Visual Network Explorer

Joseph True
jtrueprojects@gmail.com
Updated: Oct, 2019

Code: https://github.com/joseph-true/visual-network-explorer

Originally created and published July 1, 2014 as a project for graduate course:
CS-513 - Introduction to Local and Wide Area Networks
WPI, Summer, 2014

Abstract

This report outlines the design and development of Visual Network Explorer - a program that provides an enhanced visual interface and presentation for the information retrieved by the Ping and Traceroute networking commands.

Ping and Traceroute are standard networking commands that provide textural information about times and paths to access remote systems. This program uses the textural information retrieved by these commands to look up additional information such as city and country locations of the remote servers that are accessed. The collected information is then combined with a geographical map to visually show the physical locations and network path that was followed to access the remote systems.

The program was written in Microsoft C# to run under the Microsoft Windows operating system. It was developed in Microsoft Visual Studio 2010 on a system running Windows 7. The code combines the C# Ping API with other programming resources from MaxMind and Google. The MaxMind GeoIP C# API and GeoIP database was used to look up the physical geographical location of systems based on their IP address. The Google Maps API was used to display the resulting information on a visual geographical map along with labels and paths.

The design of the program focuses on the visual presentation of the network information to the user. The combined information displays in a combined HTML and text format inside of an embedded web browser control within a standard Windows application.

The report includes test cases used to verify the correct operation of the program. Note that some web sites do not allow the ping command. They have configured their web servers to not accept ICMP commands. This program will only work with remote systems and web sites that accept ICMP commands.

Contents

Project Description

This section contains a brief description of the project and summarizes what the program can do.

Program Overview

The overall goal was to create a Windows-based program with a user interface that visually displays information about network hosts based on the information retrieved from the ping and traceroute commands. The basic goal was to combine the ping command with other APIs to provide a richer, visual display of the retrieved text information. The extended goal was to perform a traceroute network command and plot the results on a map to show the network path and nodes from the user's computer to the remote host system.

These goals were achieved by combining the C# ping API with geolocation information from the MaxMind GeoIP API and the map display features of the Google Static Maps API.

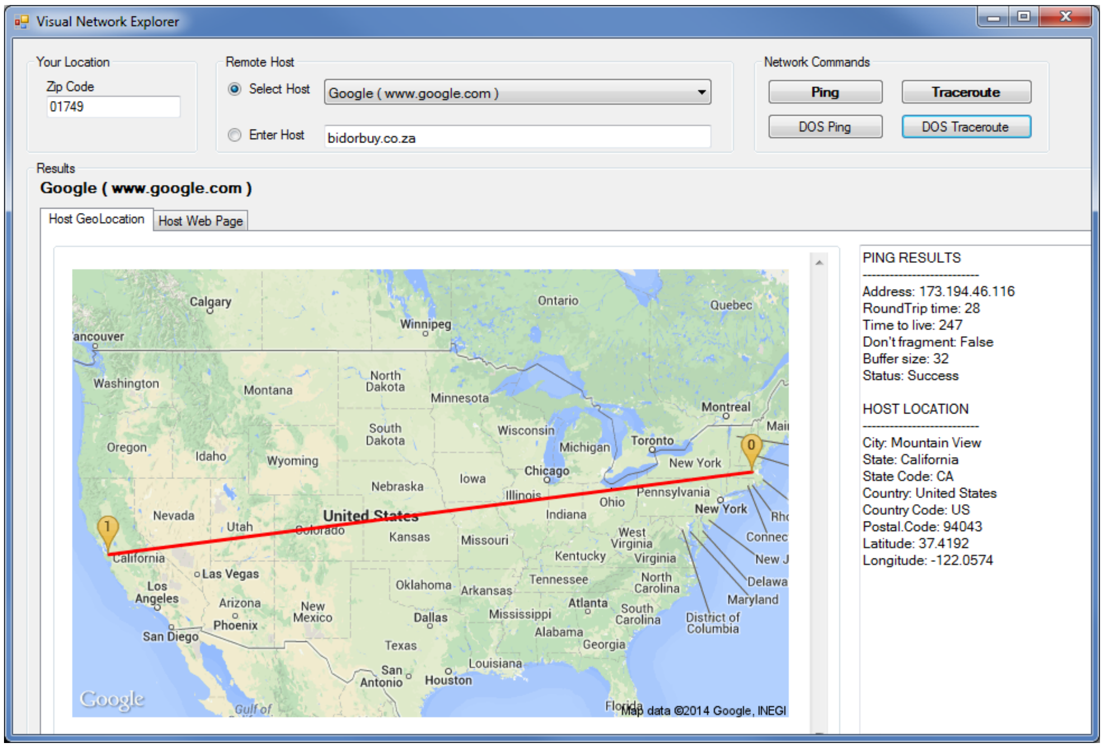
What the Program Can Do

You can use the program to run a ping or traceroute command for a selected host and view the results in the following formats:

- A map of the resulting path from the user to the host system. For the traceroute command, the path includes all of the nodes in between that have valid geolocation information.
- A text summary of the results.
- The web page of the host system (if applicable).

The program includes a list of about 25 example hosts that you can run ping and traceroute on. You can also enter your own host name to test.

An example of the program displaying the results from a traceroute to the host google.com is shown in the following screenshot.



```
PING RESULTS
---------------------------
Address: 173.194.46.116
RoundTrip time: 28
Time to live: 247
Don't fragment: False
Buffer size: 32
Status: Success

HOST LOCATION
---------------------------
City: Mountain View
State: California
State Code: CA
Country: United States
Country Code: US
Postal Code: 94043
Latitude: 37.4192
Longitude: -122.0574
```

Development Environment and Configuration

The program was developed in the C# programming language in Microsoft Visual Studio 2010. It was designed to run in the Windows operating system and was developed on a system running Windows 7.

The code uses the following APIs:

- **C# Ping API** – provides programmatic access to ping and traceroute functions
- **MaxMind GeoIP API** – provides geolocation information for known IP addresses
- **Google Maps API** – draws maps with an overlay of lines (paths) and points (markers) in a web browser or HTML document

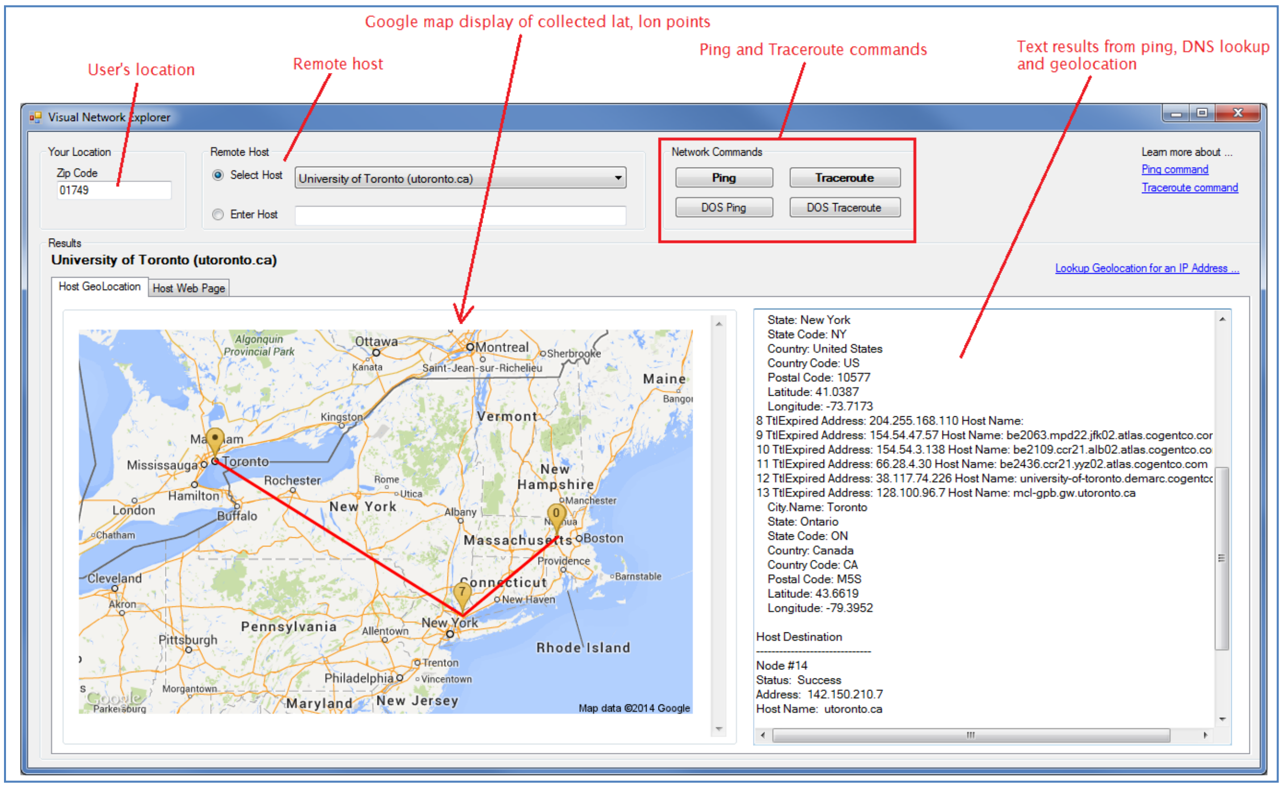
Detailed Design

The overall design approach was divided into two efforts;

1. User Interface - Creating the user interface and displaying the text and map results
2. Code - Programming the logic for using the ping, GeoIP, and Google map APIs.

User Interface

A screenshot of the user interface is shown in the following image.



Zip Code: 01749

Select Host: University of Toronto (utoronto.ca)

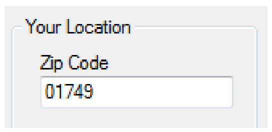
Results

University of Toronto (utoronto.ca)

```
   State: New York
   State Code: NY
   Country: United States
   Country Code: US
   Postal Code: 10577
   Latitude: 41.0387
   Longitude: -73.7173
8 TtlExpired Address: 204.255.168.110 Host Name:
9 TtlExpired Address: 154.54.47.57 Host Name: be2063.mpd22.jfk02.atlas.cogentco.cor
10 TtlExpired Address: 154.54.3.138 Host Name: be2109.ccr21.alb02.atlas.cogentco.co
11 TtlExpired Address: 66.28.4.30 Host Name: be2436.ccr21.yyz02.atlas.cogentco.com
12 TtlExpired Address: 38.117.74.226 Host Name: university-of-toronto.demarc.cogentco
13 TtlExpired Address: 128.100.96.7 Host Name: mcl-gpb.gw.utoronto.ca
   City.Name: Toronto
   State: Ontario
   State Code: ON
   Country: Canada
   Country Code: CA
   Postal Code: M5S
   Latitude: 43.6619
   Longitude: -79.3952

Host Destination
--------------------------------
Node #14
Status:  Success
Address:  142.150.210.7
Host Name:  utoronto.ca
```

User's location

The user's location can be entered in a Zip Code text box. This location is used as the starting point for the paths that are displayed on the map.

