Comp-3670 Finals Project Report

Network Discovery Tool

Ryan Raffoul and Donovan Longo

Contents

1 Overview

The application we chose to create is a Network Discovery Tool. In this Network Discovery tool, the Server begins by getting a Hostname (not IP Address) from the Client and completes a WHOIS Lookup (if applicable, WHOIS Lookup is a Server that can fetch information about any Hostname), finds all IP addresses for this Hostname, and then for each of these IP Addresses found, the tool will find if it is reachable (ping request), and find all the open ports for this IP Address. All this information is sent to the Client and the Client proceeds to print all information to the Discovery Data txt file.

2 Objectives

The goal of this Network Discovery Tool is to be able to provide a way to get valuable information about a Hostname to be able to then use useful information to then connect to these IP Addresses in any way, or just save to files for later use. The goals of the design of the application include making it as simple as possible, efficient, easy to use, error-free, and as extensible as possible.

3 Design

1 Communication Pattern

The Communication architecture of this Network Discovery Tool is a Server Client architecture. The Server is multi-threaded to allow for the multiple Clients to connect to the Server even though the connection is one-to-one. The Server is always running, and the Client can connect to the Server using the IP Address and port number of the Server. The Server is responsible for all the methods to get the information on the Hostname and the Client is responsible for sending a

Hostname and then getting all information from the Server to output to the Discovery Data txt file.

2 Design Goals

The Design goals of this Network Discovery Tool are making it as simple as possible, ensuring the data transfer is efficient, make the application scalable, gain extensibility, reliable message exchange, ensure security (cannot steal information or connection using Java Sockets), persistent connections, and error-free (use Java catch to get any errors or failures).

3 Message Design

The Message design of this Network Discovery tool was created using text-protocols. The Server and Client can read and write to each other. All data being transferred is sent using Strings so all data can be transferred easily and prevent any reading errors. The data is sent mainly in loops by first sending the size, so the Client has the ability to read in a loop if it is needed. The message design goal was to make it as easy to implement as possible, easy to understand, easy to extend, and easy to test. Command and Control was used for the connection to ensure a swift and easy connection between Server and Client.

4 Communication Rules

The Communication rules of this Network Discovery Tool are as follows:

- The Client enters the Server’s IP Address (127.0.0.1).
- The Server waits for a Client(s) to connect.
- The Client(s) connects to the Server using IP address and Port number.
- The Server asks the Client for a Hostname (not IP Address).

- The Server then does the WHOIS Lookup if applicable.
- The Server finds all IP Addresses for this Hostname.
- The Server check if each IP Address is reachable (ping request).
- The Server finds all Open Ports for each of these IP Addresses that are reachable.
- The Server sends all the information above to the Client.
- The Client saves all this data to a txt file.
- The Server and Client are then both asked to continue or terminate the connection.

