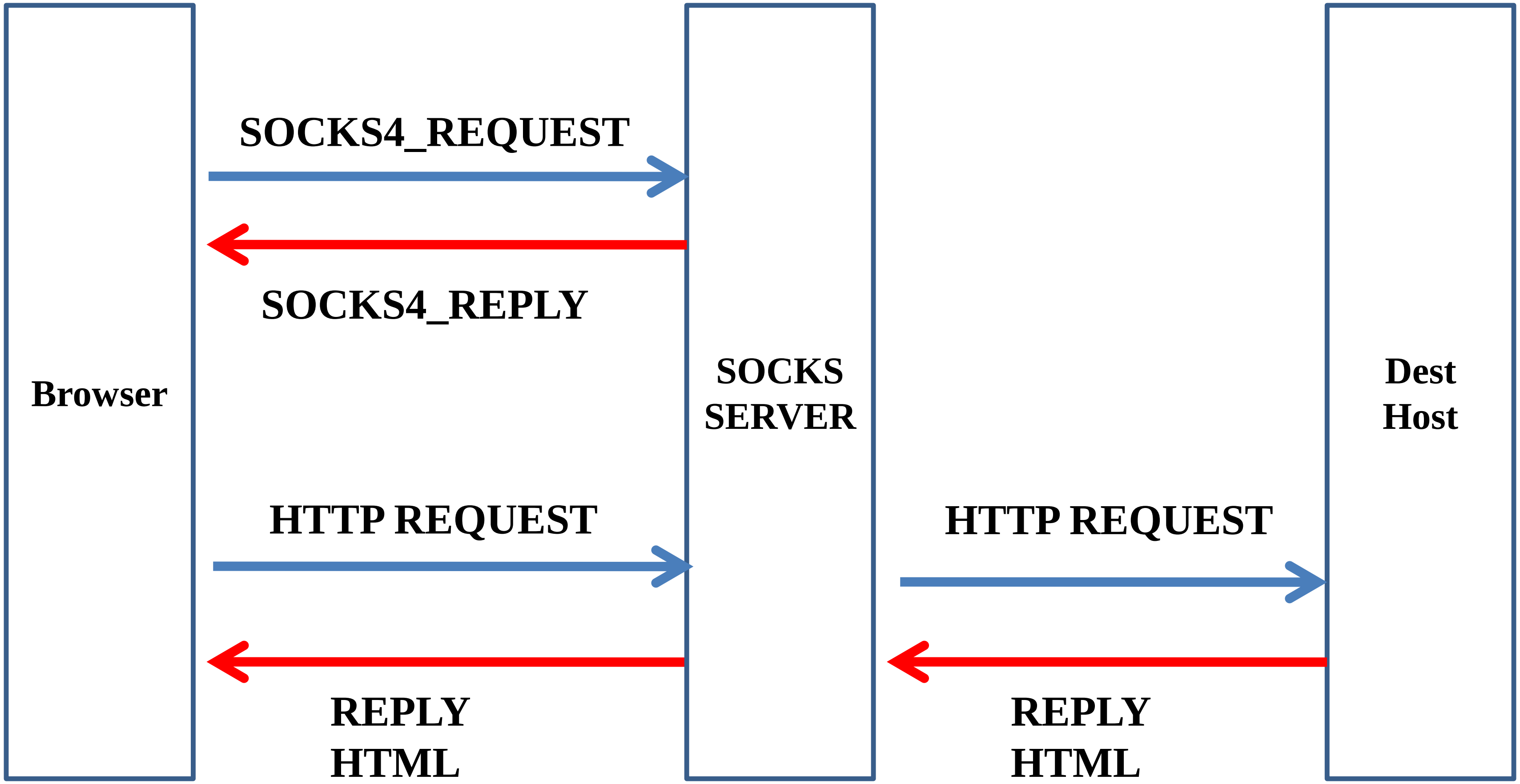
- Code
  - Part I: Socks4 Server Connect Mode
  - Part II: Socks4 Server Bind Mode
  - Part III: CGI Proxy
- Others
  - Name your cgi as hw4.cgi
  - Wrap your code into .zip (Do not upload test cases and git files)
  - Use the concurrent, connection-oriented paradigm.

Schedule

- Deadline:
  - 2017/12/31 23:59 (Sunday)
- Demo:
  - 2018/01/02 10:00~17:00 (Tuesday)

Part I: Socks4 Server Connect Mode

Web Browser(Connect mode)