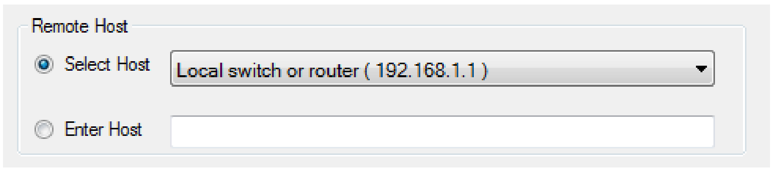


Your Location

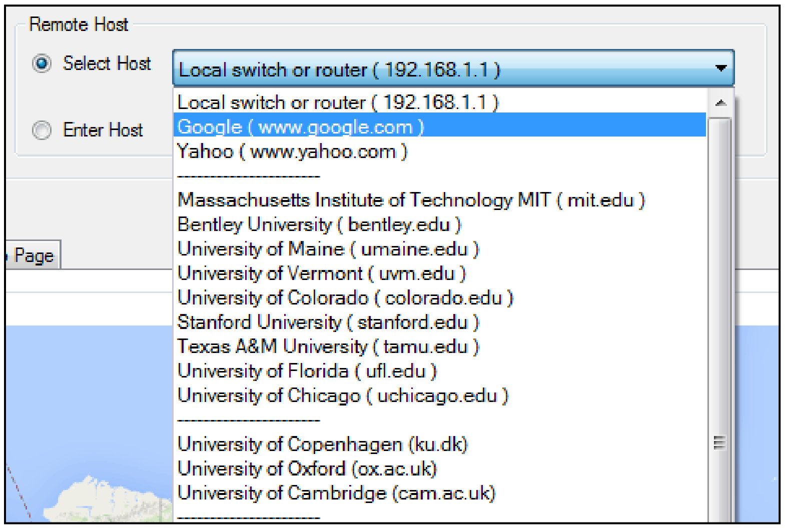
Zip Code: 01749

Remote Host selection

The user interface allows the user to either select a host from a list of example hosts or to enter their host name.



A list of 25 example hosts are included in a drop-down list to allow the user to quickly try the ping and traceroute commands on hosts that respond to ping requests. Having a pre-determined list of example hosts was an important part of developing the program since not all hosts allow ping requests.



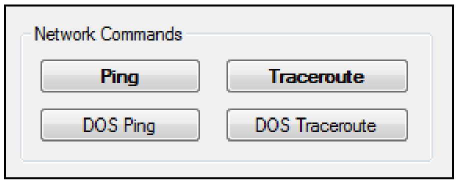
Local switch or router (192.168.1.1)
Google (www.google.com)
Yahoo (www.yahoo.com)

Massachusetts Institute of Technology MIT (mit.edu)
Bentley University (bentley.edu)
University of Maine (umaine.edu)
University of Vermont (uvm.edu)
University of Colorado (colorado.edu)
Stanford University (stanford.edu)
Texas A&M University (tamu.edu)
University of Florida (ufl.edu)
University of Chicago (uchicago.edu)

University of Copenhagen (ku.dk)
University of Oxford (ox.ac.uk)
University of Cambridge (cam.ac.uk)

Ping and Traceroute buttons

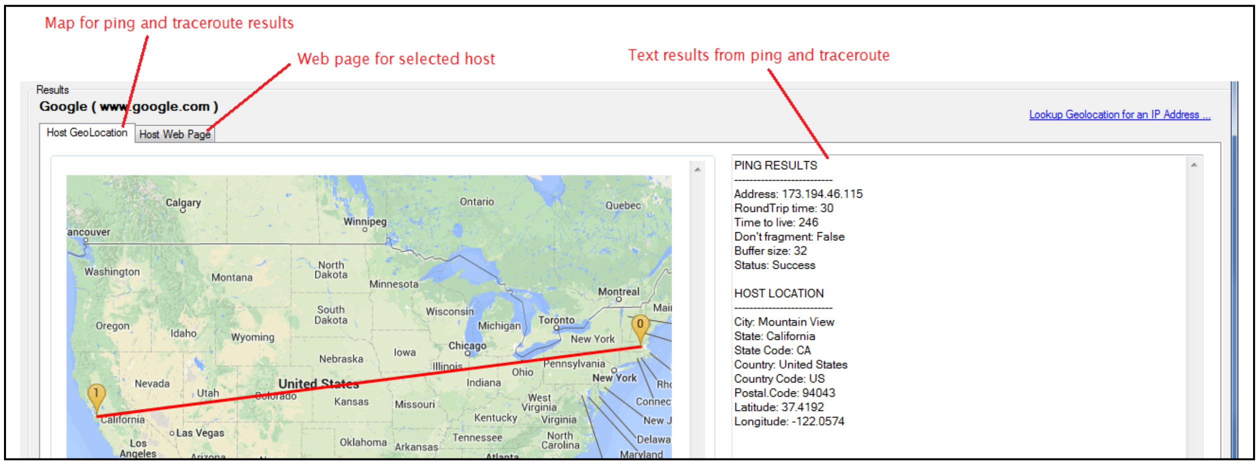
Two sets of buttons are provided to perform ping and traceroute commands on the selected host using either the C# ping API or the DOS versions of the network commands.



The buttons for the DOS versions of the ping (ping.exe) and traceroute (tracert.exe) allow the user to run these commands in a console window for comparison purposes. The ability to run the DOS commands was also helpful during development to compare results.

Displaying Results

The program uses embedded web browser controls and a textbox to display the results from running ping and traceroute.


Map for ping and traceroute results
Web page for selected host
Text results from ping and traceroute
Results
Google (www.google.com)
Host GeoLocation
Host Web Page
Lookup Geolocation for an IP Address ...

```
PING RESULTS
---------------------------
Address: 173.194.46.115
RoundTrip time: 30
Time to live: 246
Don't fragment: False
Buffer size: 32
Status: Success

HOST LOCATION
---------------------------
City: Mountain View
State: California
State Code: CA
Country: United States
Country Code: US
Postal.Code: 94043
Latitude: 37.4192
Longitude: -122.0574
```

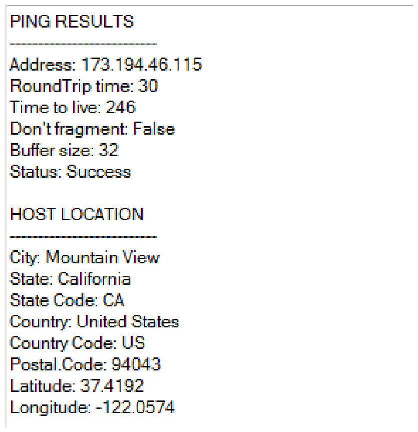
Displaying Results with the Web Browser Control

The program includes two embedded web browser controls. One web browser displays the Google map. The second web browser displays the web page of the selected host. Displaying the web page of the host (if available) provides an additional reference to the user about the host and what it provides.

Displaying Text Results

The text results for the ping command display the standard fields of Address, RoundtripTime (milliseconds) and Status. If available, geolocation and host name are also displayed for the host system.

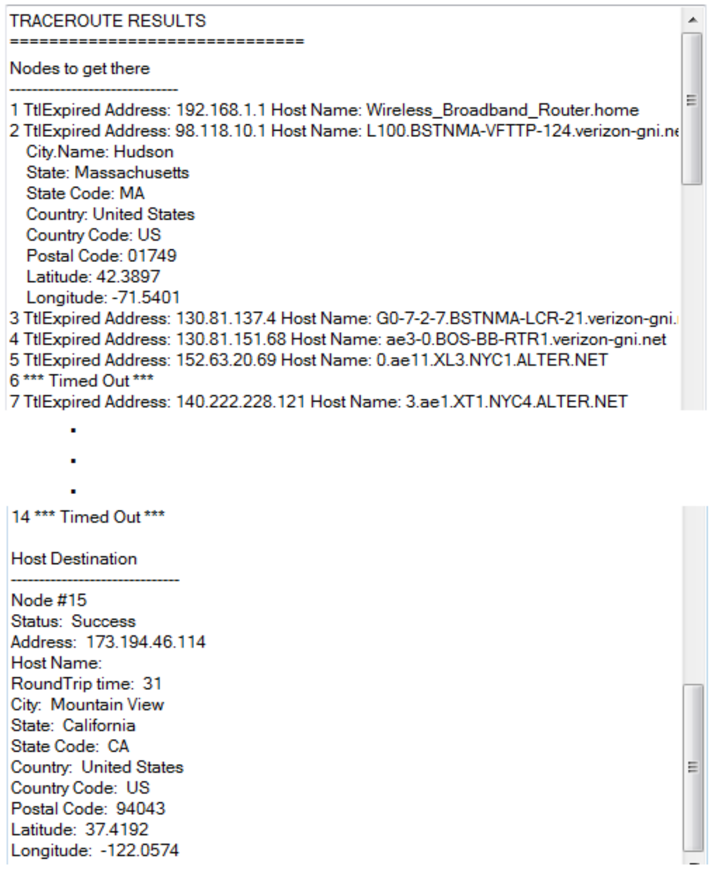
Example of ping results



```
PING RESULTS
-------------------------
Address: 173.194.46.115
RoundTrip time: 30
Time to live: 246
Don't fragment: False
Buffer size: 32
Status: Success

HOST LOCATION
-------------------------
City: Mountain View
State: California
State Code: CA
Country: United States
Country Code: US
Postal.Code: 94043
Latitude: 37.4192
Longitude: -122.0574
```

The Traceroute results display information for each node along the network path to the host destination. If available, geolocation and host name are displayed for each node and for the host system

Example of traceroute results



```
TRACEROUTE RESULTS
==============================
Nodes to get there
------------------------------
1 TtlExpired Address: 192.168.1.1 Host Name: Wireless_Broadband_Router.home
2 TtlExpired Address: 98.118.10.1 Host Name: L100.BSTNMA-VFTTP-124.verizon-gni.ne
   City.Name: Hudson
   State: Massachusetts
   State Code: MA
   Country: United States
   Country Code: US
   Postal Code: 01749
   Latitude: 42.3897
   Longitude: -71.5401
3 TtlExpired Address: 130.81.137.4 Host Name: G0-7-2-7.BSTNMA-LCR-21.verizon-gni.
4 TtlExpired Address: 130.81.151.68 Host Name: ae3-0.BOS-BB-RTR1.verizon-gni.net
5 TtlExpired Address: 152.63.20.69 Host Name: 0.ae11.XL3.NYC1.ALTER.NET
6 *** Timed Out ***
7 TtlExpired Address: 140.222.228.121 Host Name: 3.ae1.XT1.NYC4.ALTER.NET
        .
        .
        .
14 *** Timed Out ***

Host Destination
------------------------------
Node #15
Status:  Success
Address:  173.194.46.114
Host Name:
RoundTrip time:  31
City:  Mountain View
State:  California
State Code:  CA
Country:  United States
Country Code:  US
Postal Code:  94043
Latitude:  37.4192
Longitude:  -122.0574
```

Code Structure and APIs

This section describes how the code and APIs were organized and used in the program.