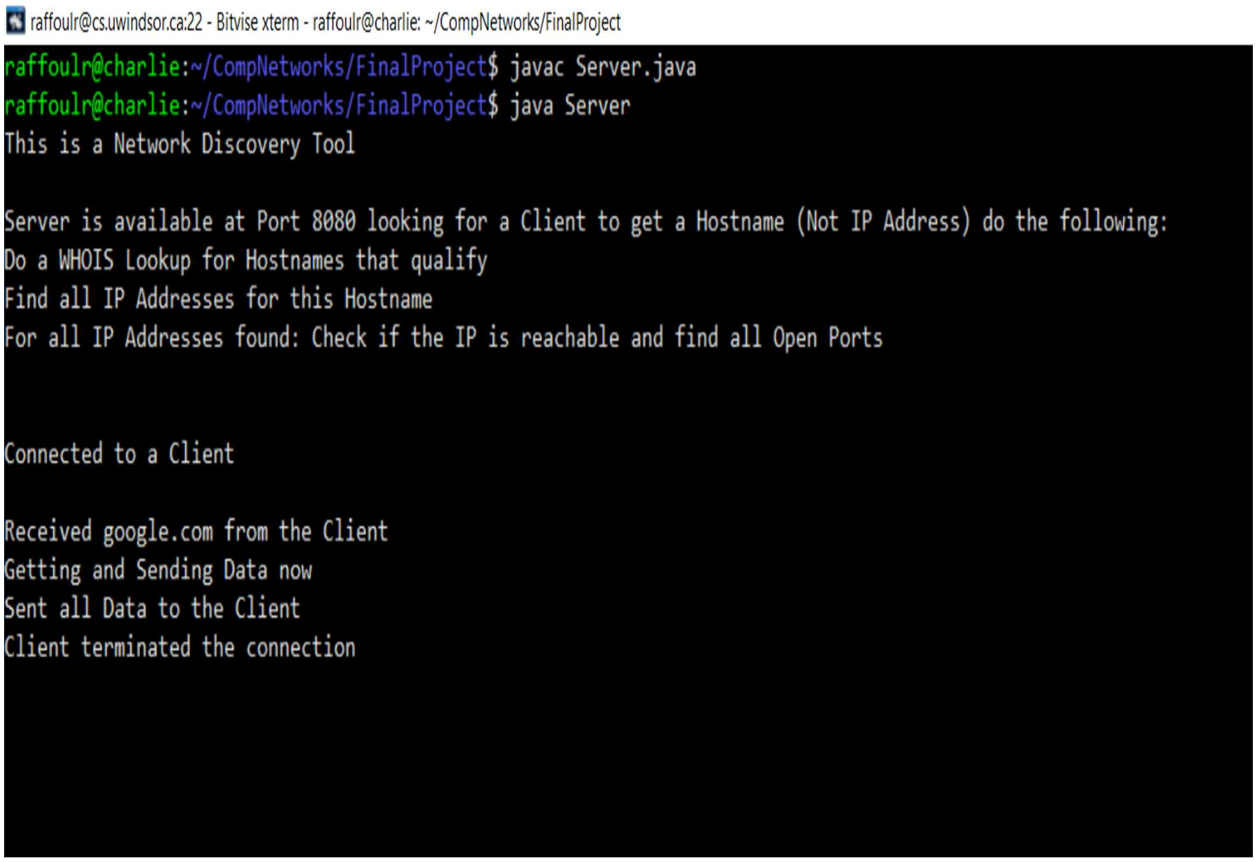
4 Deployment and Testing Instructions

The Deployment and Testing Instructions for this Network Discovery Tool are as follows:

- The Server runs first by compiling and running the Java file (Server.java).
- The Client then runs by compiling and running the Java file (Client.java).
- The Server and Client then communicate until the connection is terminated.

5 Experiment

Below is a sample experiment using the Hostname google.com to test the application. This is the input and output of the Server and Client.