Browser

SOCKS4_REQUEST

SOCKS4_REPLY

HTTP REQUEST

REPLY
HTML

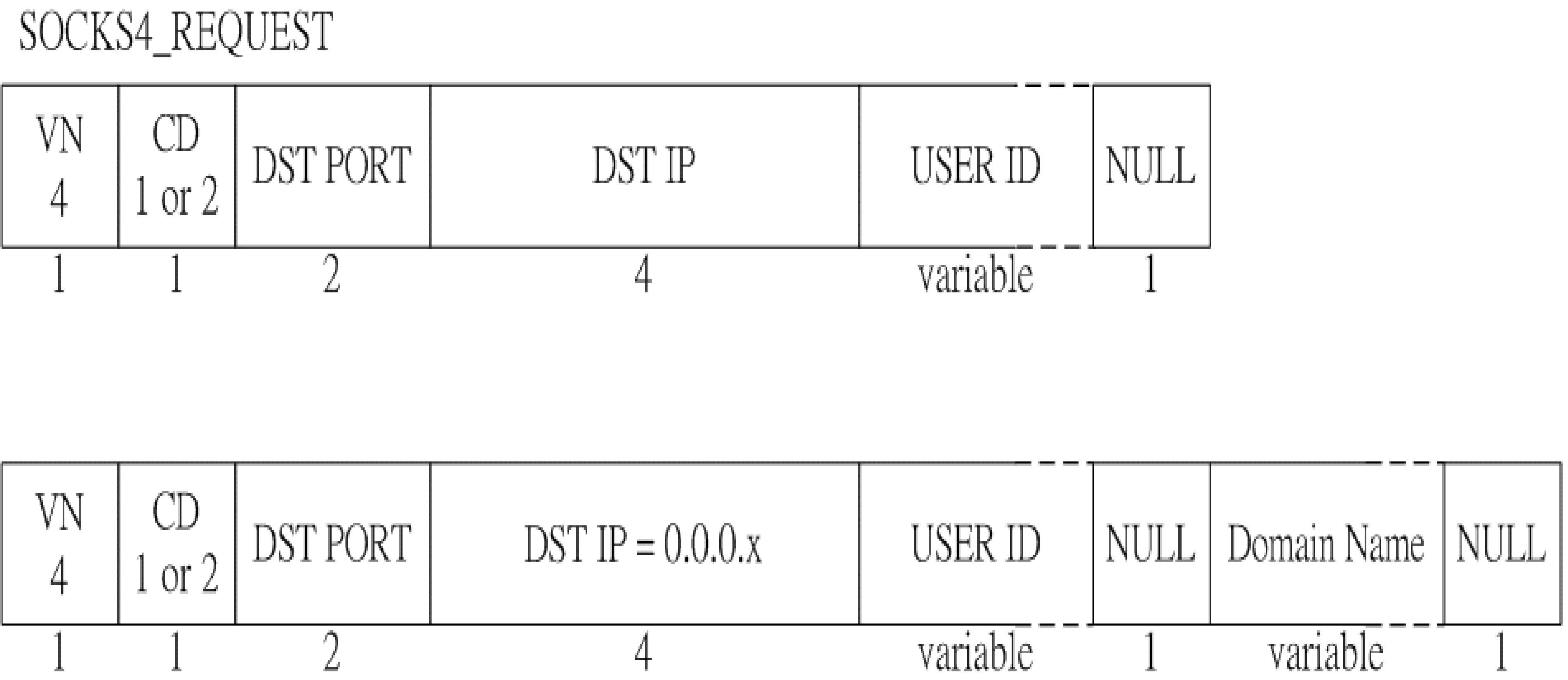
SOCKS
SERVER

HTTP REQUEST

REPLY
HTML

Dest
Host

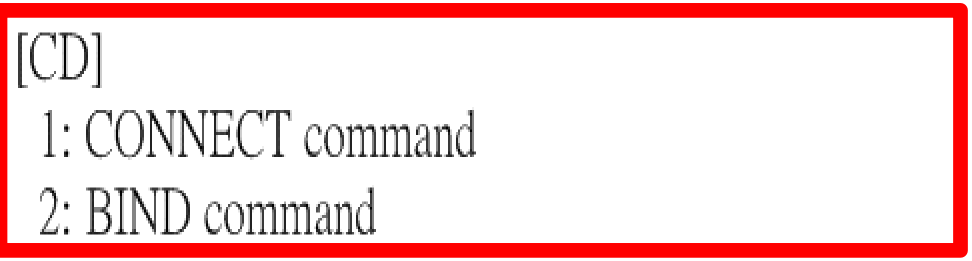
SOCKS4_REQUEST



SOCKS4_REQUEST

VN 4	CD 1 or 2	DST PORT	DST IP	USER ID	NULL
1	1	2	4	variable	1

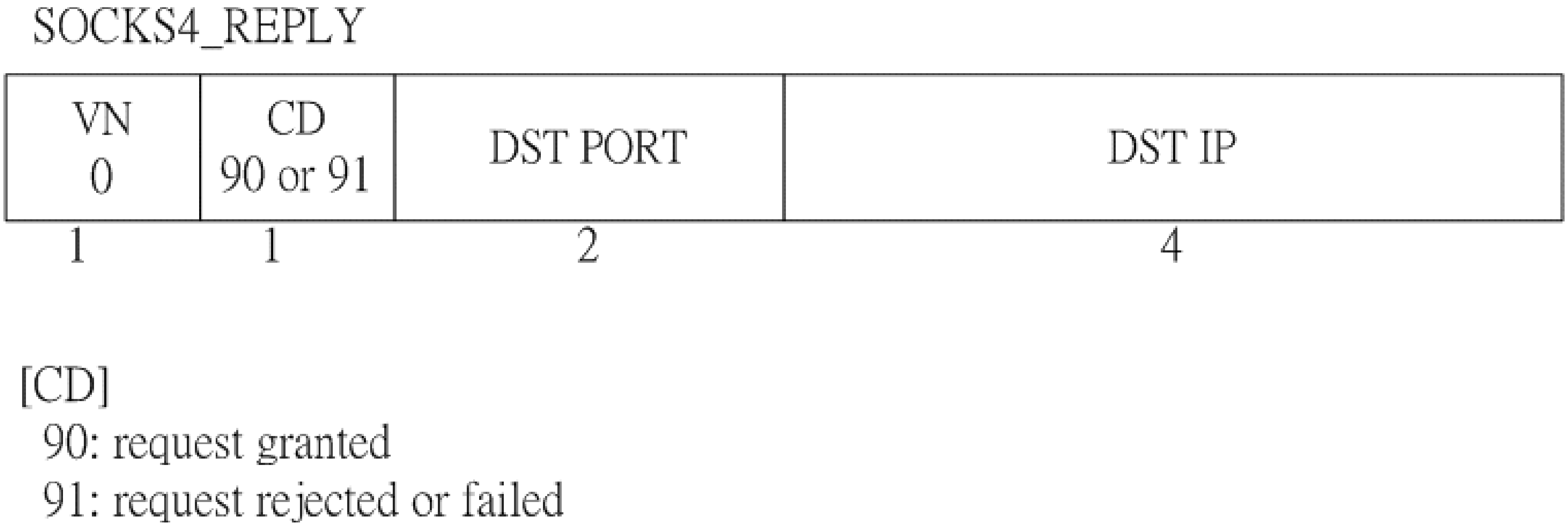
VN 4	CD 1 or 2	DST PORT	DST IP = 0.0.0.x	USER ID	NULL	Domain Name	NULL
1	1	2	4	variable	1	variable	1



[CD]
1: CONNECT command
2: BIND command

[DST IP]
Connect mode: is the DST IP in SOCKS4_REQUEST
Bind mode: 0

SOCKS_REPLY



SOCKS4_REPLY

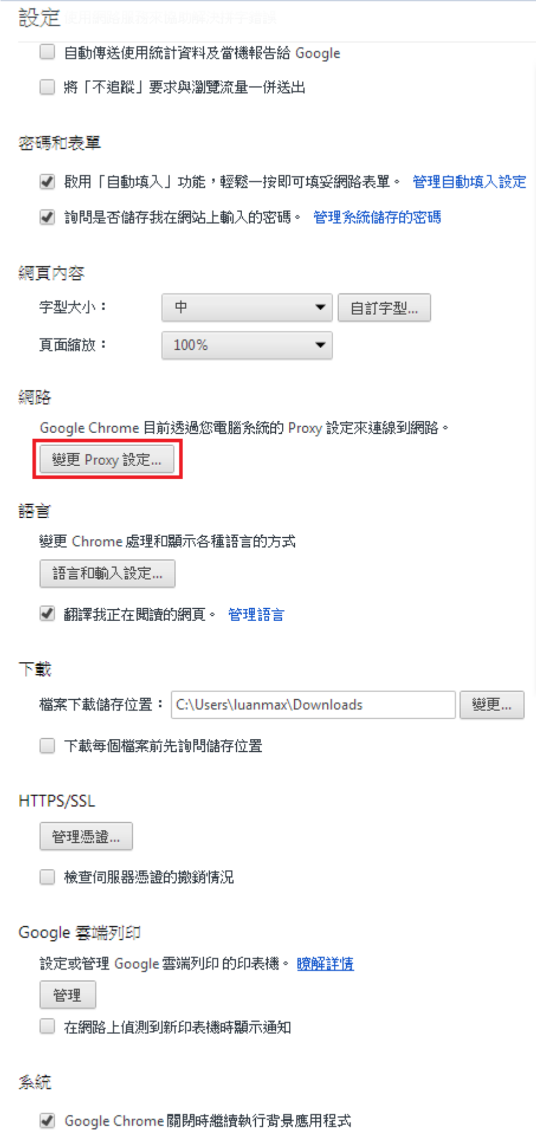
VN 0	CD 90 or 91	DST PORT	DST IP
1	1	2	4

[CD]

90: request granted

91: request rejected or failed

Browser Setting



設定　搜尋設定

□ 自動傳送使用統計資料及當機報告給 Google

□ 將「不追蹤」要求與瀏覽流量一併送出

密碼和表單

☑ 啟用「自動填入」功能，輕鬆一按即可填妥網路表單。 管理自動填入設定

☑ 詢問是否儲存我在網站上輸入的密碼。 管理系統儲存的密碼

網頁內容

字型大小： 中　自訂字型...

頁面縮放： 100%

網路

Google Chrome 目前透過您電腦系統的 Proxy 設定來連線到網路。

變更 Proxy 設定...

語言

變更 Chrome 處理和顯示各種語言的方式

語言和輸入設定...

☑ 翻譯我正在閱讀的網頁。 管理語言

下載

檔案下載儲存位置： C:\Users\luanmax\Downloads　變更...

□ 下載每個檔案前先詢問儲存位置

HTTPS/SSL

管理憑證...

□ 檢查伺服器憑證的撤銷情況

Google 雲端列印

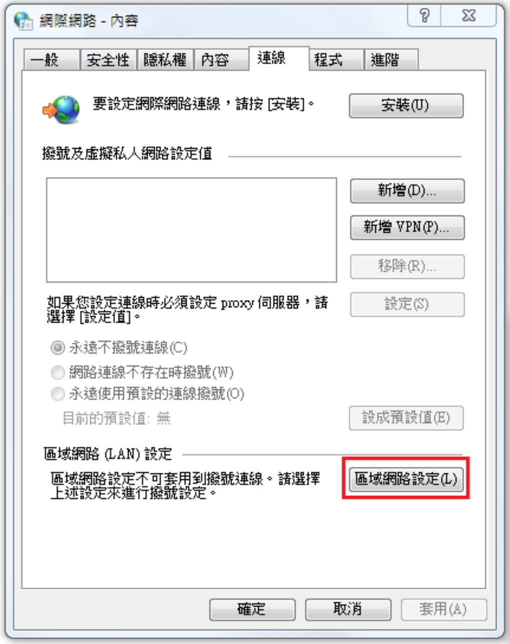
設定或管理 Google 雲端列印 的印表機。 瞭解詳情

管理

□ 在網路上偵測到新印表機時顯示通知

系統

☑ Google Chrome 關閉時繼續執行背景應用程式



網際網路 - 內容

一般 | 安全性 | 隱私權 | 內容 | 連線 | 程式 | 進階

要設定網際網路連線，請按 [安裝]。　安裝(U)

撥號及虛擬私人網路設定值

新增(D)...　新增 VPN(P)...　移除(R)...

如果您設定連線時必須設定 proxy 伺服器，請選擇 [設定值]。　設定(S)

◉ 永遠不撥號連線(C)

○ 網路連線不存在時撥號(W)

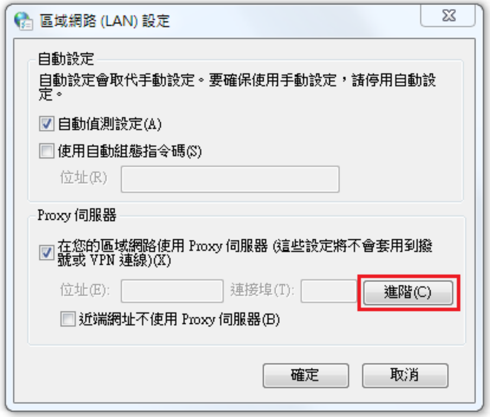
○ 永遠使用預設的連線撥號(O)

目前的預設值: 無　設成預設值(E)

區域網路 (LAN) 設定

區域網路設定不可套用到撥號連線。請選擇上述設定來進行撥號設定。　區域網路設定(L)

確定　取消　套用(A)



區域網路 (LAN) 設定

自動設定

自動設定會取代手動設定。要確保使用手動設定，請停用自動設定。

☑ 自動偵測設定(A)

□ 使用自動組態指令碼(S)

位址(R)

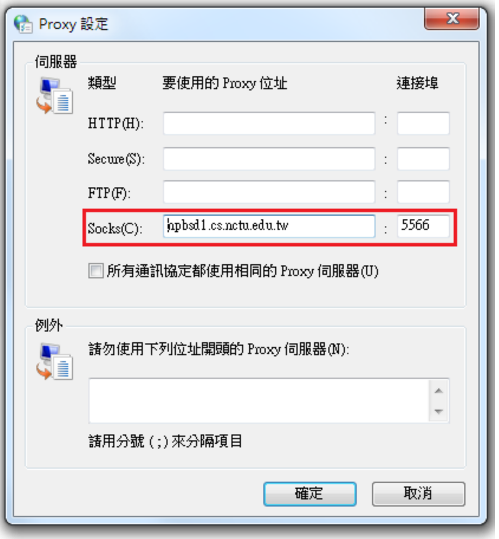
Proxy 伺服器

☑ 在您的區域網路使用 Proxy 伺服器 (這些設定將不會套用到撥號或 VPN 連線)(X)

位址(E):　連接埠(T):　進階(C)

□ 近端網址不使用 Proxy 伺服器(B)

確定　取消



Proxy 設定

伺服器

類型	要使用的 Proxy 位址	連接埠
HTTP(H):		
Secure(S):		
FTP(F):		
Socks(C):	bpbsd1.cs.nctu.edu.tw	5566

□ 所有通訊協定都使用相同的 Proxy 伺服器(U)

例外

請勿使用下列位址開頭的 Proxy 伺服器(N):

請用分號 (;) 來分隔項目

確定　取消

Part I points

- **[SOCKS Server : CONNECT]** (15 points)
  - 1. Open your browser and connect to any webpage
  - 2. Turn on and set your socks server, then

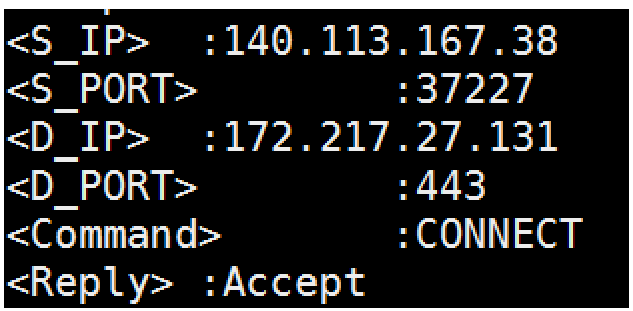
    (1)5%: be able to connect any webpages on google search

    (2)5%: turn off your socks server, connection to the same page will be failed.