Ping Code Example

The program uses the C# Ping API to send ping requests to the remote host as shown in the following code. A timeout of 2 seconds (2000 msec) was used. This could possibly be set even lower to speed up the process, but it might miss some ping requests.

Since some web sites do not allow the ping command, the ping request was wrapped in a standard Try and Catch statements to catch any exceptions.

```
Ping pingSender = new Ping();
PingOptions options = new PingOptions();

// Use the default Ttl value which is 128,
// but change the fragmentation behavior.
options.DontFragment = true;

// Create a buffer of 32 bytes of data to be transmitted.
string data = "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa";
byte[] buffer = Encoding.ASCII.GetBytes(data);
int timeout = 2000;

// Try the ping command
PingReply reply;
try
{
    reply = pingSender.Send(host, timeout, buffer, options);
}
catch (Exception er)
{
    textBox1.Text = "Ping did not work\r\nException caught: " + er;
    return;
}
```

Traceroute Code Example

Since Visual Studio C# does not include a dedicated Traceroute command, you have to use the C# Ping API to create one. A number of examples were available online.

To use the C# ping API for traceroute, you set the ping's Time-to-Live (TTL) option at 1 and then increment the TTL value by one in increasing amounts until you reach the host destination. The TTL value controls how many "hops" the ping takes each time to try to get to the host.

The PingOptions class was the important part for doing a traceroute because it has an option for TTL.

```
// Set vars and options for ping command
Ping pingSender = new Ping();
PingOptions options = new PingOptions();

options.DontFragment = true;
options.Ttl = 1;

...

PingReply reply = pingSender.Send(host, timeout, buffer, options);
```

The reply status from the ping API lets you see the status of each node that you ping. A status of "TtlExpired" means you haven't reached the host yet, while a reply status of "Success" means you are at the host destination.

```
if (reply.Status == IPStatus.Success)
{
        ...
}
else if (reply.Status == IPStatus.TtlExpired)
{
        ...
}
else if (reply.Status == IPStatus.TimedOut)
{
...
}
```

For this program, a maximum value 32 hops was used for the TTL value to find the host. All of the example hosts that were tested with the program were reachable in less than 32 hops.

The overall pseudo code for using ping to do traceroute is shown in the following example:

```
for (int i = 1; i <= 32; i++)
{
    Run ping
    Do geolocation lookup
    Do DNS lookup to get host name
    Add markers and path to the map
    Check ping reply status
        if status = "success" we are at the host so exit loop
        if status = "TtlExpired" we are just at a node so continue looping
    Increment the TTL value to get to next node in path
}
Draw map
```

```
Display text results
```

Host Name Lookup Code Example

After you retrieve the IP address of an intermediate node or the host destination, you can use the IP address to lookup the related host name. This can be done with the C# DNS GetHostEntry() method, but it has a built-in timeout of 5 seconds that cannot be changed. To make this run faster, an alternative approach was used to run the DNS lookup asynchronously using the following code.

This approach provided a host name for each of the nodes along the network path that responded to the ping request.

```
// Try DNS lookup.
// NOTE: Runs asynchronously
string strHostLookup = "";
try
{
    strHostLookup = DoGetHostEntryAsync(reply.Address.ToString());
}
catch
{
    strHostLookup = "";
}


public string DoGetHostEntryAsync(string hostname)
{
    GetHostEntryFinished.Reset();
    ResolveState ioContext = new ResolveState(hostname);

    Dns.BeginGetHostEntry(ioContext.host,
        new AsyncCallback(GetHostEntryCallback), ioContext);

    // Wait here until the resolve completes (the callback
    // calls .Set())

    // Try waiting period of .25 sec to lookup host name
    GetHostEntryFinished.WaitOne(250, true);

    if (ioContext.IPs.HostName != null)
    {
        return ioContext.IPs.HostName;
    }
    else
    {
        return ("");
    }
}
```

GeoIP API Code Example

The MaxMind GeoIP API uses a local database to lookup the physical geographical location of a remote system based on a given IP address. The database does not contain records for all remote hosts, systems or equipment on the internet, but it does include enough to trace a network path that can be plotted on a map.

The GeoIP API provides a DatabaseReader object for accessing and querying the database. The database contains city, state, country, and latitude and longitude for a number of known IP addresses.

The following code example from the project shows how the GeoIP API uses the DatabaseReader object to lookup the geolocation information based on the IP address (`reply.Address`) returned by the ping API.

```
string geoIPText = "";
var reader = new DatabaseReader("GeoLite2-City.mmdb");

// GeoIP db reader
var city = reader.City(reply.Address.ToString());

// Assemble geolocation results
geoIPText = "\r\n\r\nHOST LOCATION";
geoIPText = geoIPText + "\r\n--------------------------";
geoIPText = geoIPText + "\r\nCity: " + city.City.Name; // 'Minneapolis'
geoIPText = geoIPText + "\r\nState: " + city.MostSpecificSubdivision.Name; // 'Minnesota'
geoIPText = geoIPText + "\r\nState Code: " + city.MostSpecificSubdivision.IsoCode; // 'MN'
geoIPText = geoIPText + "\r\nCountry: " + city.Country.Name; // 'United States'
geoIPText = geoIPText + "\r\nCountry Code: " + city.Country.IsoCode; // 'US'
geoIPText = geoIPText + "\r\nPostal.Code: " + city.Postal.Code; // '55455'
geoIPText = geoIPText + "\r\nLatitude: " + city.Location.Latitude; // 44.9733
geoIPText = geoIPText + "\r\nLongitude: " + city.Location.Longitude; // -93.2323
```

The assembled text string is then displayed in the program's results box. The latitude and longitude values can also be used to plot actual locations on a map with the Google Map API.

Google Maps API Code Example

The Google Static Maps API is a text-based URL API that can display a static image of a user-defined map in a web browser. When the user runs a ping or traceroute command, the program dynamically assembles a URL with the correct syntax and assigns it to the address of the embedded web browser control in the program's user interface.

The marker attributes display pushpin icons as locations on the map. The path attributes allow you to define and draw lines between points on the map.

```
// Create URL for map -----------------
string marker1 = "&markers=color:yellow%7Clabel:0%7C" + m_strMyLoc;
string marker2 = "&markers=color:yellow%7Clabel:1%7C" + lat2 + "," + lon2;
string path12 = "&path=color:0xff0000ff|weight:3|" + m_strMyLoc + "|" + lat2 + "," + lon2;

// Display map
webBrowserMap.Navigate("http://maps.googleapis.com/maps/api/staticmap?size=" + m_mapSize +
marker1 + marker2 + path12 + "&maptype=roadmap");
```

Putting it all Together

The final step was to combine the user interface with all of the behind-the-scenes code to make the program actually work. All of the supporting code to ping the remote host, lookup additional DNS and geolocation information was combined to then dynamically assemble the results into a Google map URL and a summarized set of text results.

1. User enters the zip code for their location.
2. User selects a host from the combo box or enters one into the textbox.
3. User clicks Ping or Traceroute button.
4. For the current host (or a node if doing a traceroute).
    a. Ping host (or node if doing a traceroute) using C# Ping API.
    b. Check ping reply status (“success”, “TtlExpired”, “TimedOut“ or other).
        i. Add info to text string
    c. Do geolocation lookup with MaxMind GeoIP API.
        i. Add info to text string
    d. Add location markers and path to the URL string for the Google map.
    e. Do DNS lookup to get host name.
        i. Add info to text string
    f. If doing a traceroute, then continue looping for all network nodes between user and host.
5. Display Google map in web browser control #1 using the assembled URL.
6. Display web page for the selected host in web browser control #2.
7. Display the assembled text string as results in the textbox.

Testing and Evaluation

This section explains how I tested the program to make sure it was working properly. Detailed test results and various test cases are included in the appendix.

Testing Strategy

The overall testing strategy was to run the program and compare the results to the DOS-based ping and traceroute commands that are available in the Windows command console.

Comparing the results to the DOS versions of the ping and traceroute commands also helped to troubleshoot the program during development.

Manual checking of geolocation information was done using the MaxMind GeoIP demo tool at https://www.maxmind.com/en/geoip_demo. This tool lets you enter an IP address and see the related geolocation information. The program uses the GeoIP API that offers similar information.

Monitoring of response time was not a primary concern.

Test Data

Test data included a collection of 25 different, individually selected hosts that allow ICMP commands such as ping requests. Since not all web sites allow the ping command, this involved a little bit of researching and discovery to find remote hosts and web sites that worked with the ping command.

To research and discover ping-compatible hosts, I searched on Google for ideas and then used the DOS ping command to quickly check them. Using this approach, I was able to find interesting web sites from around the world. Once the program was running reliably, it could be used to test hosts.

The resulting collection of example hosts includes different university, commercial, and government web sites from around the US and around the world such as Canada, Europe and Brazil.

For a complete list of all the example hosts, see “List of Example Hosts” in the Appendix.

Some examples from the final list included the following hosts:

- Local switch or router (192.168.1.1)

- Google (www.google.com)
- University of Copenhagen (ku.dk)
- University of Oxford (ox.ac.uk)
- University of Toronto (utoronto.ca)
- South Africa travel and tourism (www.southafrica.net)
- Brazil - Technological Institute of Aeronautics (www.ita.br)
- Cabinet Office, Government of Japan (www.cao.go.jp)

Some testing also included being aware of hosts that configure their web servers to not accept ICMP commands. Some examples include microsoft.com, amazon.com, and harvard.edu.

Test Cases

A number of different test cases were used during the development of the program. Some of these tests included the following cases:

- Compare results with Windows command console/DOS ping and traceroute commands.
- Ran the program with wired and wireless networks.
- Ran the program on different Windows computers.
- Ran the program without an internet connection.