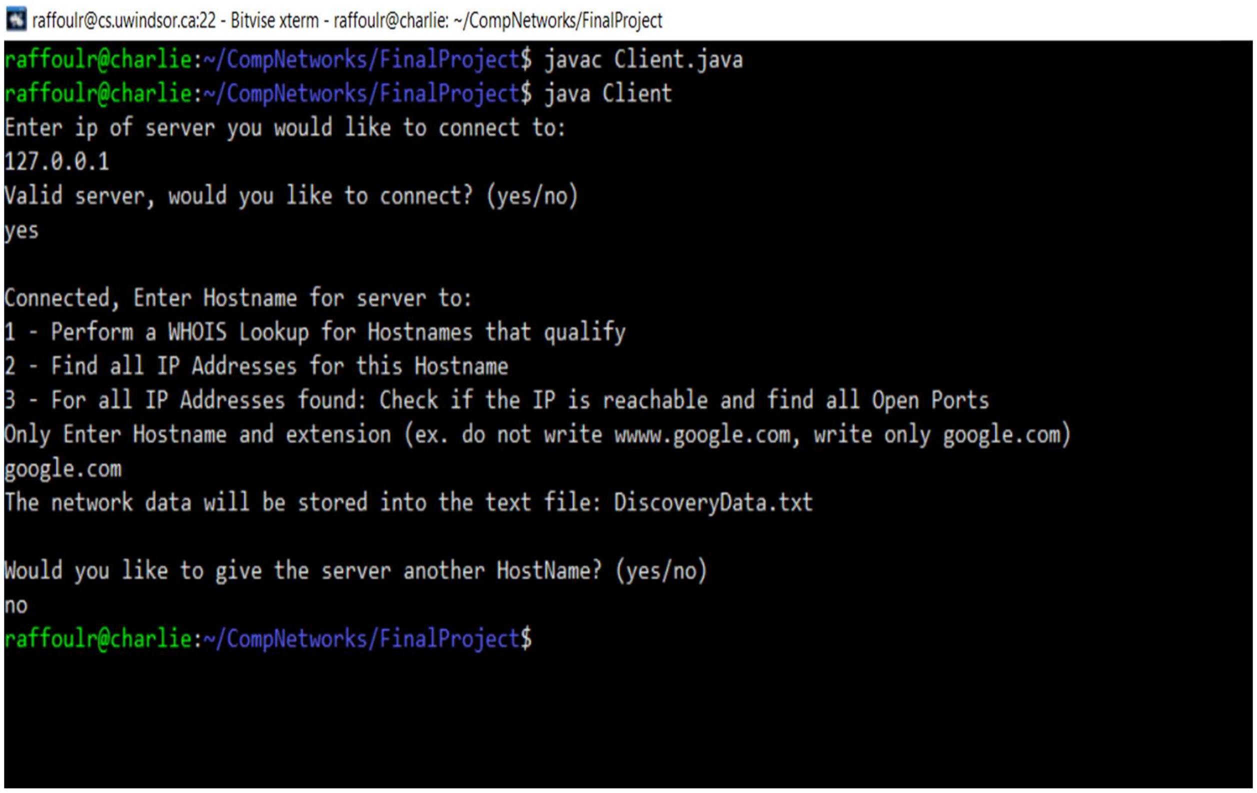
raffoulr@cs.uwindsor.ca:22 - Bitvise xterm - raffoulr@charlie: ~/CompNetworks/FinalProject

```
raffoulr@charlie:~/CompNetworks/FinalProject$ javac Server.java
raffoulr@charlie:~/CompNetworks/FinalProject$ java Server
This is a Network Discovery Tool

Server is available at Port 8080 looking for a Client to get a Hostname (Not IP Address) do the following:
Do a WHOIS Lookup for Hostnames that qualify
Find all IP Addresses for this Hostname
For all IP Addresses found: Check if the IP is reachable and find all Open Ports


Connected to a Client

Received google.com from the Client
Getting and Sending Data now
Sent all Data to the Client
Client terminated the connection
```



raffoulr@cs.uwindsor.ca:22 - Bitvise xterm - raffoulr@charlie: ~/CompNetworks/FinalProject