    (3)5%: turn on your socks server, the connection should be built again.

Part I points

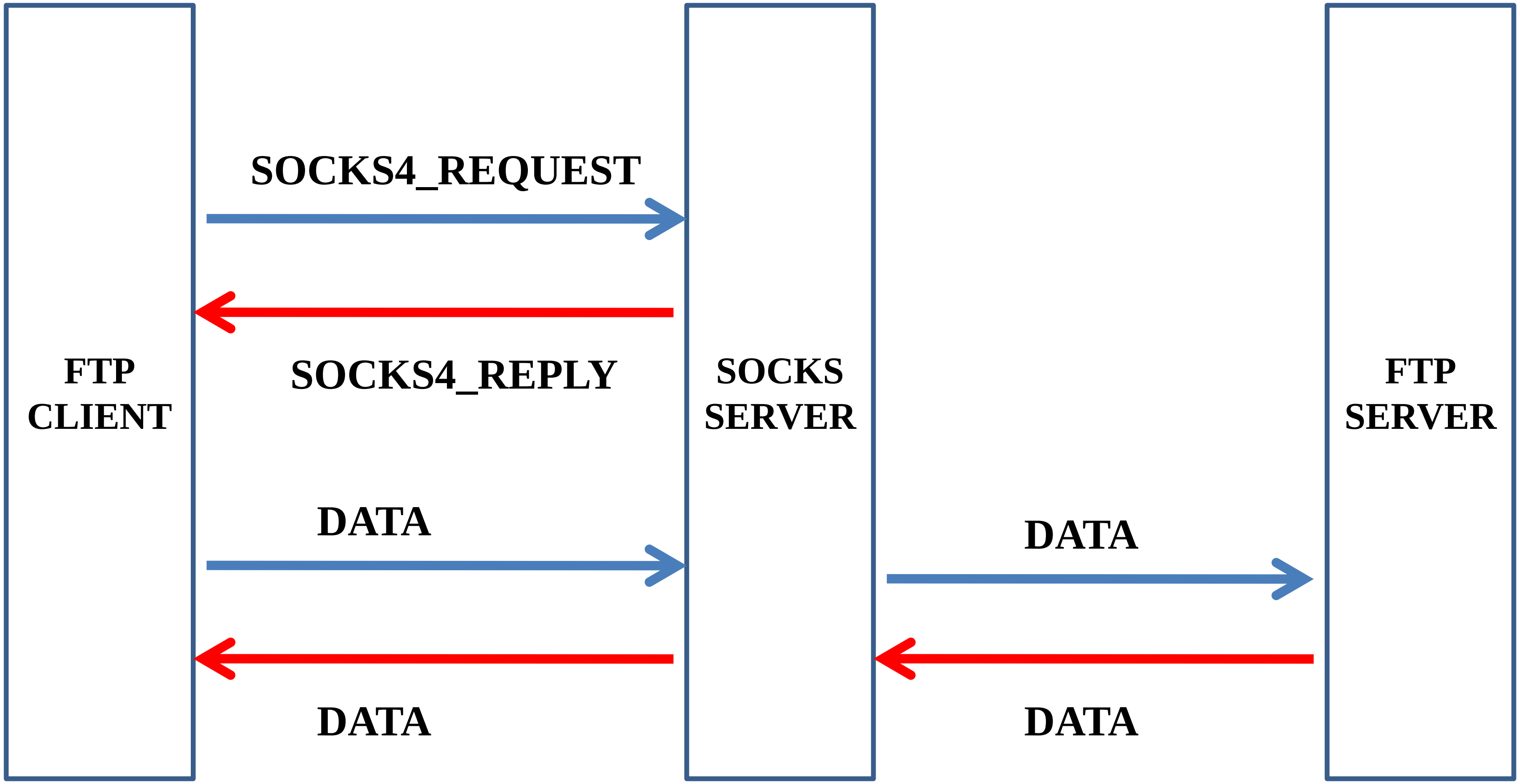
- **[SOCKS Server : Messages]** (5 points)
  - Your socks server need to show messages below :
    - <S_IP> : source ip
    - <S_PORT> : source port
    - <D_IP> : destination ip
    - <D_PORT> : destination port
    - <Command> : CONNECT or BIND
    - <Reply> : Accept or Reject
    - <Content> : Redirect **partial** socket data



```
<S_IP>  :140.113.167.38
<S_PORT>        :37227
<D_IP>  :172.217.27.131
<D_PORT>        :443
<Command>       :CONNECT
<Reply> :Accept
```

Part II: Socks4 Server Bind Mode

FTP Transfer(Bind mode)



FTP server/client

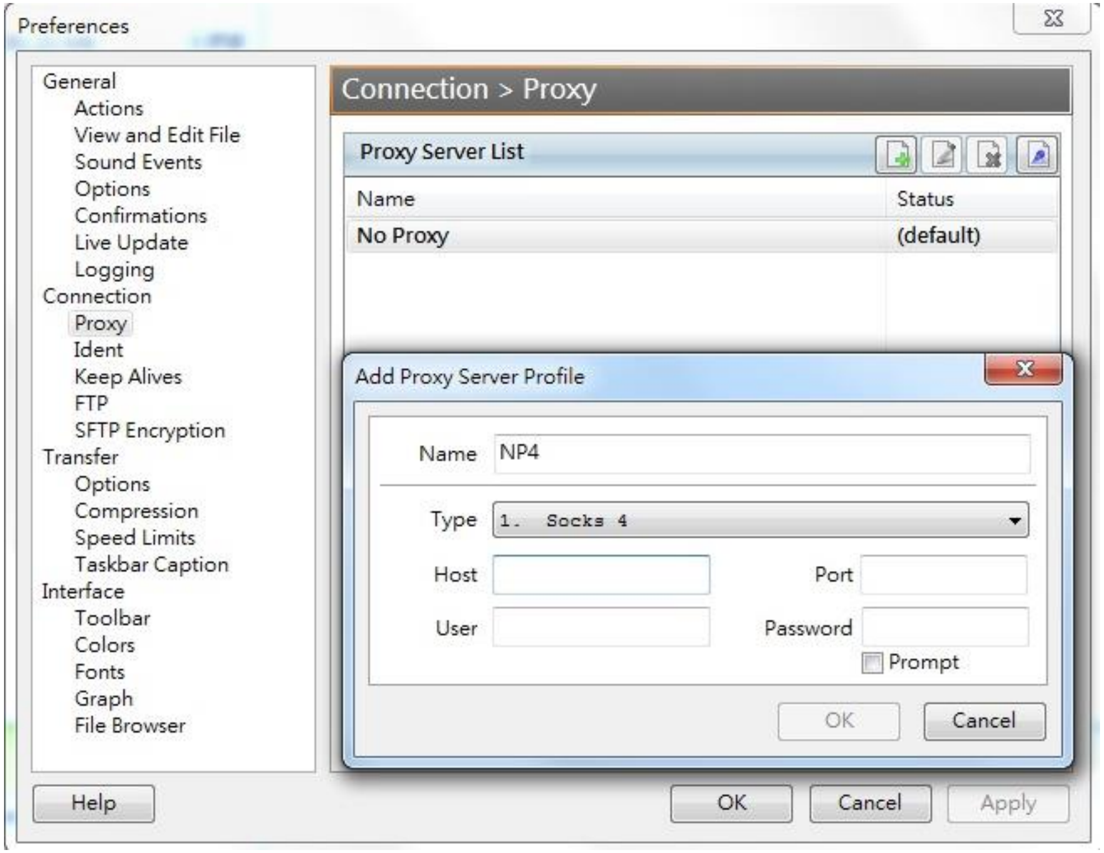
- FTP Server
  - You can apply free 5GB space on http://5gbfree.com/
    - Server: **ftp.[Username].5gbfree.com:21**
    - Account: [Username]
  - Or build a FTP server on your own computer
    - Reference: http://goo.gl/UjrFwy

- FTP client
  - Use FlashFXP (http://www.flashfxp.com/)
  - FileZilla is not recommended

- Steps
  - Download and set FlashFXP
  - Connect to FTP server
  - Upload/Download files

FlashFXP

- Options → Preferences→Connection→Proxy
  - Add entry
    - Socks4
    - Host/Port



Name	Status
No Proxy	(default)

Name NP4

Type 1. Socks 4

Host

Port

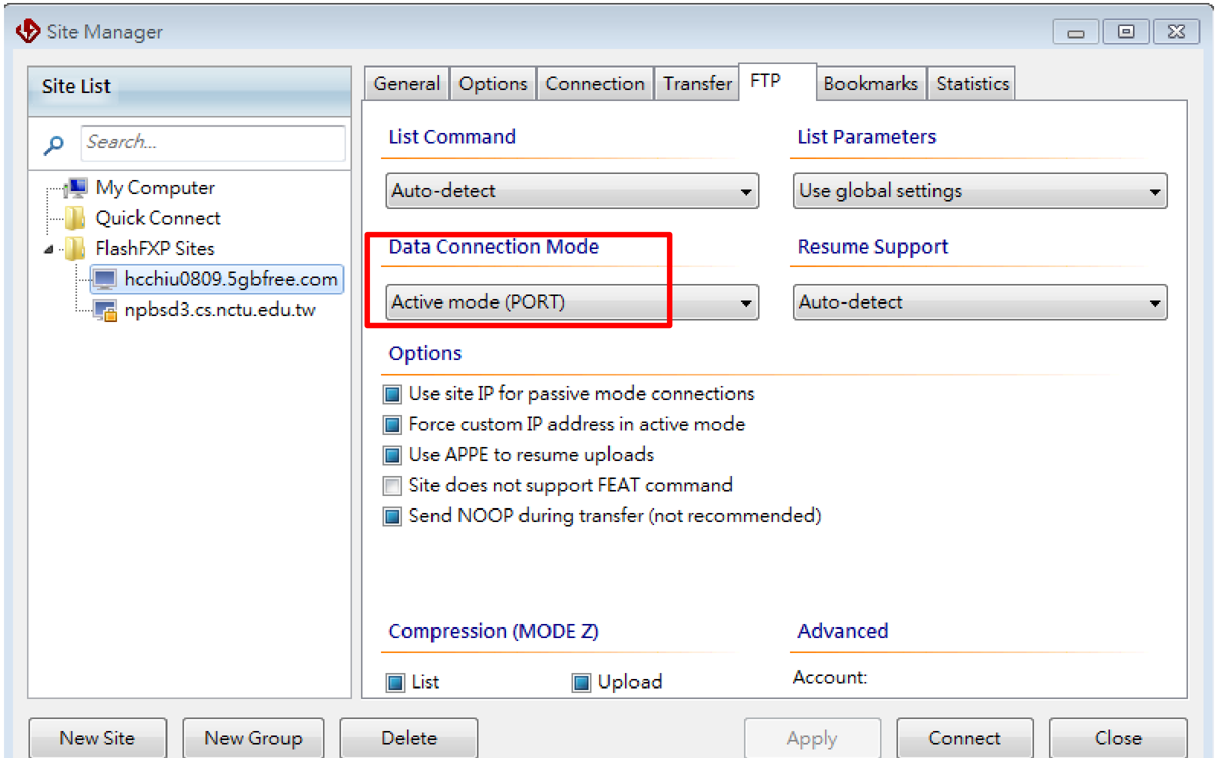
User

Password

□ Prompt

FlashFXP – set FTP server

- Sites → Site Manager→ New Site → FTP→ Data Connection Mode
  - Change to Active Mode(PORT)