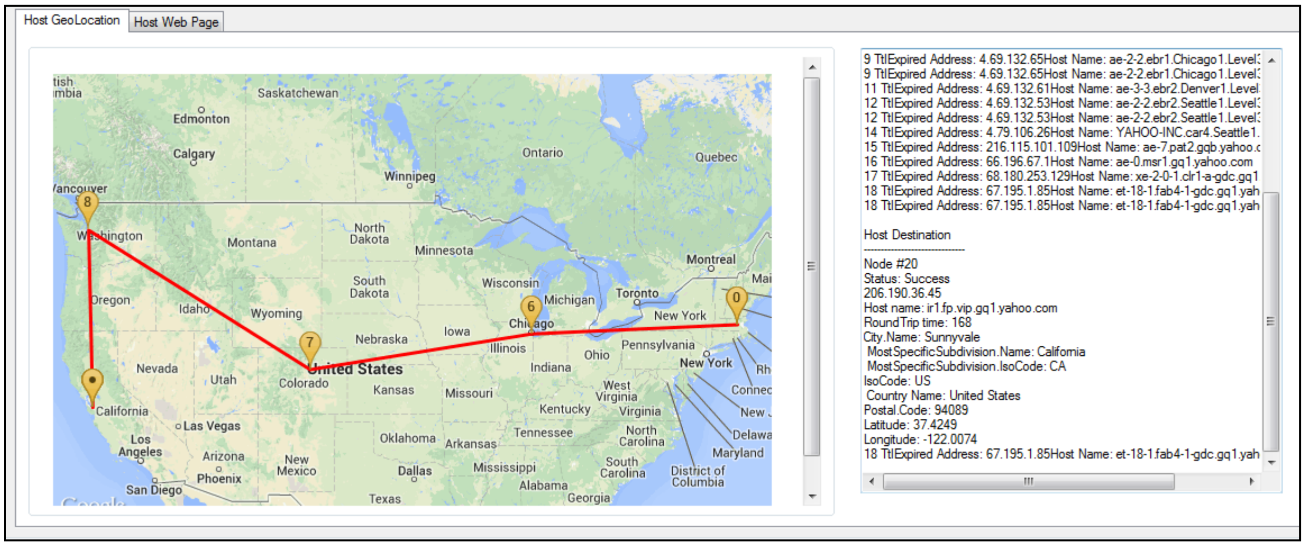
The complete list of test cases and their details are included in "Test Cases and Test Results" in the Appendix.

Interesting Results

Some interesting and curious results were achieved as I tested different hosts and test cases with the program. For screenshots of the network paths, see "Screenshots of Interesting Results" in the Appendix.

For some hosts, the network path seemed to repeat the same path every time. For other hosts, such as mit.edu, the network path was often a little different each time. However, some differences in the path might be associated with timeouts at certain nodes. Timing out at a node might cause the path to miss a valid latitude and longitude point in the path.

Here’s an interesting cross-country network path to Yahoo.com that came up only once or twice.

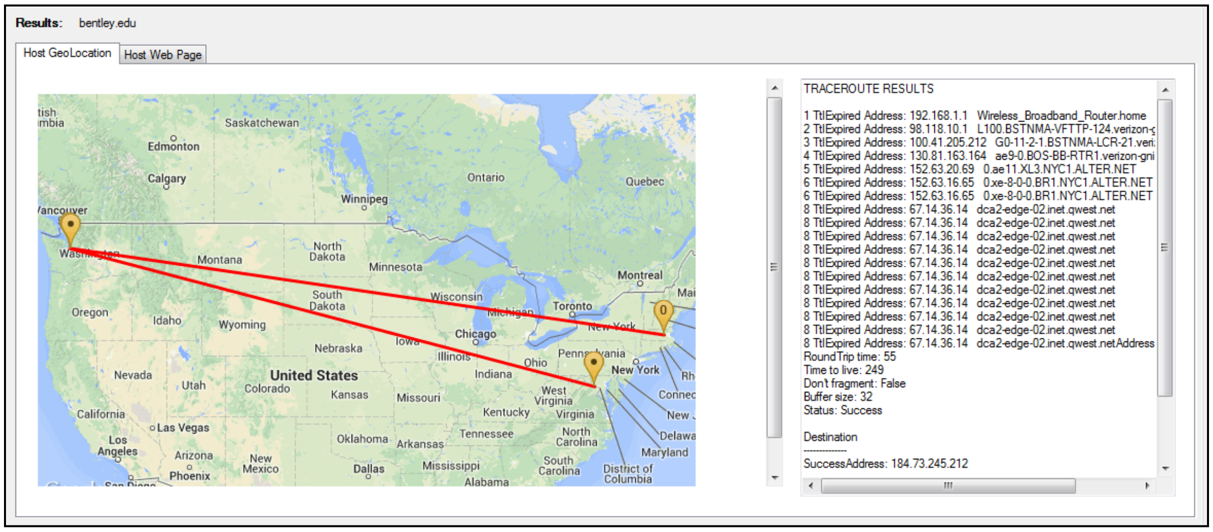


```
9 TtlExpired Address: 4.69.132.65Host Name: ae-2-2.ebr1.Chicago1.Level3
9 TtlExpired Address: 4.69.132.65Host Name: ae-2-2.ebr1.Chicago1.Level3
11 TtlExpired Address: 4.69.132.61Host Name: ae-3-3.ebr2.Denver1.Level
12 TtlExpired Address: 4.69.132.53Host Name: ae-2-2.ebr2.Seattle1.Level3
12 TtlExpired Address: 4.69.132.53Host Name: ae-2-2.ebr2.Seattle1.Level3
14 TtlExpired Address: 4.79.106.26Host Name: YAHOO-INC.car4.Seattle1.
15 TtlExpired Address: 216.115.101.109Host Name: ae-7.pat2.gqb.yahoo.c
16 TtlExpired Address: 66.196.67.1Host Name: ae-0.msr1.gq1.yahoo.com
17 TtlExpired Address: 68.180.253.129Host Name: xe-2-0-1.clr1-a-gdc.gq1
18 TtlExpired Address: 67.195.1.85Host Name: et-18-1.fab4-1-gdc.gq1.yah
18 TtlExpired Address: 67.195.1.85Host Name: et-18-1.fab4-1-gdc.gq1.yah

Host Destination
----------------------------
Node #20
Status: Success
206.190.36.45
Host name: ir1.fp.vip.gq1.yahoo.com
RoundTrip time: 168
City.Name: Sunnyvale
 MostSpecificSubdivision.Name: California
 MostSpecificSubdivision.IsoCode: CA
IsoCode: US
 Country Name: United States
Postal.Code: 94089
Latitude: 37.4249
Longitude: -122.0074
18 TtlExpired Address: 67.195.1.85Host Name: et-18-1.fab4-1-gdc.gq1.yah
```

Overall, some reoccurring patterns of network nodes were noticeable. For example, many of the network paths went through the same areas, such as Purchase, NY, Ashburn, VA, and Englewood, CO.

In some cases, the traceroute command allowed you to see how the host computers for some web sites are actually hosted in entirely different parts of the world. For example, some international web sites are mirrored on host systems here in the US. In other cases, the host for a local Boston-area university was actually located in another state.

For example, Bentley University is located in Waltham, MA, but a traceroute to its web site (bentley.edu) goes all the way across the country from Boston to Washington state back across the country to Ashburn, Virginia.



```
TRACEROUTE RESULTS

1 TtlExpired Address: 192.168.1.1   Wireless_Broadband_Router.home
2 TtlExpired Address: 98.118.10.1   L100.BSTNMA-VFTTP-124.verizon-g
3 TtlExpired Address: 100.41.205.212   G0-11-2-1.BSTNMA-LCR-21.veri
4 TtlExpired Address: 130.81.163.164   ae9-0.BOS-BB-RTR1.verizon-gni
5 TtlExpired Address: 152.63.20.69   0.ae11.XL3.NYC1.ALTER.NET
6 TtlExpired Address: 152.63.16.65   0.xe-8-0-0.BR1.NYC1.ALTER.NET
6 TtlExpired Address: 152.63.16.65   0.xe-8-0-0.BR1.NYC1.ALTER.NET
8 TtlExpired Address: 67.14.36.14   dca2-edge-02.inet.qwest.net
8 TtlExpired Address: 67.14.36.14   dca2-edge-02.inet.qwest.net
8 TtlExpired Address: 67.14.36.14   dca2-edge-02.inet.qwest.net
8 TtlExpired Address: 67.14.36.14   dca2-edge-02.inet.qwest.net
8 TtlExpired Address: 67.14.36.14   dca2-edge-02.inet.qwest.net
8 TtlExpired Address: 67.14.36.14   dca2-edge-02.inet.qwest.net
8 TtlExpired Address: 67.14.36.14   dca2-edge-02.inet.qwest.net
8 TtlExpired Address: 67.14.36.14   dca2-edge-02.inet.qwest.net
8 TtlExpired Address: 67.14.36.14   dca2-edge-02.inet.qwest.net
8 TtlExpired Address: 67.14.36.14   dca2-edge-02.inet.qwest.net
8 TtlExpired Address: 67.14.36.14   dca2-edge-02.inet.qwest.netAddress
RoundTrip time: 55
Time to live: 249
Don't fragment: False
Buffer size: 32
Status: Success

Destination
--------------
SuccessAddress: 184.73.245.212
```

Future Development

For the most, the program met the goals that we had: to create a program that provided an enhanced visual interface and map-based presentation for the information retrieved by the ping and traceroute networking commands. However, a number of improvements and areas to research remain for future updates to the program. These future development ideas include the following items:

- Have the program auto-detect the user's location.
- Try testing with different parameter values. For example a lower timeout level for ping.
- Research the other features and properties of C# ping command.
- Add more user options for ping and traceroute, such as maximum number of hops for TTL.
- Use an interactive map instead of a static map with the ability to pan and zoom the map.
- Add more maps to show detail of areas with concentrated network nodes.
- Find more interesting hosts to ping, especially global web sites and hosts.
- Improve the text display into a rich HTML format inside of an embedded web browser control.
- For traceroute, add a timer to monitor elapsed time for the response from each node. The ping API only returns a roundtrip time for a host and not for intermediary nodes when using the TTL option.
- Improve the logic for doing a traceroute on a host that doesn't accept ping requests. Right now the program goes through all 32 TTL steps even though the complete path is not available.

Conclusion – Solution Summary

This report has described the successful design and development of Visual Network Explorer - a program which provides an enhanced visual interface and presentation for the information retrieved by the ping and traceroute networking commands. Using the textural information retrieved by the standard ping and traceroute network commands, we were able to visually map the geolocation points of remote hosts and the network paths to reach them.

The combination of the C# Ping API with the MaxMind GeoIP and Google Maps APIs made this program possible. The MaxMind GeoIP API and related database enabled the program to look up the physical geographical location of host and node systems based on their IP address. The Google Maps API was able to draw location markers and network paths on local and global maps. The DNS lookup function allowed us to find host names for hosts, nodes and even home network equipment and computers.

