**Computer Systems Technology**

British Columbia Institute of Technology

COMP 8005 – Final Project- Design

Albert Huang&

Aiyan Ma

Mar 25, 2018

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Date** | **Version** | **Revised by** | **Remarks** |
| 1 | 2018-Mar-26 | V0.1 | Aiyan,Ma | The initial draft |
|  |  |  |  |  |
|  |  |  |  |  |

# Table of Contents

[Table of Contents 1](#__RefHeading___Toc1005_382563431)

[Design Work for Port forwarding Server 2](#__RefHeading___Toc1007_382563431)

[1. Introduction 2](#__RefHeading___Toc1009_382563431)

[1.1 Config File Format 2](#__RefHeading___Toc1011_382563431)

[1.2 Main Flow 3](#__RefHeading___Toc1013_382563431)

[2. Flow Chart 4](#__RefHeading___Toc1015_382563431)

[3. Pseudo Code 5](#__RefHeading___Toc1017_382563431)

[4 GUI Module 5](#__RefHeading___Toc1019_382563431)

# Design Work for Port forwarding Server

# Introduction

The port forwarding server is based on the Select server tasks done in the previous project, with the main changes as below:

1) Handing the configuration of forwarding pair entries

2) Instead of echoing back, setup the connection with remote hosts and forwarding the data to them

3) get the response and deliver them to the initial clients

## 1.1 Config File Format

The config file for port forwarding is designed to record the entry of localPort and rHost pairing, the format is:

$localPort<SPACE>$remoteIP:$remotePort

means:

All tcp traffices sent to local port $localPort will be forwarded to $remoteIP with port $remotePort.

For instance:

*7000 192.168.0.22:22*

*8000 192.168.0.25:22*

*9000 192.168.0.25:80*

Means:

All tcp traffices sent to local port 7000 will be forwarded to 192.158.0.22 port 22

All tcp traffices sent to local port 8000 will be forwarded to 192.158.0.22 port 22

All tcp traffices sent to local port 9000 will be forwarded to 192.158.0.22 port 80

## 1.2 Main Flow

When the server launched,

1) Register the signal to handle the Control+C;

2) Reading the config file to put it in the dictionary for each entry pair *ports\_setting.update({lport:rhost})*

3) Bind and start the server socket for each entry pair

4) For the Select Server, doing the call back for:

When Accept: accept new connections, get connection to the remote hosts,and Register the EVENT\_READ;

When ReadyForRead: doing datatrans()

5) Main PortForwarding:

Get remote part from socket pairs:

Receive the data and send to the remote side;

# Flow Chart

# 

# **Pseudo Code**

1.ReadPortsList from ports.conf

2.According to the Portslist pairs, started the server sockets by binding to the local port one by one, config them to UN\_BLOCK;

3 register the select EVENTS ( EVENT\_ACCEPT and EVENT\_READ) by callbacks

4 Execute the ACCEPT and DataTrans function according to the call back events ( Forever or when Server is Running)

# **4 GUI Module**

For the GUI module, it makes use of the tkinter python module, to provide the GUI for configuration and monitoring purpose. This module is quite straight forward

and the brief Pseudo code as following:

# readPortsList when launched

# init the window

# retrieve and display the data from server

## 