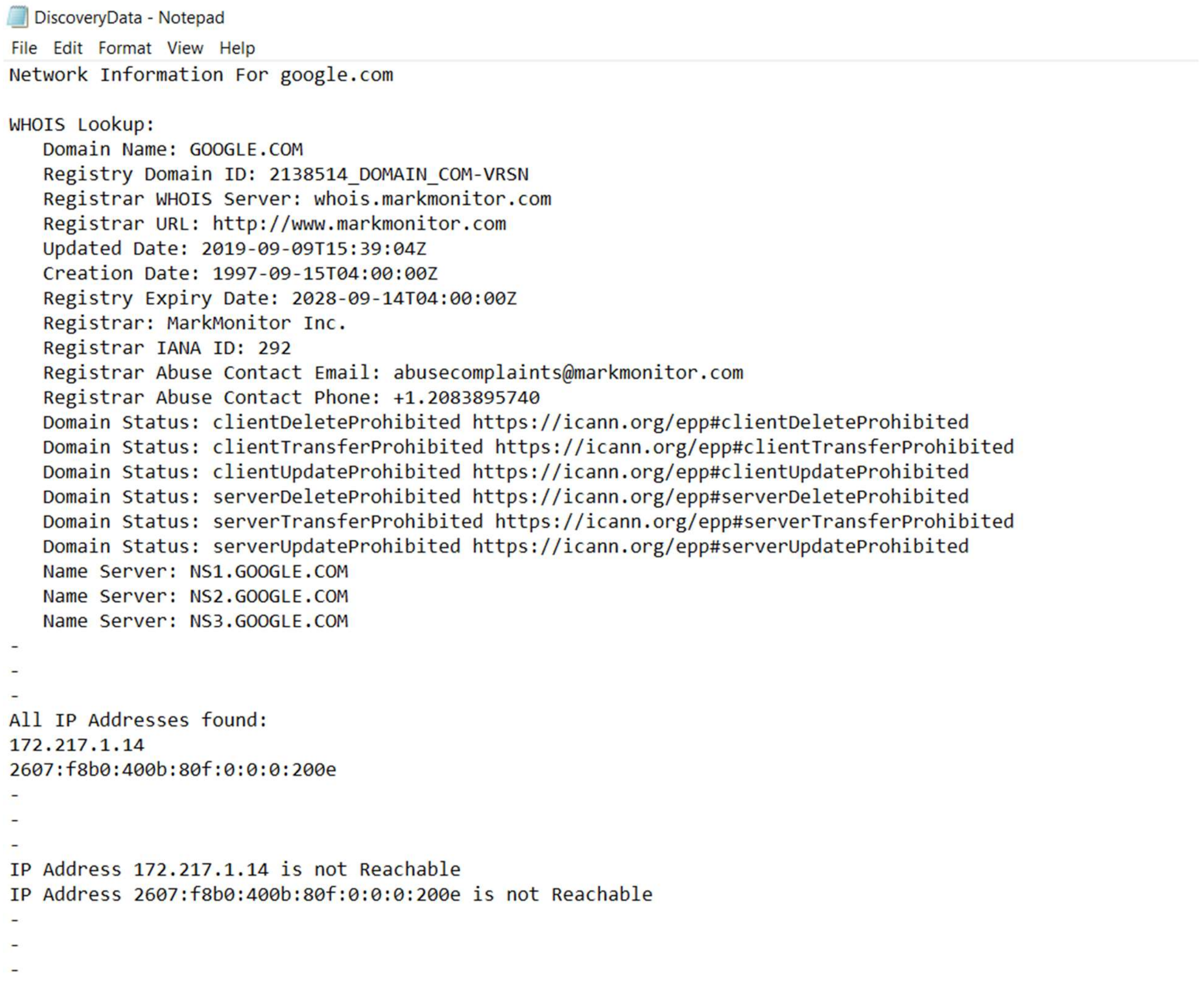
```
raffoulr@charlie:~/CompNetworks/FinalProject$ javac Client.java
raffoulr@charlie:~/CompNetworks/FinalProject$ java Client
Enter ip of server you would like to connect to:
127.0.0.1
Valid server, would you like to connect? (yes/no)
yes

Connected, Enter Hostname for server to:
1 - Perform a WHOIS Lookup for Hostnames that qualify
2 - Find all IP Addresses for this Hostname
3 - For all IP Addresses found: Check if the IP is reachable and find all Open Ports
Only Enter Hostname and extension (ex. do not write wwww.google.com, write only google.com)
google.com
The network data will be stored into the text file: DiscoveryData.txt

Would you like to give the server another HostName? (yes/no)
no
raffoulr@charlie:~/CompNetworks/FinalProject$
```

6 Results

The results of this experiment are seen below. This is what was written to the Data Discovery txt file.



```
Network Information For google.com

WHOIS Lookup:
   Domain Name: GOOGLE.COM
   Registry Domain ID: 2138514_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.markmonitor.com
   Registrar URL: http://www.markmonitor.com
   Updated Date: 2019-09-09T15:39:04Z
   Creation Date: 1997-09-15T04:00:00Z
   Registry Expiry Date: 2028-09-14T04:00:00Z
   Registrar: MarkMonitor Inc.
   Registrar IANA ID: 292
   Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
   Registrar Abuse Contact Phone: +1.2083895740
   Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
   Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
   Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
   Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
   Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
   Name Server: NS1.GOOGLE.COM
   Name Server: NS2.GOOGLE.COM
   Name Server: NS3.GOOGLE.COM
-
-
-
All IP Addresses found:
172.217.1.14
2607:f8b0:400b:80f:0:0:0:200e
-
-
-
IP Address 172.217.1.14 is not Reachable
IP Address 2607:f8b0:400b:80f:0:0:0:200e is not Reachable
-
-
-
```

In this case, the WHOIS Lookup was done for google.com along with finding 2 IP Addresses.

These IP Addresses are not reachable.

See the Test Cases document for all Experiments and Results.