The test data for the project played an important part in the development of the program. Since some internet hosts do not accept ping requests, it was critical to have a pre-determined list of known hosts to use as examples and for testing the program.

The test data also revealed some interesting results for network paths and geolocations of hosts and nodes. For some hosts, the network path seemed to repeat the same path every time. For other hosts, the network path was often a little different each time. The traceroute also showed that the host computers for some web sites are actually hosted in entirely different parts of the world. For example, the traceroute for a local Boston-area university (bentley.edu) traveled all the way across the country to Seattle, Washington, and then back to the east coast to the destination host in Virginia.

Patterns also started to appear from the test data. Many of the network paths went through the same areas, such as Purchase, NY, and Ashburn, VA.

Combining all this network and geolocation information with maps in an easy-to-use Windows user interface established a successful foundation for future work.

Appendices

The following resources are included in this section: References, Test Cases and Results, and Relevant Code.

References

The following resources and references were used while creating this project.

C# Ping API

http://msdn.microsoft.com/en-us/library/system.net.networkinformation.ping%28v=vs.110%29.aspx

C# IPStatus codes

http://msdn.microsoft.com/query/dev10.query?appId=Dev10IDEF1&l=EN-US&k=k%28SYSTEM.NET.NETWORKINFORMATION.IPSTATUS%29

MaxMind GeoIP databases

http://dev.maxmind.com/geoip/geoip2/downloadable/#MaxMind_APIs

Google Static Maps API V2 Developer Guide

https://developers.google.com/maps/documentation/staticmaps/

Ping (networking utility)

http://en.wikipedia.org/wiki/Ping_%28networking_utility%29

traceroute

http://en.wikipedia.org/wiki/Traceroute

Internet Control Message Protocol

http://en.wikipedia.org/wiki/Internet_Control_Message_Protocol

Test Cases and Test Results

This section lists all of the detailed test results and various test cases for the program.

Test Case	Result
No internet connection • Turn off wireless • Disconnect network cable	Program displays the following results: Ping did not work Exception caught: System.Net.NetworkInformation.PingException: ...
Invalid host/server name • Enter an invalid name in the host textbox • Run ping, then run traceroute	Program displays the following results: Ping did not work Exception caught: System.Net.NetworkInformation.PingException: ...
Try to ping a remote server that does not accept ping requests • Enter one of the following sites ○ microsoft.com ○ amazon.com ○ harvard.edu ○ ebay.com ○ cnn.com • Run ping, then run traceroute	Ping displays: Ping did not work As currently coded, the Traceroute process actually tries to find the path and goes through all 32 TTL steps. Note: Need to address this in future version to detect this scenario.
Test with valid server that accepts Ping/ICMP commands. • Select or enter a valid host that accepts ping requests. • Run ping, then run traceroute.	This was done repeatedly for the list of 25 example hosts. At times, different network paths are reported back for the same host when running traceroute.
Try ping and traceroute with typical local router address of 192.168.1.1.	Adjusted code to handle this. Ping and traceroute both return results. RoundTrip time is usually 0 to 1 milliseconds. DNS lookup returned the name of the router. Host Name: Wireless_Broadband_Router.home
Try ping and traceroute with other home network addresses (192.168.1.x.)	Ping and traceroute both return results. RoundTrip time is usually 0. DNS lookup returned the name of the computer.

Test Case	Result
Compare results with the DOS versions of ping and traceroute.	Results were often a little different, but in other cases nearly or exactly the same. Comparing the results helped to troubleshoot the program during development.

List of Example Hosts

The program was tested with the following example hosts. All of these hosts accept ping requests.

- Local switch or router (192.168.1.1)
- Google (www.google.com)
- Yahoo (www.yahoo.com)
- Massachusetts Institute of Technology MIT (mit.edu)
- Bentley University (bentley.edu)
- University of Maine (umaine.edu)
- University of Vermont (uvm.edu)
- University of Colorado (colorado.edu)
- Stanford University (stanford.edu)
- Texas A&M University (tamu.edu)
- University of Florida (ufl.edu)
- University of Chicago (uchicago.edu)
- University of Copenhagen (ku.dk)
- University of Oxford (ox.ac.uk)
- University of Cambridge (cam.ac.uk)
- Canadian Tire (www.canadiantire.ca)
- University of Toronto (utoronto.ca)
- University of Manitoba (umanitoba.ca)
- University of British Columbia (www.ubc.ca)
- BBC (bbc.com)"
- South Africa travel and tourism (www.southafrica.net)

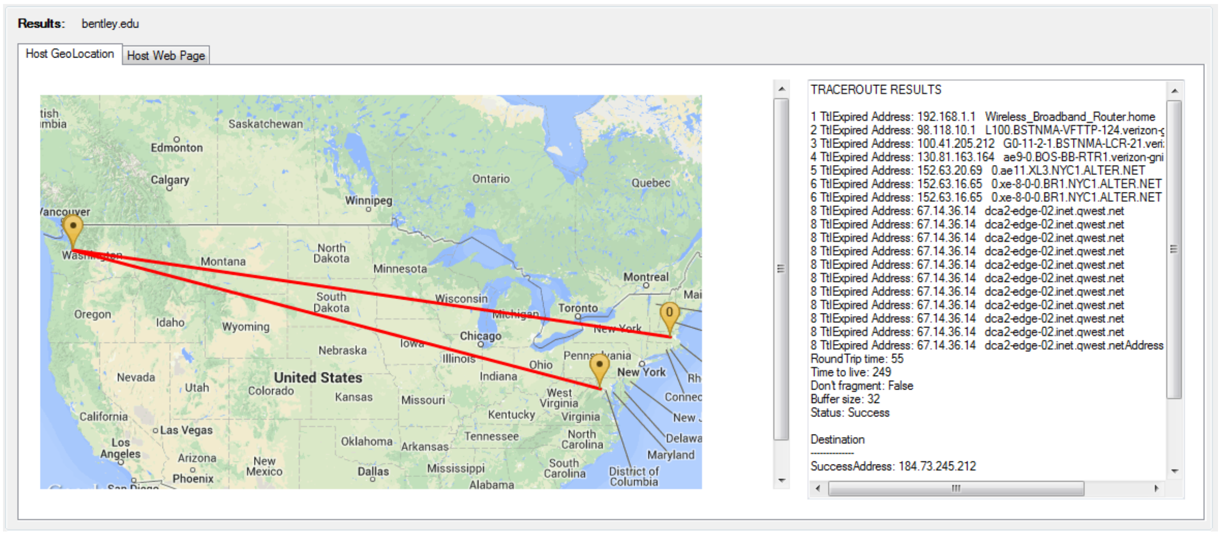
- South Africa information gateway (www.southafrica.info)
- Brazil - Technological Institute of Aeronautics (www.ita.br)
- NHK Online - Japanese news (www3.nhk.or.jp)
- NHK Online - Japanese news (www.nhk.or.jp)
- Cabinet Office, Government of Japan (www.cao.go.jp)

Screenshots of Interesting Results

This section includes a collection screenshots of interesting ping and traceroute results for different hosts around the world.

Bentley University

Bentley University is located in Waltham, MA, but its web site (bentley.edu) is hosted at .compute-1.amazonaws.com and goes all the way across the country from Boston to Washington state and then back across the country again to Ashburn, Virginia.

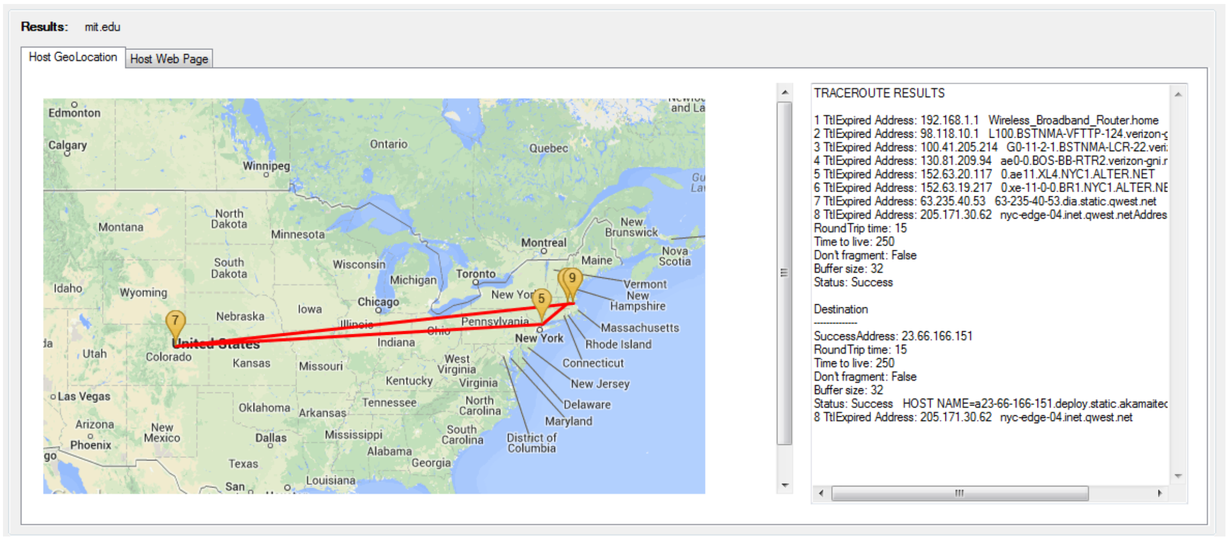


```
TRACEROUTE RESULTS

1 TtlExpired Address: 192.168.1.1   Wireless_Broadband_Router.home
2 TtlExpired Address: 98.118.10.1   L100.BSTNMA-VFTTP-124.verizon-
3 TtlExpired Address: 100.41.205.212   G0-11-2-1.BSTNMA-LCR-21.veri
4 TtlExpired Address: 130.81.163.164   ae9-0.BOS-BB-RTR1.verizon-gni
5 TtlExpired Address: 152.63.20.69   0.ae11.XL3.NYC1.ALTER.NET
6 TtlExpired Address: 152.63.16.65   0.xe-8-0-0.BR1.NYC1.ALTER.NET
6 TtlExpired Address: 152.63.16.65   0.xe-8-0-0.BR1.NYC1.ALTER.NET
8 TtlExpired Address: 67.14.36.14   dca2-edge-02.inet.qwest.net
8 TtlExpired Address: 67.14.36.14   dca2-edge-02.inet.qwest.net
8 TtlExpired Address: 67.14.36.14   dca2-edge-02.inet.qwest.net
8 TtlExpired Address: 67.14.36.14   dca2-edge-02.inet.qwest.net
8 TtlExpired Address: 67.14.36.14   dca2-edge-02.inet.qwest.net
8 TtlExpired Address: 67.14.36.14   dca2-edge-02.inet.qwest.net
8 TtlExpired Address: 67.14.36.14   dca2-edge-02.inet.qwest.net
8 TtlExpired Address: 67.14.36.14   dca2-edge-02.inet.qwest.net
8 TtlExpired Address: 67.14.36.14   dca2-edge-02.inet.qwest.net
8 TtlExpired Address: 67.14.36.14   dca2-edge-02.inet.qwest.net
8 TtlExpired Address: 67.14.36.14   dca2-edge-02.inet.qwest.netAddress
RoundTrip time: 55
Time to live: 249
Don't fragment: False
Buffer size: 32
Status: Success

Destination
--------------
SuccessAddress: 184.73.245.212
```