List Command: Auto-detect

List Parameters: Use global settings

Data Connection Mode: Active mode (PORT)

Resume Support: Auto-detect

Options

☑ Use site IP for passive mode connections
☑ Force custom IP address in active mode
☑ Use APPE to resume uploads
□ Site does not support FEAT command
☑ Send NOOP during transfer (not recommended)

Compression (MODE Z)

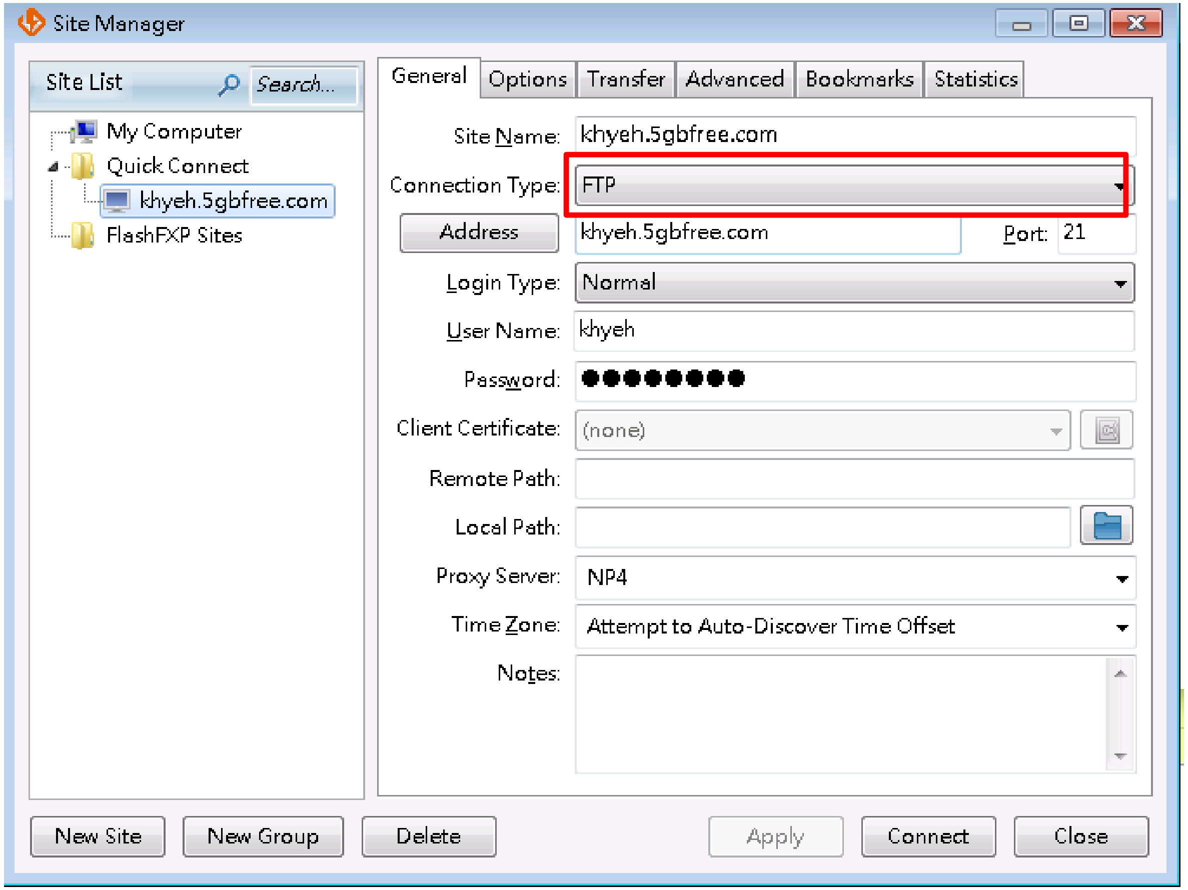
☑ List ☑ Upload

Advanced

Account:

FlashFXP – set FTP server

- Sites → Site Manager→General → Apply
  - Connect



Site Name: khyeh.5gbfree.com

Connection Type: FTP

Address: khyeh.5gbfree.com Port: 21

Login Type: Normal

User Name: khyeh

Password: ●●●●●●●●●

Client Certificate: (none)

Remote Path:

Local Path:

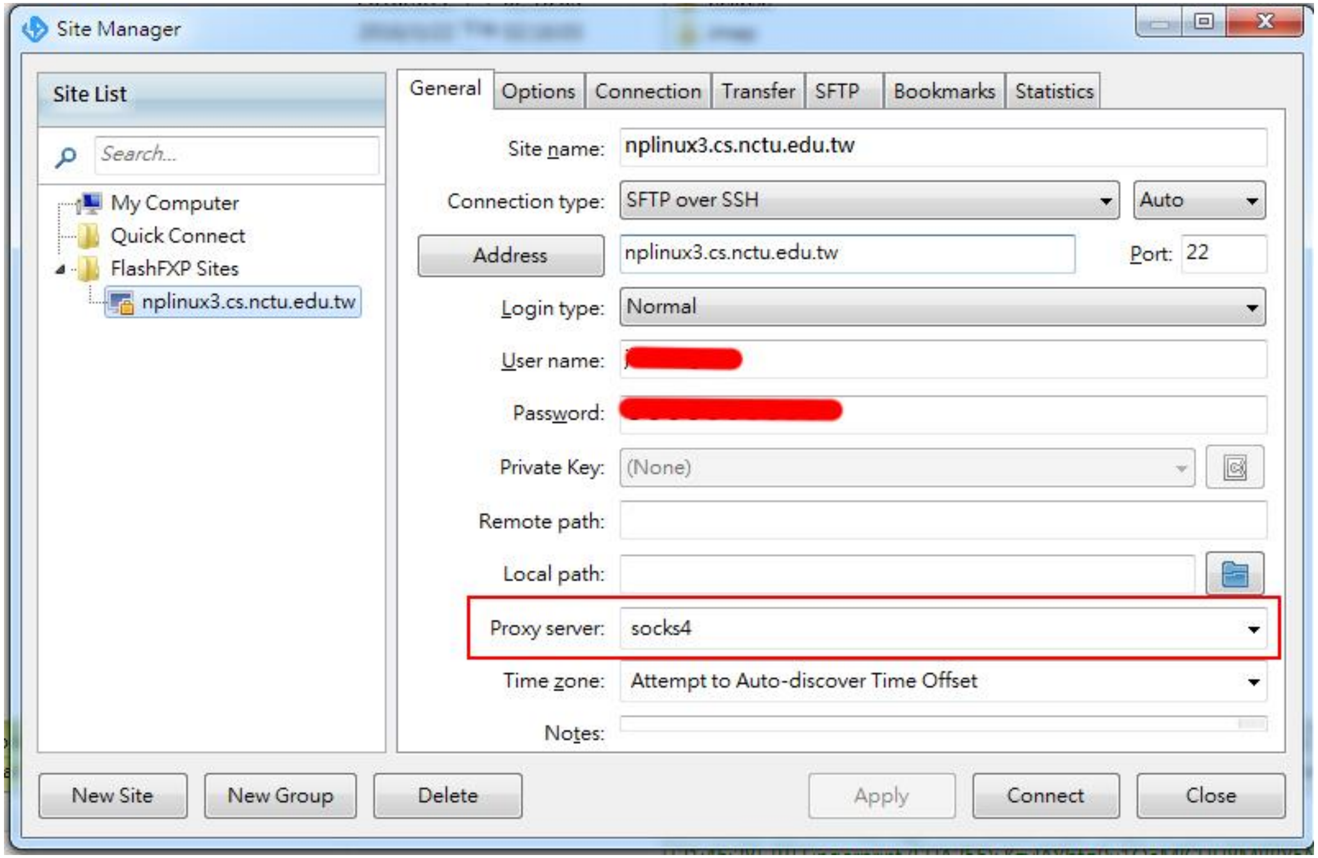
Proxy Server: NP4

Time Zone: Attempt to Auto-Discover Time Offset

Notes:

FlashFXP

- Sites → Site Manager →General → Proxy Server



Site name: nplinux3.cs.nctu.edu.tw

Connection type: SFTP over SSH | Auto

Address: nplinux3.cs.nctu.edu.tw | Port: 22

Login type: Normal

User name: [redacted]

Password: [redacted]

Private Key: (None)

Remote path:

Local path:

Proxy server: socks4

Time zone: Attempt to Auto-discover Time Offset

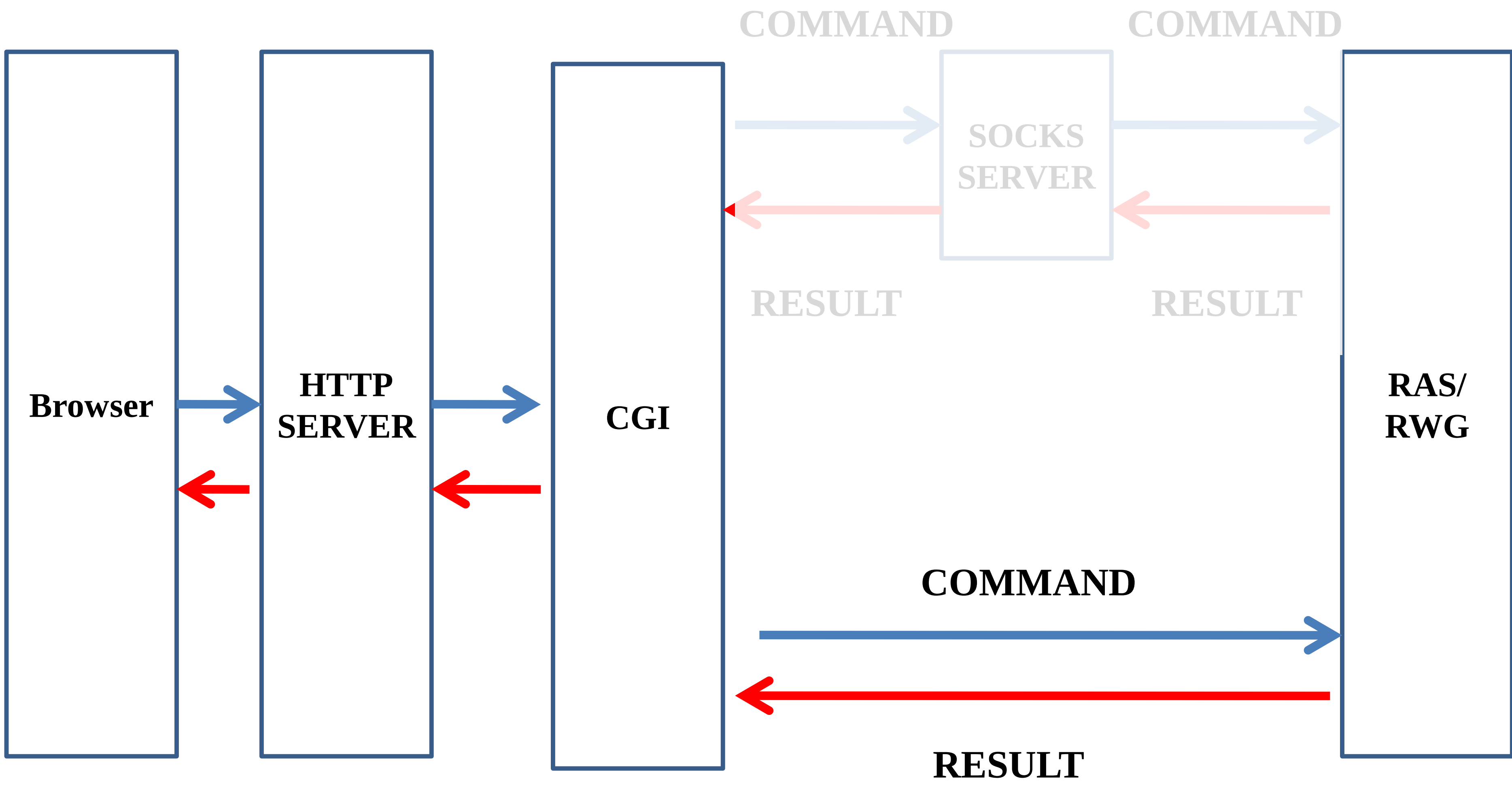
Notes:

Part II points

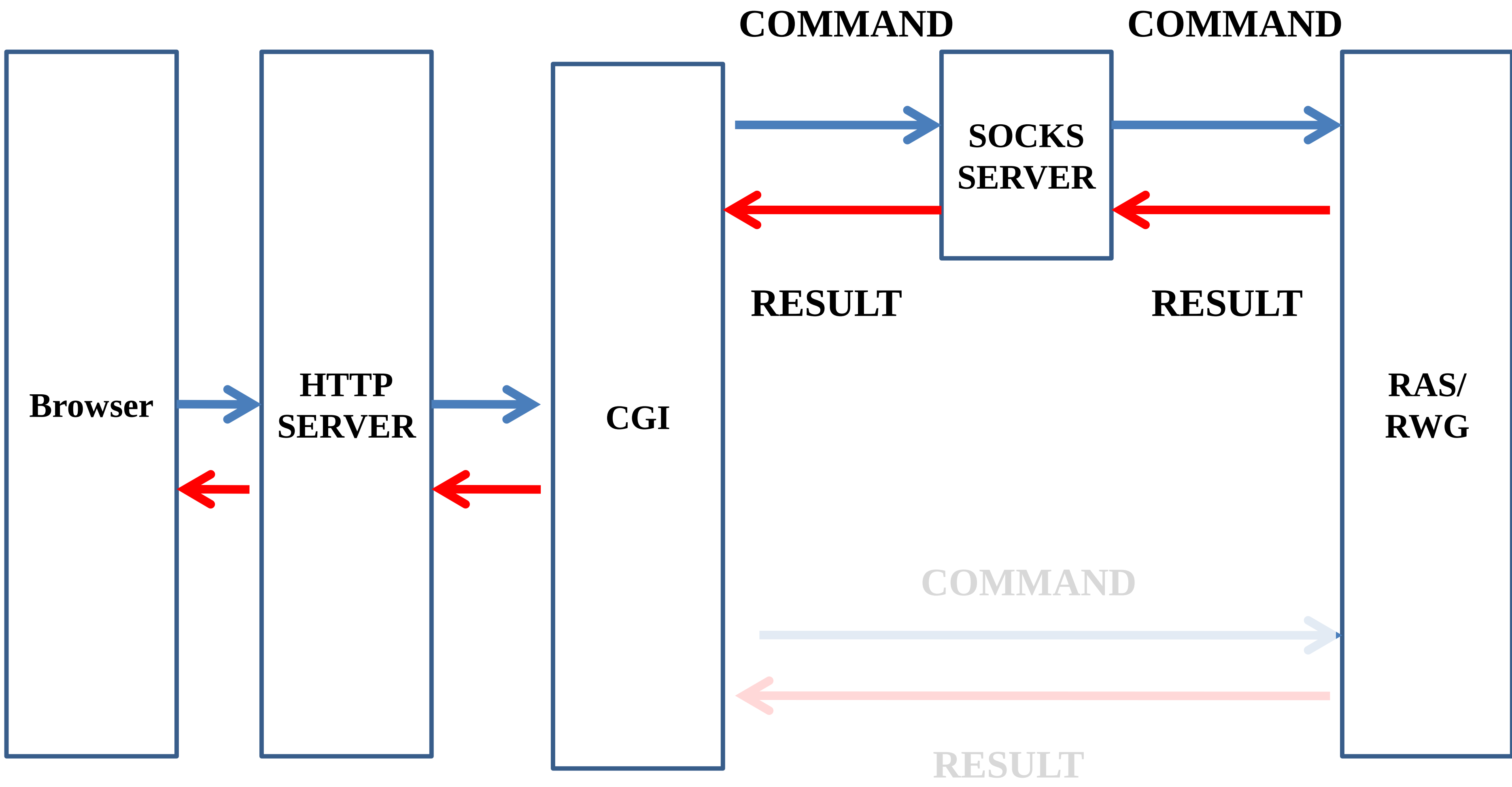
- **[SOCKS Server : BIND]** (15 points)
  - – Open FlashFXP and set your socks server
  - – Connect to ftp server, and upload/download file > 1GB.
    - E.g. Ubuntu 16.04 iso (download link)
    - Upload/Download file and check whether the size remains the same and be able to open (5 points each)
  - – Check whether SOSKS server's output has used BIND mode(5 points)

Part III: CGI Proxy

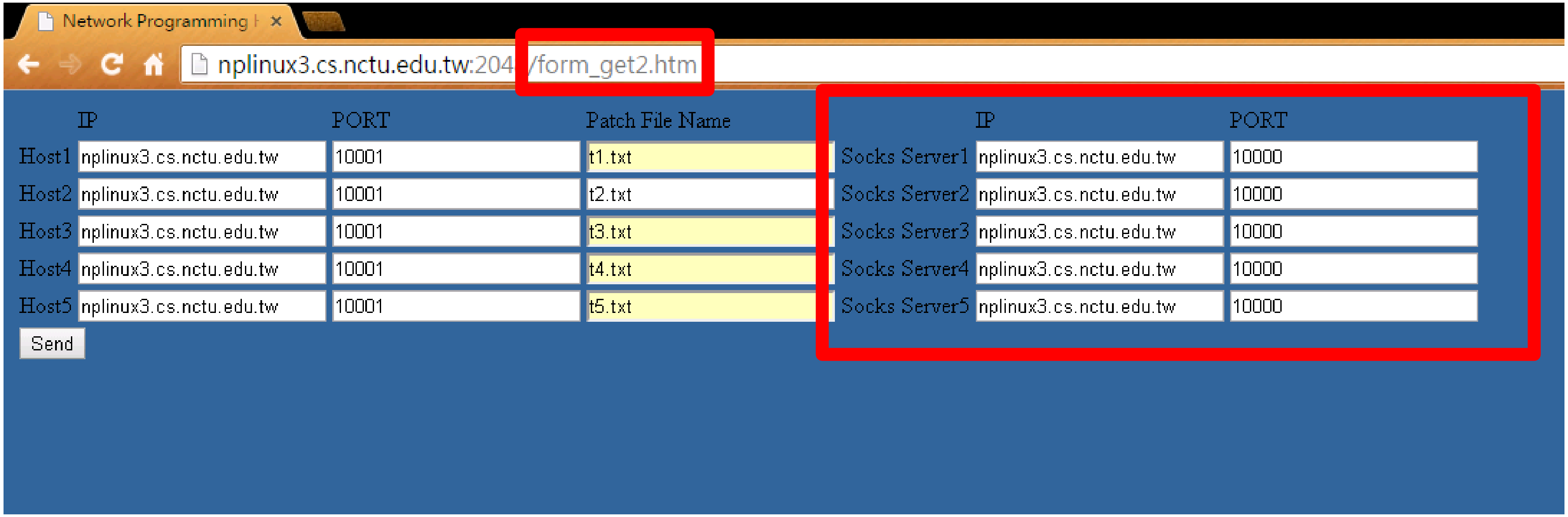
CGI Connection



CGI Connection(Connect mode)



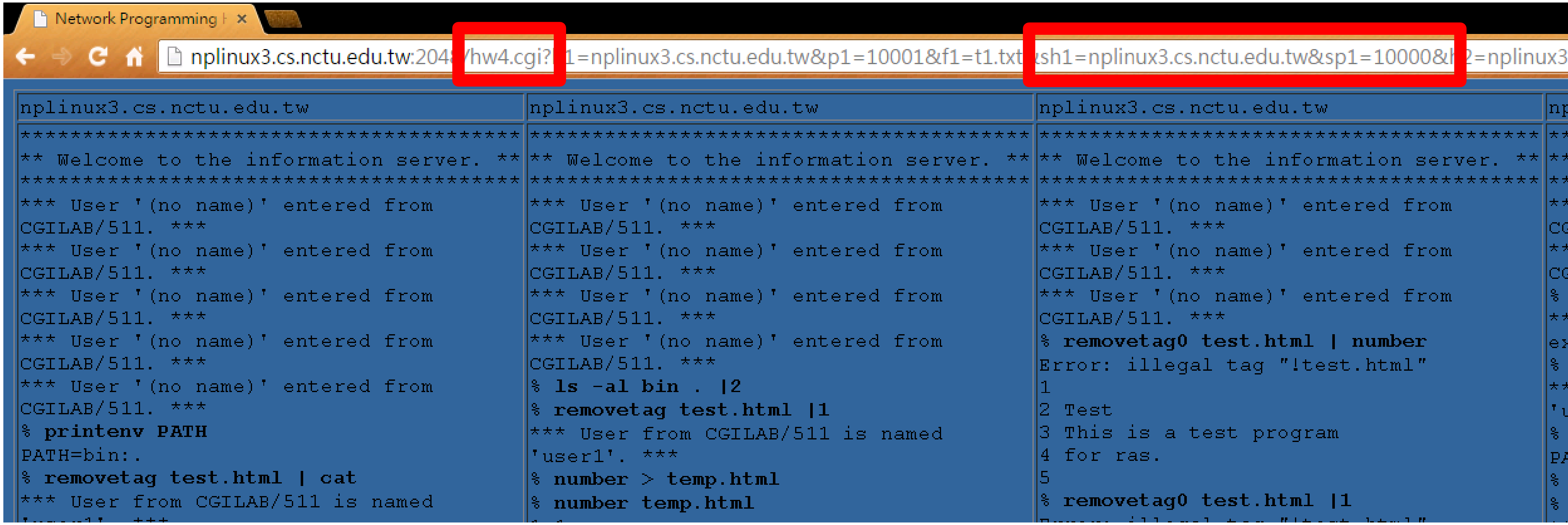
Demo



nplinux3.cs.nctu.edu.tw:204 /form_get2.htm

	IP	PORT	Patch File Name		IP	PORT
Host1	nplinux3.cs.nctu.edu.tw	10001	t1.txt	Socks Server1	nplinux3.cs.nctu.edu.tw	10000
Host2	nplinux3.cs.nctu.edu.tw	10001	t2.txt	Socks Server2	nplinux3.cs.nctu.edu.tw	10000
Host3	nplinux3.cs.nctu.edu.tw	10001	t3.txt	Socks Server3	nplinux3.cs.nctu.edu.tw	10000
Host4	nplinux3.cs.nctu.edu.tw	10001	t4.txt	Socks Server4	nplinux3.cs.nctu.edu.tw	10000
Host5	nplinux3.cs.nctu.edu.tw	10001	t5.txt	Socks Server5	nplinux3.cs.nctu.edu.tw	10000

Demo



Network Programming

```
nplinux3.cs.nctu.edu.tw:204 /hw4.cgi? 1=nplinux3.cs.nctu.edu.tw&p1=10001&f1=t1.txt sh1=nplinux3.cs.nctu.edu.tw&sp1=10000& 2=nplinux3
```

```
nplinux3.cs.nctu.edu.tw
****************************************
** Welcome to the information server. **
****************************************
*** User '(no name)' entered from
CGILAB/511. ***
*** User '(no name)' entered from
CGILAB/511. ***
*** User '(no name)' entered from
CGILAB/511. ***
*** User '(no name)' entered from
CGILAB/511. ***
*** User '(no name)' entered from
CGILAB/511. ***
% printenv PATH
PATH=bin:.
% removetag test.html | cat
*** User from CGILAB/511 is named
```

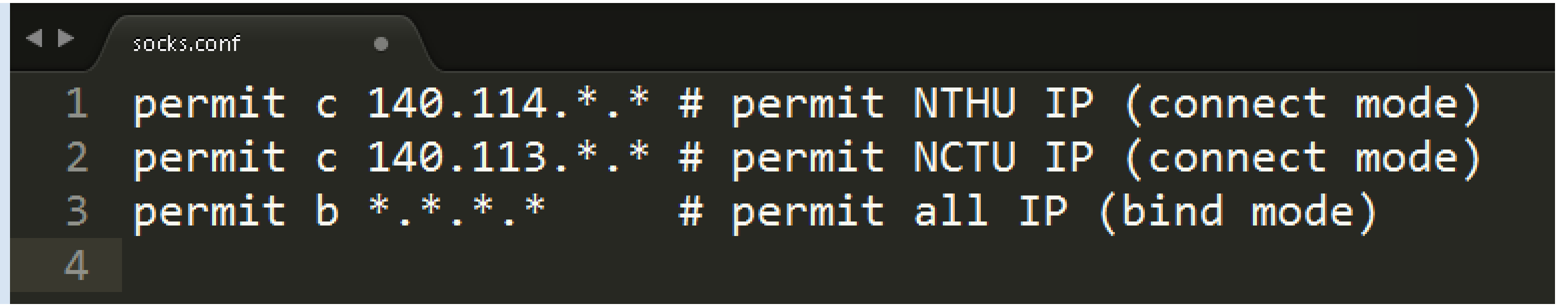
```
nplinux3.cs.nctu.edu.tw
****************************************
** Welcome to the information server. **
****************************************
*** User '(no name)' entered from
CGILAB/511. ***
*** User '(no name)' entered from
CGILAB/511. ***
*** User '(no name)' entered from
CGILAB/511. ***
*** User '(no name)' entered from
CGILAB/511. ***
% ls -al bin . |2
% removetag test.html |1
*** User from CGILAB/511 is named
'user1'. ***
% number > temp.html
% number temp.html
```

```
nplinux3.cs.nctu.edu.tw
****************************************
** Welcome to the information server. **
****************************************
*** User '(no name)' entered from
CGILAB/511. ***
*** User '(no name)' entered from
CGILAB/511. ***
*** User '(no name)' entered from
CGILAB/511. ***
% removetag0 test.html | number
Error: illegal tag "!test.html"
1
2 Test
3 This is a test program
4 for ras.
5
% removetag0 test.html |1
```

```
np
**
**
**
**
CG
**
CG
%
**
ex
%
**
'u
%
PA
%
%
```

Firewall

- When socks server accept a SOCKS4_REQUEST, it will analysis whether DEST_IP is a permitted IP. If the DEST_IP is not allowed, send a SOCKS_REPLY with CD is 91 (rejected).
- You only need to implement a simple firewall. Write permitted IPs into socks.conf
- Your socks server will read socks.conf to make judgement.



```
permit c 140.114.*.* # permit NTHU IP (connect mode)
permit c 140.113.*.* # permit NCTU IP (connect mode)
permit b *.*.*.*     # permit all IP (bind mode)
```

Part III points

- **[CGI SOCKS Client]** (25 points)
  - Close browser’s proxy setting
  - Open your http server, connect to form_get2.htm
  - Key in IP, port, filename, SocksIP, SocksPort
  - Connect to 5 ras/rwg servers through socks sever and check the output
- Test Case (as same as Project III, no hidden test case)
  - t1.txt~t5.txt (5 points each)