MIT.edu

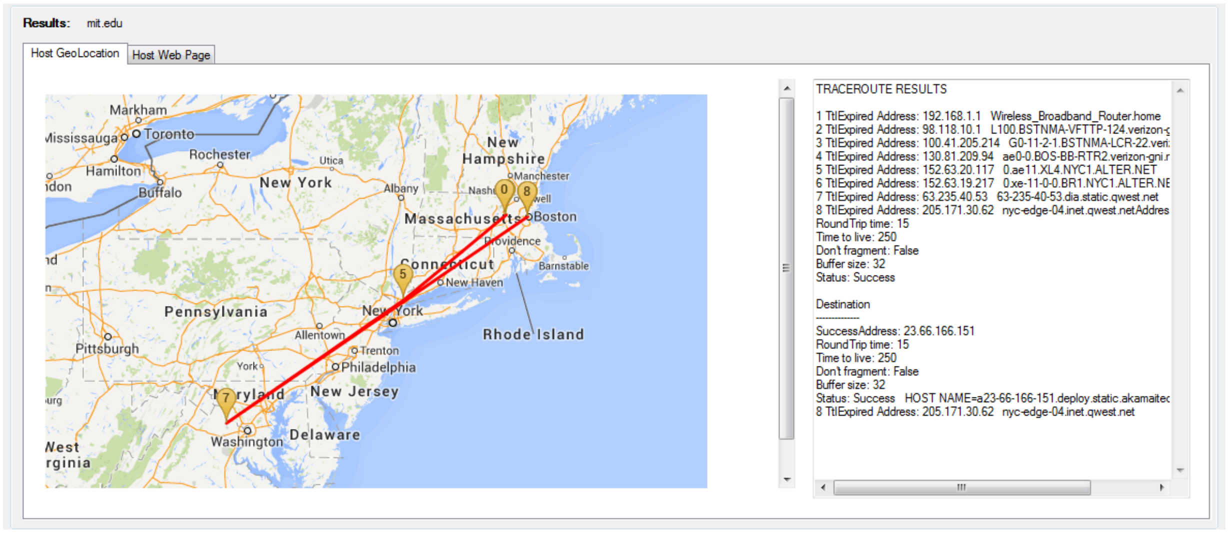
Running traceroute multiple times for MIT.edu resulted in different network paths as shown in the following screenshots.

MIT.edu network path #1



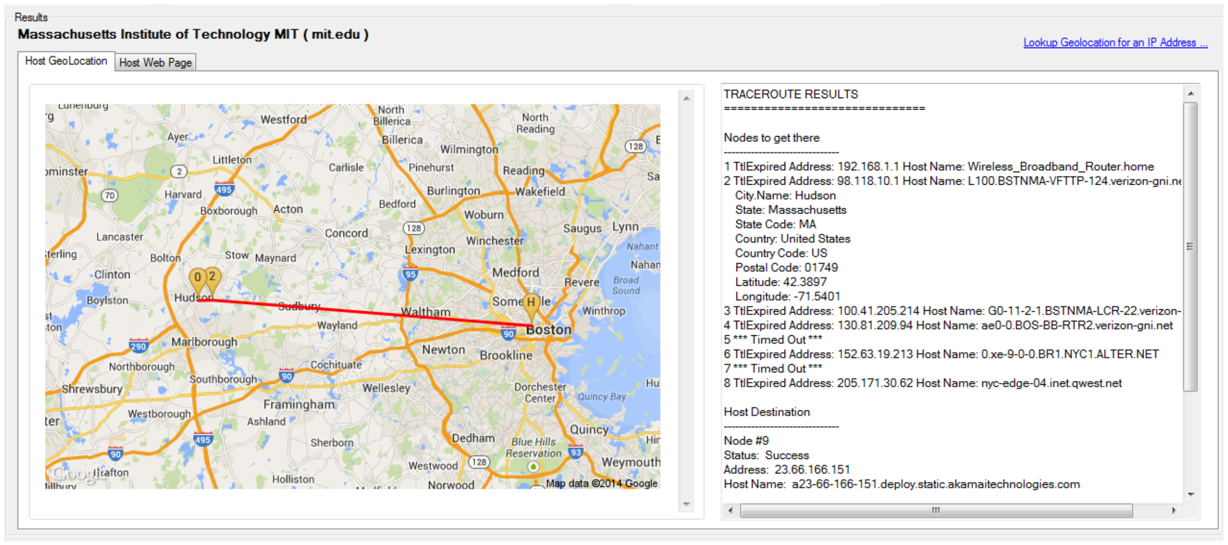
```
TRACEROUTE RESULTS

1 TtlExpired Address: 192.168.1.1   Wireless_Broadband_Router.home
2 TtlExpired Address: 98.118.10.1   L100.BSTNMA-VFTTP-124.verizon-
3 TtlExpired Address: 100.41.205.214   G0-11-2-1.BSTNMA-LCR-22.veri
4 TtlExpired Address: 130.81.209.94   ae0-0.BOS-BB-RTR2.verizon-gni.r
5 TtlExpired Address: 152.63.20.117   0.ae11.XL4.NYC1.ALTER.NET
6 TtlExpired Address: 152.63.19.217   0.xe-11-0-0.BR1.NYC1.ALTER.NE
7 TtlExpired Address: 63.235.40.53   63-235-40-53.dia.static.qwest.net
8 TtlExpired Address: 205.171.30.62   nyc-edge-04.inet.qwest.netAddres
RoundTrip time: 15
Time to live: 250
Don't fragment: False
Buffer size: 32
Status: Success

Destination
--------------
SuccessAddress: 23.66.166.151
RoundTrip time: 15
Time to live: 250
Don't fragment: False
Buffer size: 32
Status: Success   HOST NAME=a23-66-166-151.deploy.static.akamaitec
8 TtlExpired Address: 205.171.30.62   nyc-edge-04.inet.qwest.net
```

MIT.edu network path #2



```
TRACEROUTE RESULTS

1 TtlExpired Address: 192.168.1.1   Wireless_Broadband_Router.home
2 TtlExpired Address: 98.118.10.1   L100.BSTNMA-VFTTP-124.verizon-
3 TtlExpired Address: 100.41.205.214   G0-11-2-1.BSTNMA-LCR-22.veri
4 TtlExpired Address: 130.81.209.94   ae0-0.BOS-BB-RTR2.verizon-gni.r
5 TtlExpired Address: 152.63.20.117   0.ae11.XL4.NYC1.ALTER.NET
6 TtlExpired Address: 152.63.19.217   0.xe-11-0-0.BR1.NYC1.ALTER.NE
7 TtlExpired Address: 63.235.40.53   63-235-40-53.dia.static.qwest.net
8 TtlExpired Address: 205.171.30.62   nyc-edge-04.inet.qwest.netAddres
RoundTrip time: 15
Time to live: 250
Don't fragment: False
Buffer size: 32
Status: Success

Destination
--------------
SuccessAddress: 23.66.166.151
RoundTrip time: 15
Time to live: 250
Don't fragment: False
Buffer size: 32
Status: Success   HOST NAME=a23-66-166-151.deploy.static.akamaitec
8 TtlExpired Address: 205.171.30.62   nyc-edge-04.inet.qwest.net
```

MIT.edu network path #3


Results
Massachusetts Institute of Technology MIT (mit.edu)

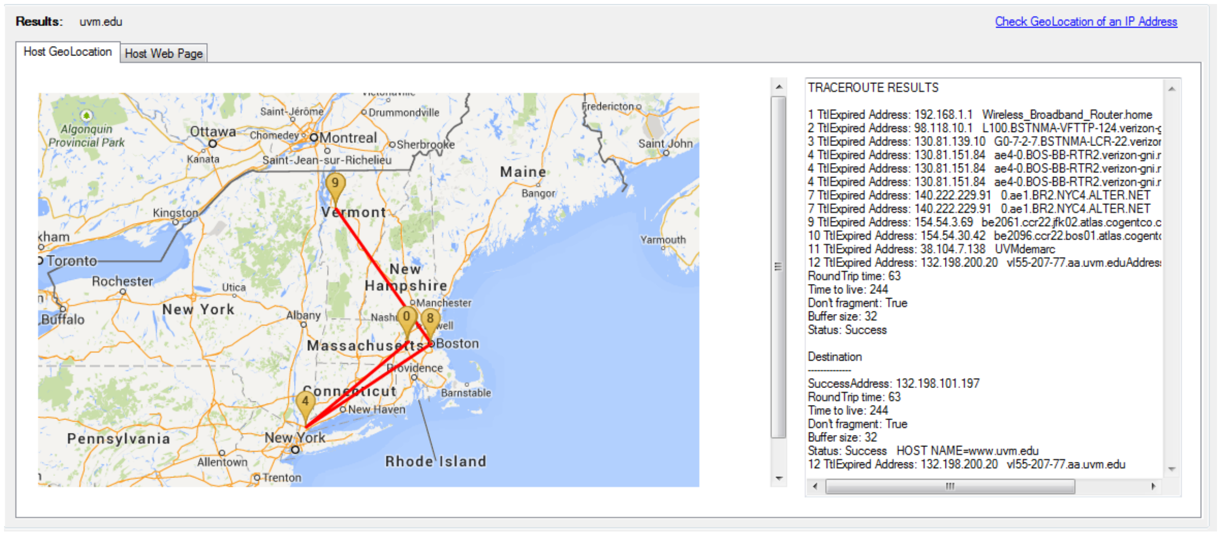
Host GeoLocation | Host Web Page

Lookup Geolocation for an IP Address ...

```
TRACEROUTE RESULTS
================================

Nodes to get there
------------------------------
1 TtlExpired Address: 192.168.1.1 Host Name: Wireless_Broadband_Router.home
2 TtlExpired Address: 98.118.10.1 Host Name: L100.BSTNMA-VFTTP-124.verizon-gni.ne
   City.Name: Hudson
   State: Massachusetts
   State Code: MA
   Country: United States
   Country Code: US
   Postal Code: 01749
   Latitude: 42.3897
   Longitude: -71.5401
3 TtlExpired Address: 100.41.205.214 Host Name: G0-11-2-1.BSTNMA-LCR-22.verizon-
4 TtlExpired Address: 130.81.209.94 Host Name: ae0-0.BOS-BB-RTR2.verizon-gni.net
5 *** Timed Out ***
6 TtlExpired Address: 152.63.19.213 Host Name: 0.xe-9-0-0.BR1.NYC1.ALTER.NET
7 *** Timed Out ***
8 TtlExpired Address: 205.171.30.62 Host Name: nyc-edge-04.inet.qwest.net

Host Destination
------------------------------
Node #9
Status:  Success
Address:  23.66.166.151
Host Name:  a23-66-166-151.deploy.static.akamaitechnologies.com
```

University of Vermont

The network path from Hudson, MA to the following three universities goes first south to Purchase, New York, then back north to their respective hosts in Vermont, Maine, or Toronto.