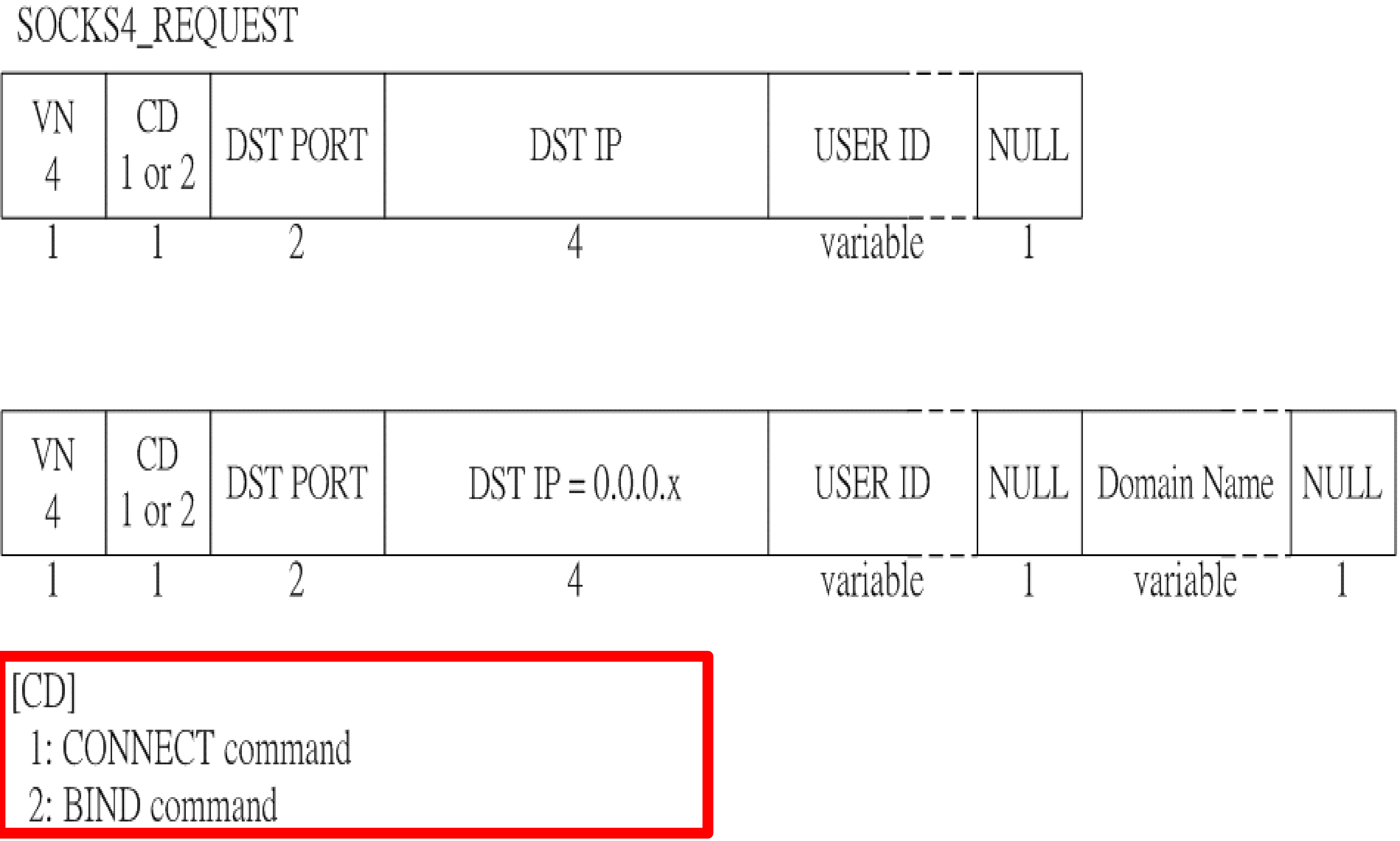
Firewall

- **[Firewall]** (10 points)
  - Allow any DEST_IPs. “*.*.*.*” in socks.conf
  - Only allow connections to NCTU (5 points)
    - “140.113.*.*” in socks.conf
  - Only allow connections to NTHU (5 points)
    - “140.114.*.*” in socks.conf

END

Implementation Details

SOCKS4_REQUEST



SOCKS4_REQUEST

VN 4	CD 1 or 2	DST PORT	DST IP	USER ID	NULL
1	1	2	4	variable	1

VN 4	CD 1 or 2	DST PORT	DST IP = 0.0.0.x	USER ID	NULL	Domain Name	NULL
1	1	2	4	variable	1	variable	1

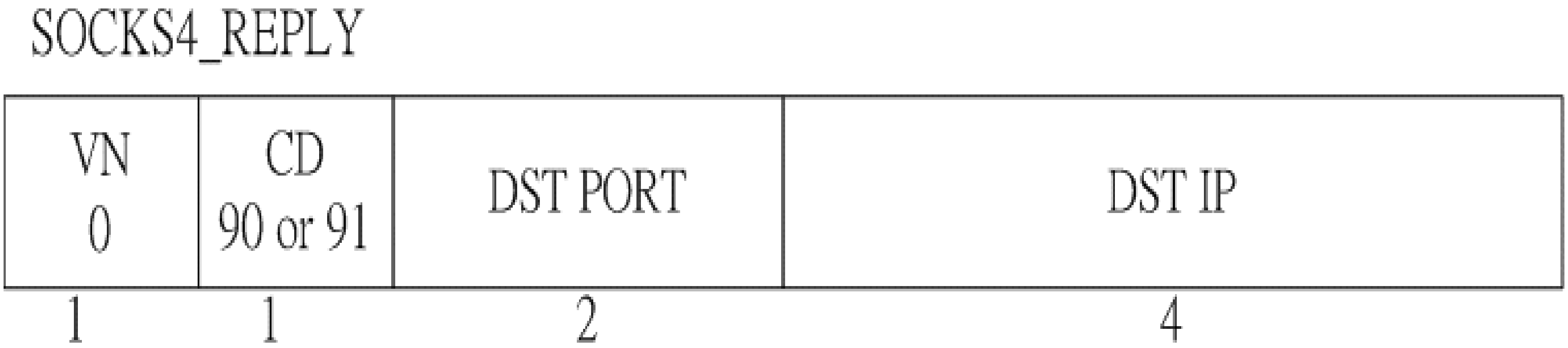
[CD]
1: CONNECT command
2: BIND command

SOCKS4_REQUEST

Request

```
read(sock, buffer, size);
unsigned char VN = buffer[0] ;
unsigned char CD = buffer[1] ;
unsigned int DST_PORT = buffer[2] << 8 |
                              buffer[3] ;
unsigned int DST_IP = buffer[4] << 24 |
                         buffer[5] << 16 |
                         buffer[6] << 8 |
                         buffer[7] ;
char* USER_ID = buffer + 8 ;
```

SOCKS4_REPLY



SOCKS4_REPLY

VN	CD	DST PORT	DST IP
0	90 or 91		
1	1	2	4

[CD]

90: request granted

91: request rejected or failed

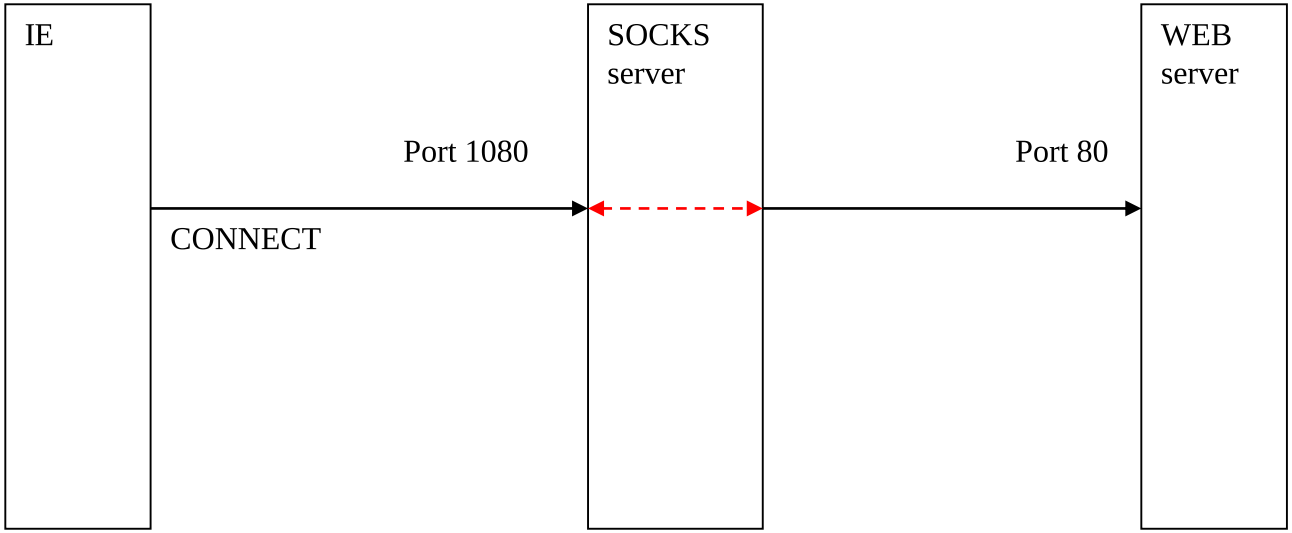
SOCKS4_REPLY

Reply

```
package[0] = 0;
package[1] = (unsigned char) CD ; // 90 or 91
package[2] = port / 256;
package[3] = port % 256;
package[4] = ip >> 24;
    // ip = ip in SOCKS4_REQUEST for connect mode
    // ip = 0 for bind mode
package[5] = (ip >> 16) & 0xFF;
package[6] = (ip >> 8)  & 0xFF;
package[7] = ip & 0xFF;
write(sock, package, 8);
```

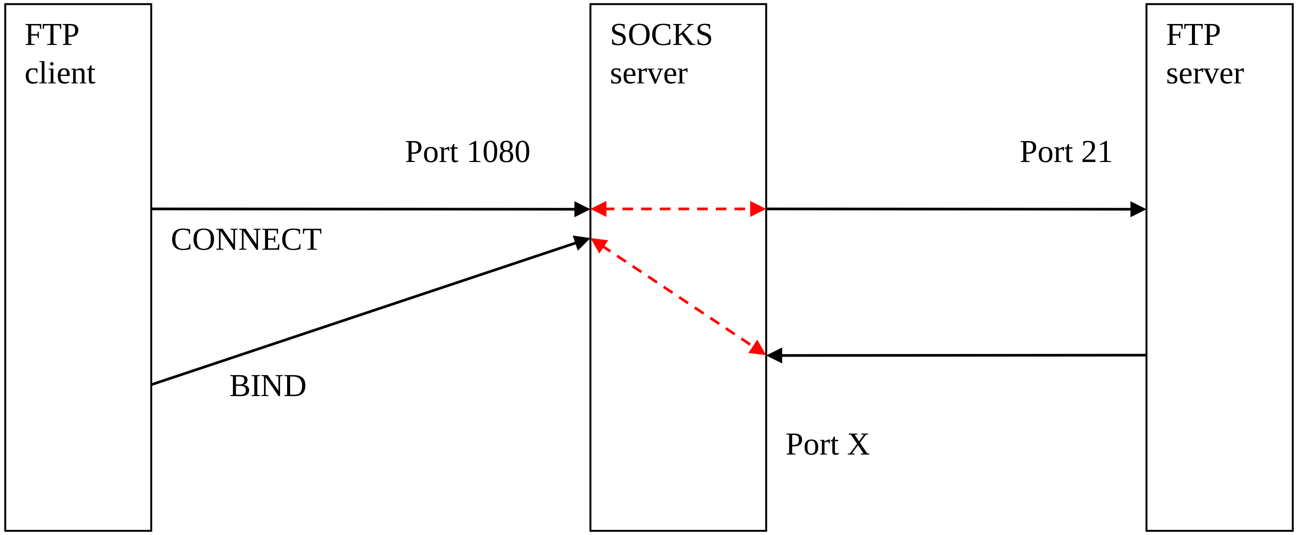
Port

- [Port]
  - Connect mode: is the DST PORT in SOCKS4_REQUEST



Port

- [Port]
  - Bind mode: newly binded port in SOCKS server



Implementation Details

- Process：
  - master socket(listener)不斷地listen，有連線(SRC)來就fork一個process(SOCKS) 去處理，然後繼續listen
  - SOCKS 與SRC 連線溝通

    1.收SOCKS4_REQUEST格式封包

    2.check 是否可以過防火牆(socks.conf)，並回傳SRC SOCKS4_REPLY
  - CONNECT mode

    1.從REQUEST裡取出dest的IP與PORT

    2.SOCKS連線到DEST

    3.SOCKS幫SRC與DEST做資料傳導的動作

    - SRC傳來的資料 －＞傳給DEST
    - DEST傳來的資料 －＞傳給SRC

Implementation Details

- BIND mode：

  1. SOCKS 先去BIND一個port(BIND_PORT)
  2. SOCKS listen該port，回傳給SRC監聽Port，DEST就會自己連過來
  3. SOCKS accept DEST之後，要再丟一個SOCKS4_REPLY給SRC <-- **<u>重要</u>**!!!!!!!!!