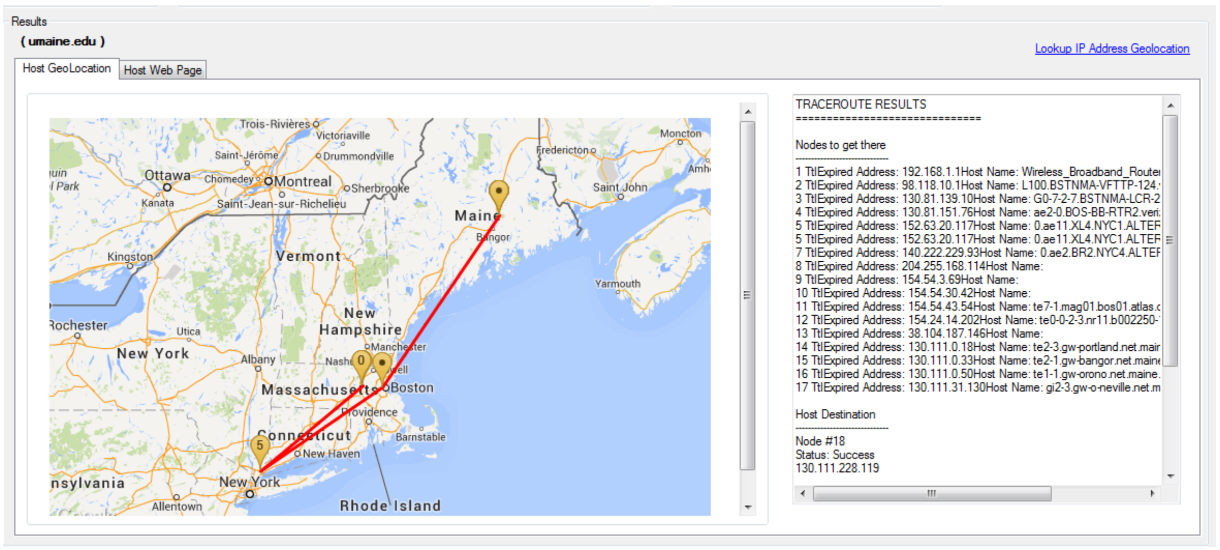

Results: uvm.edu

Host GeoLocation | Host Web Page

Check GeoLocation of an IP Address

```
TRACEROUTE RESULTS

1 TtlExpired Address: 192.168.1.1   Wireless_Broadband_Router.home
2 TtlExpired Address: 98.118.10.1   L100.BSTNMA-VFTTP-124.verizon-g
3 TtlExpired Address: 130.81.139.10   G0-7-2-7.BSTNMA-LCR-22.verizor
4 TtlExpired Address: 130.81.151.84   ae4-0.BOS-BB-RTR2.verizon-gni.r
4 TtlExpired Address: 130.81.151.84   ae4-0.BOS-BB-RTR2.verizon-gni.r
4 TtlExpired Address: 130.81.151.84   ae4-0.BOS-BB-RTR2.verizon-gni.r
7 TtlExpired Address: 140.222.229.91   0.ae1.BR2.NYC4.ALTER.NET
7 TtlExpired Address: 140.222.229.91   0.ae1.BR2.NYC4.ALTER.NET
9 TtlExpired Address: 154.54.3.69   be2061.ccr22.jfk02.atlas.cogentco.c
10 TtlExpired Address: 154.54.30.42   be2096.ccr22.bos01.atlas.cogent
11 TtlExpired Address: 38.104.7.138   UVMdemarc
12 TtlExpired Address: 132.198.200.20   vl55-207-77.aa.uvm.eduAddres
RoundTrip time: 63
Time to live: 244
Don't fragment: True
Buffer size: 32
Status: Success

Destination
--------------
SuccessAddress: 132.198.101.197
RoundTrip time: 63
Time to live: 244
Don't fragment: True
Buffer size: 32
Status: Success   HOST NAME=www.uvm.edu
12 TtlExpired Address: 132.198.200.20   vl55-207-77.aa.uvm.edu
```

University of Maine



Results

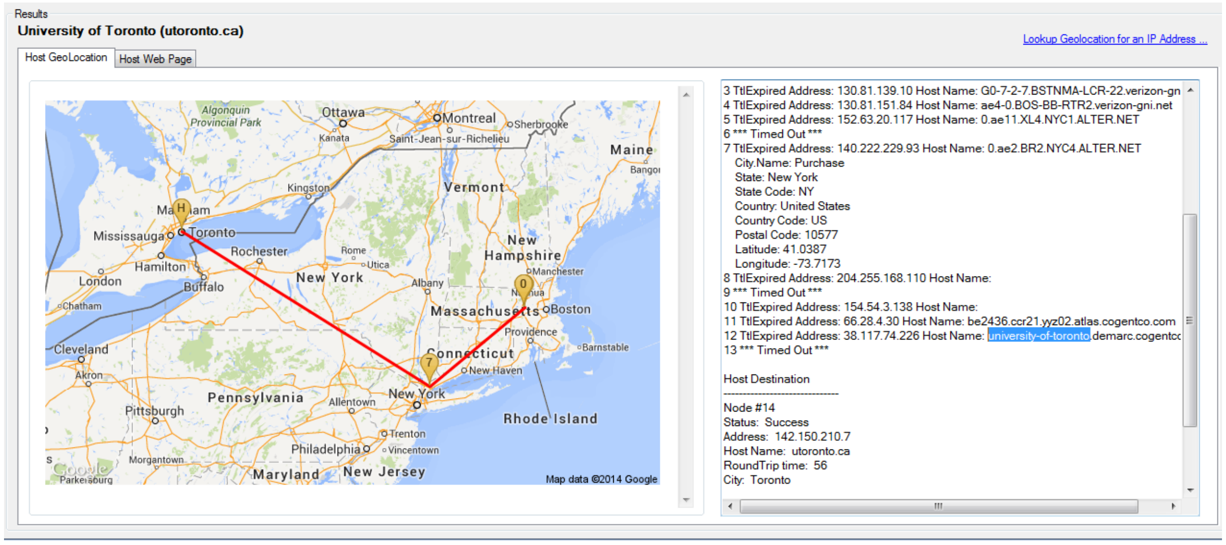
(umaine.edu)

Lookup IP Address Geolocation

Host GeoLocation | Host Web Page

```
TRACEROUTE RESULTS
================================

Nodes to get there
--------------------------------
1 TtlExpired Address: 192.168.1.1Host Name: Wireless_Broadband_Router
2 TtlExpired Address: 98.118.10.1Host Name: L100.BSTNMA-VFTTP-124.
3 TtlExpired Address: 130.81.139.10Host Name: G0-7-2-7.BSTNMA-LCR-2
4 TtlExpired Address: 130.81.151.76Host Name: ae2-0.BOS-BB-RTR2.veri
5 TtlExpired Address: 152.63.20.117Host Name: 0.ae11.XL4.NYC1.ALTEF
5 TtlExpired Address: 152.63.20.117Host Name: 0.ae11.XL4.NYC1.ALTEF
7 TtlExpired Address: 140.222.229.93Host Name: 0.ae2.BR2.NYC4.ALTEF
8 TtlExpired Address: 204.255.168.114Host Name:
9 TtlExpired Address: 154.54.3.69Host Name:
10 TtlExpired Address: 154.54.30.42Host Name:
11 TtlExpired Address: 154.54.43.54Host Name: te7-1.mag01.bos01.atlas.c
12 TtlExpired Address: 154.24.14.202Host Name: te0-0-2-3.nr11.b002250-
13 TtlExpired Address: 38.104.187.146Host Name:
14 TtlExpired Address: 130.111.0.18Host Name: te2-3.gw-portland.net.mair
15 TtlExpired Address: 130.111.0.33Host Name: te2-1.gw-bangor.net.maine
16 TtlExpired Address: 130.111.0.50Host Name: te1-1.gw-orono.net.maine.
17 TtlExpired Address: 130.111.31.130Host Name: gi2-3.gw-o-neville.net.m

Host Destination
--------------------------------
Node #18
Status: Success
130.111.228.119
```

University-of-Toronto



Results

University of Toronto (utoronto.ca)

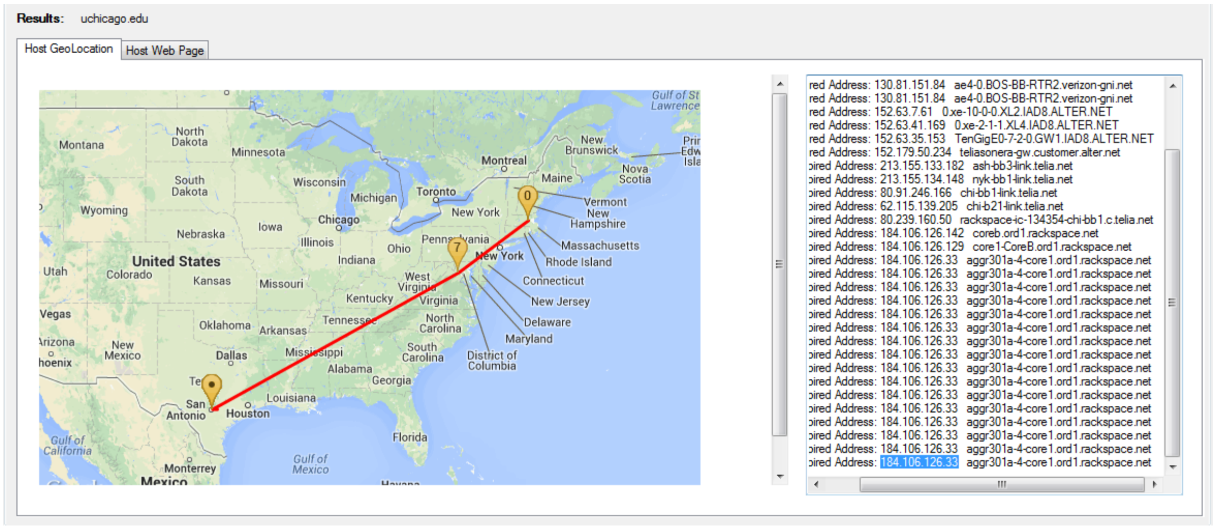
Lookup Geolocation for an IP Address ...