  4.SOCKS幫SRC與DEST做資料傳導的動作

  - SRC傳來的資料 －＞傳給DEST
  - DEST傳來的資料 －＞傳給SRC

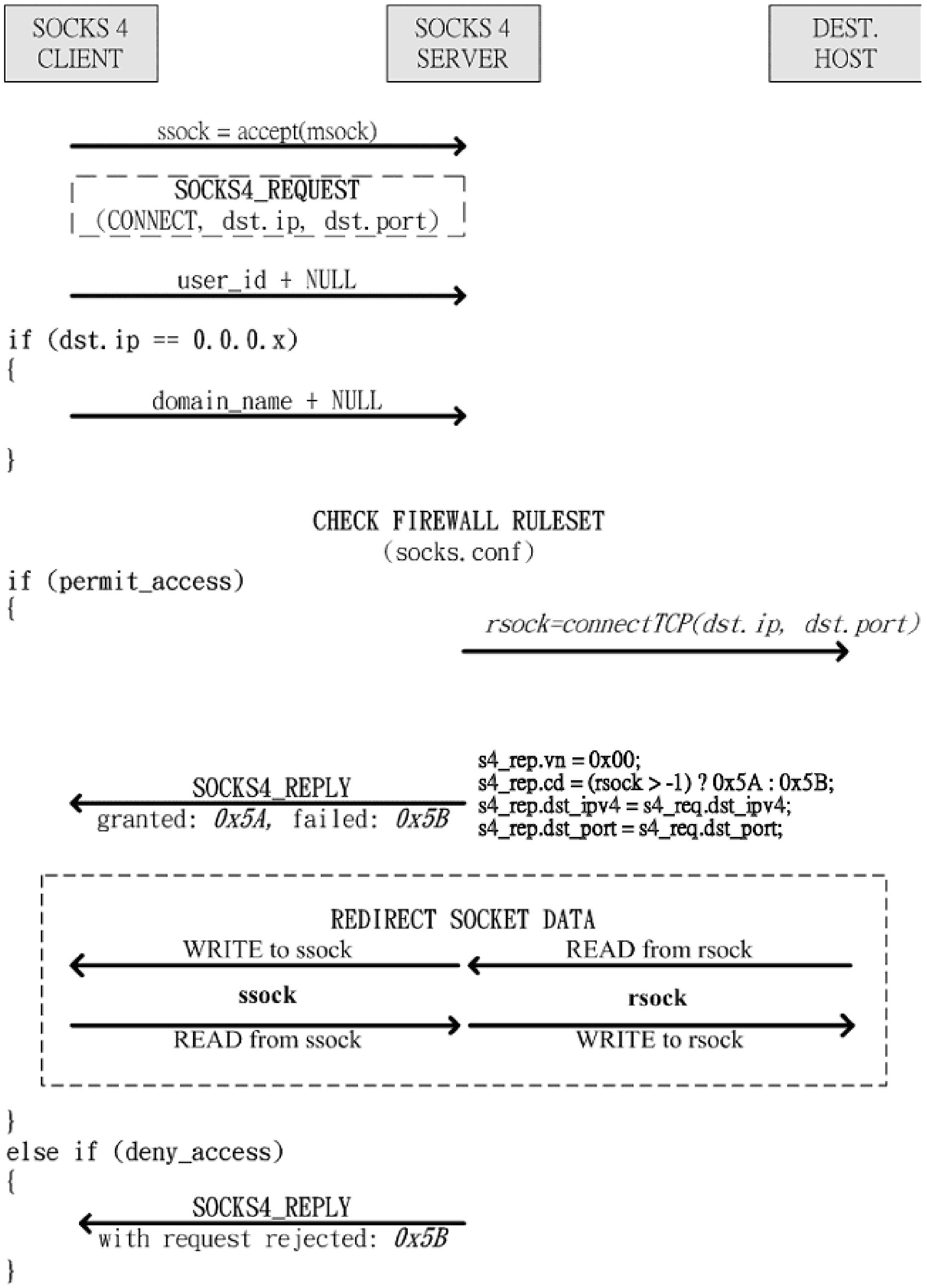
Notes

- SOCKS_REQUEST , SOCKS_REPLY
  - 1 byte : unsigned char
  - 2 byte : unsigned char[2]
  - 4 byte : unsigned char[4]
- Port formulation e.g. port = 1234
  - unsigned char port[2]
  - port[0] = 4
  - port[1] = 210
  - (hint : (int)port = port[0]*256 + port[1] ==> 1234 = 4*256 + 210)

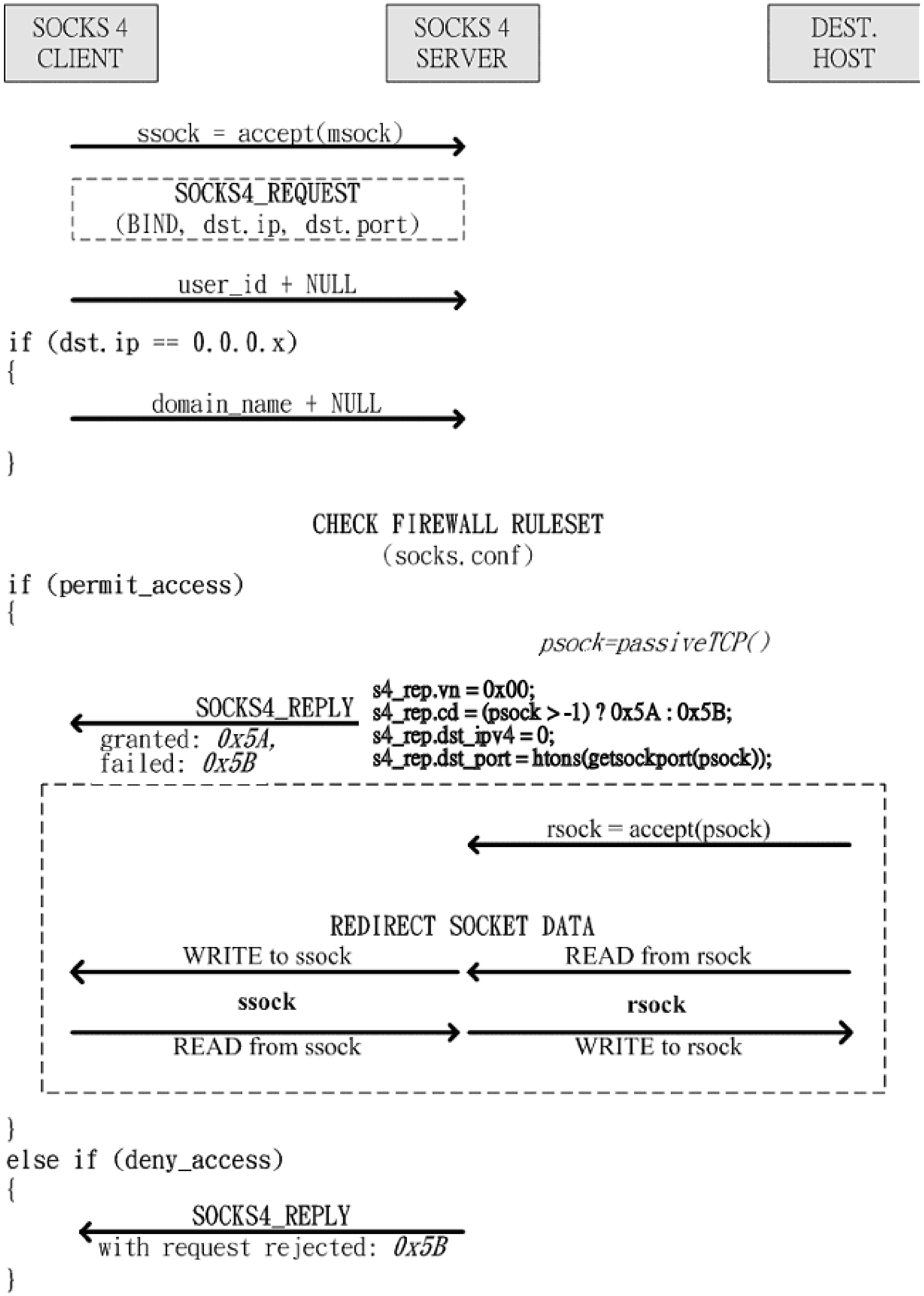
Notes

- IP formulations e.g. IP = 140.113.1.2
  - unsigned char IP[4]
    - IP[0] = 140
    - IP[1] = 113
    - IP[2] = 1
    - IP[3] = 2
- In BIND mode, you need to ensure the connection between client and server is built before data transfer.

SOCKS Version 4 Protocol
(CONNECT Operation)



SOCKS Version 4 Protocol
(BIND Operation)



```
ssock = accept(msock)

SOCKS4_REQUEST
(BIND, dst.ip, dst.port)

user_id + NULL

if (dst.ip == 0.0.0.x)
{
    domain_name + NULL
}

CHECK FIREWALL RULESET
(socks.conf)

if (permit_access)
{
    psock=passiveTCP()

    SOCKS4_REPLY
    granted: 0x5A,
    failed: 0x5B

    s4_rep.vn = 0x00;
    s4_rep.cd = (psock > -1) ? 0x5A : 0x5B;
    s4_rep.dst_ipv4 = 0;
    s4_rep.dst_port = htons(getsockport(psock));

    rsock = accept(psock)

    REDIRECT SOCKET DATA
    WRITE to ssock        READ from rsock
    ssock                 rsock
    READ from ssock       WRITE to rsock
}
else if (deny_access)
{
    SOCKS4_REPLY
    with request rejected: 0x5B
}
```