Host GeoLocation | Host Web Page

```
3 TtlExpired Address: 130.81.139.10 Host Name: G0-7-2-7.BSTNMA-LCR-22.verizon-gn
4 TtlExpired Address: 130.81.151.84 Host Name: ae4-0.BOS-BB-RTR2.verizon-gni.net
5 TtlExpired Address: 152.63.20.117 Host Name: 0.ae11.XL4.NYC1.ALTER.NET
6 *** Timed Out ***
7 TtlExpired Address: 140.222.229.93 Host Name: 0.ae2.BR2.NYC4.ALTER.NET
   City.Name: Purchase
   State: New York
   State Code: NY
   Country: United States
   Country Code: US
   Postal Code: 10577
   Latitude: 41.0387
   Longitude: -73.7173
8 TtlExpired Address: 204.255.168.110 Host Name:
9 *** Timed Out ***
10 TtlExpired Address: 154.54.3.138 Host Name:
11 TtlExpired Address: 66.28.4.30 Host Name: be2436.ccr21.yyz02.atlas.cogentco.com
12 TtlExpired Address: 38.117.74.226 Host Name: university-of-toronto.demarc.cogentcc
13 *** Timed Out ***

Host Destination
--------------------------------
Node #14
Status:  Success
Address:  142.150.210.7
Host Name:  utoronto.ca
RoundTrip time:  56
City:  Toronto
```

University of Chicago

The University of Chicago is hosted by Rackspace Hosting at 184.106.126.33 and actually located in San Antonio,Texas.

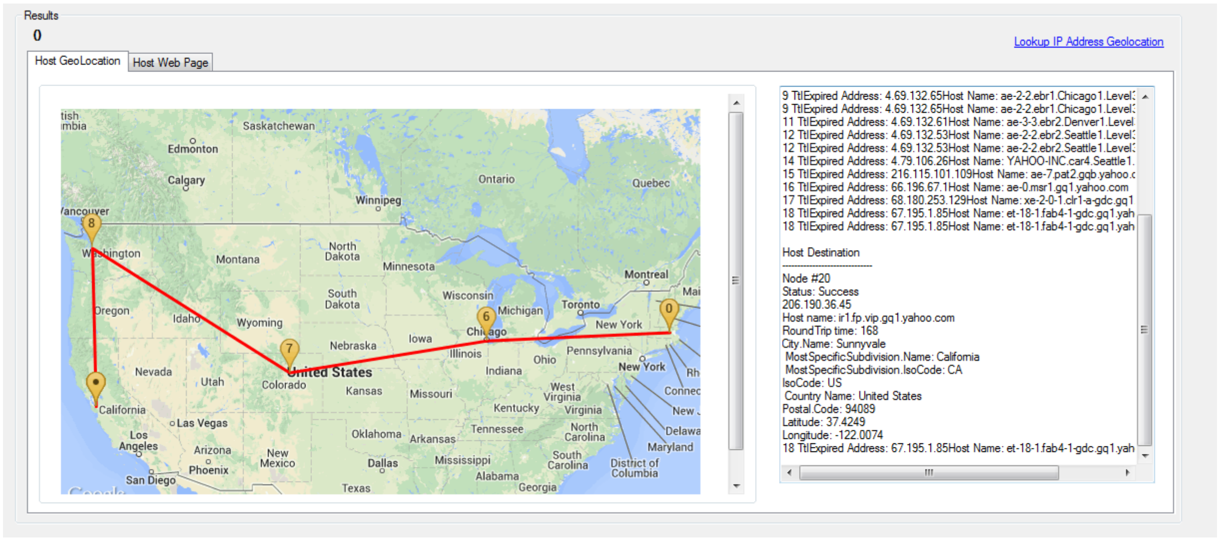


Results: uchicago.edu

```
red Address: 130.81.151.84   ae4-0.BOS-BB-RTR2.verizon-gni.net
red Address: 130.81.151.84   ae4-0.BOS-BB-RTR2.verizon-gni.net
red Address: 152.63.7.61   0.xe-10-0-0.XL2.IAD8.ALTER.NET
red Address: 152.63.41.169   0.xe-2-1-1.XL4.IAD8.ALTER.NET
red Address: 152.63.35.153   TenGigE0-7-2-0.GW1.IAD8.ALTER.NET
red Address: 152.179.50.234   teliasonera-gw.customer.alter.net
pired Address: 213.155.133.182   ash-bb3-link.telia.net
pired Address: 213.155.134.148   nyk-bb1-link.telia.net
pired Address: 80.91.246.166   chi-bb1-link.telia.net
pired Address: 62.115.139.205   chi-b21-link.telia.net
pired Address: 80.239.160.50   rackspace-ic-134354-chi-bb1.c.telia.net
pired Address: 184.106.126.142   coreb.ord1.rackspace.net
pired Address: 184.106.126.129   core1-CoreB.ord1.rackspace.net
pired Address: 184.106.126.33   aggr301a-4-core1.ord1.rackspace.net
pired Address: 184.106.126.33   aggr301a-4-core1.ord1.rackspace.net
pired Address: 184.106.126.33   aggr301a-4-core1.ord1.rackspace.net
pired Address: 184.106.126.33   aggr301a-4-core1.ord1.rackspace.net
pired Address: 184.106.126.33   aggr301a-4-core1.ord1.rackspace.net
pired Address: 184.106.126.33   aggr301a-4-core1.ord1.rackspace.net
pired Address: 184.106.126.33   aggr301a-4-core1.ord1.rackspace.net
pired Address: 184.106.126.33   aggr301a-4-core1.ord1.rackspace.net
pired Address: 184.106.126.33   aggr301a-4-core1.ord1.rackspace.net
pired Address: 184.106.126.33   aggr301a-4-core1.ord1.rackspace.net
pired Address: 184.106.126.33   aggr301a-4-core1.ord1.rackspace.net
pired Address: 184.106.126.33   aggr301a-4-core1.ord1.rackspace.net
pired Address: 184.106.126.33   aggr301a-4-core1.ord1.rackspace.net
pired Address: 184.106.126.33   aggr301a-4-core1.ord1.rackspace.net
pired Address: 184.106.126.33   aggr301a-4-core1.ord1.rackspace.net
pired Address: 184.106.126.33   aggr301a-4-core1.ord1.rackspace.net
```

Yahoo.com:

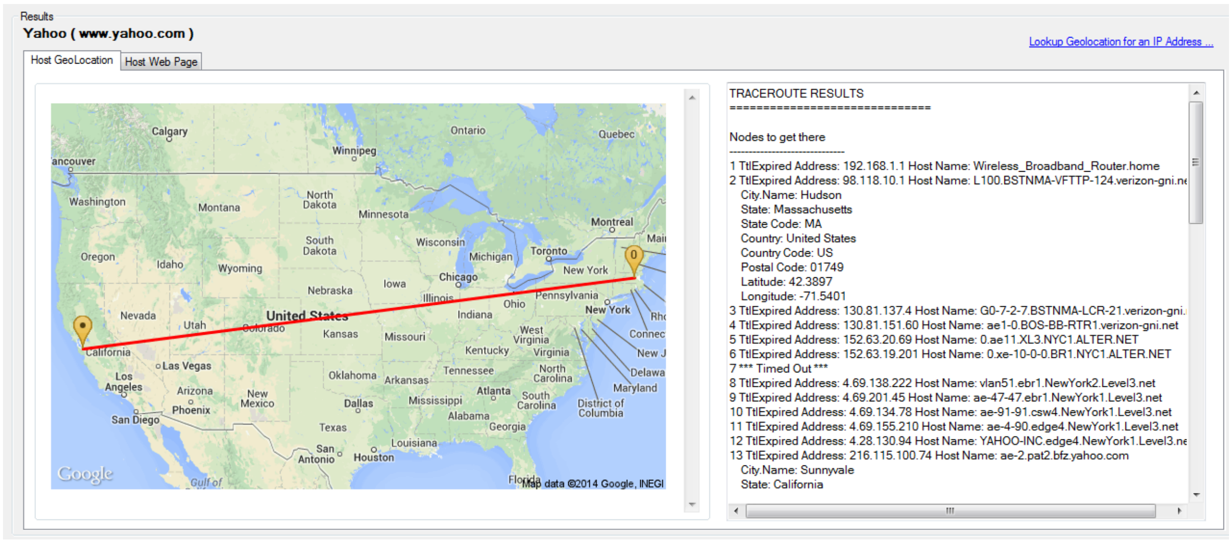
This interesting cross-country network path to Yahoo.com only appeared one or two times during testing. The rest of the time, the traceroute for Yahoo.com goes straight from Boston to California.



Results
0

```
9 TtlExpired Address: 4.69.132.65Host Name: ae-2-2.ebr1.Chicago1.Level3
9 TtlExpired Address: 4.69.132.65Host Name: ae-2-2.ebr1.Chicago1.Level3
11 TtlExpired Address: 4.69.132.61Host Name: ae-3-3.ebr2.Denver1.Level
12 TtlExpired Address: 4.69.132.53Host Name: ae-2-2.ebr2.Seattle1.Level3
12 TtlExpired Address: 4.69.132.53Host Name: ae-2-2.ebr2.Seattle1.Level3
14 TtlExpired Address: 4.79.106.26Host Name: YAHOO-INC.car4.Seattle1.
15 TtlExpired Address: 216.115.101.109Host Name: ae-7.pat2.gqb.yahoo.c
16 TtlExpired Address: 66.196.67.1Host Name: ae-0.msr1.gq1.yahoo.com
17 TtlExpired Address: 68.180.253.129Host Name: xe-2-0-1.clr1-a-gdc.gq1
18 TtlExpired Address: 67.195.1.85Host Name: et-18-1.fab4-1-gdc.gq1.yah
18 TtlExpired Address: 67.195.1.85Host Name: et-18-1.fab4-1-gdc.gq1.yah

Host Destination
----------------------------
Node #20
Status: Success
206.190.36.45
Host name: ir1.fp.vip.gq1.yahoo.com
RoundTrip time: 168
City.Name: Sunnyvale
 MostSpecificSubdivision.Name: California
 MostSpecificSubdivision.IsoCode: CA
IsoCode: US
 Country Name: United States
Postal.Code: 94089
Latitude: 37.4249
Longitude: -122.0074
18 TtlExpired Address: 67.195.1.85Host Name: et-18-1.fab4-1-gdc.gq1.yah
```

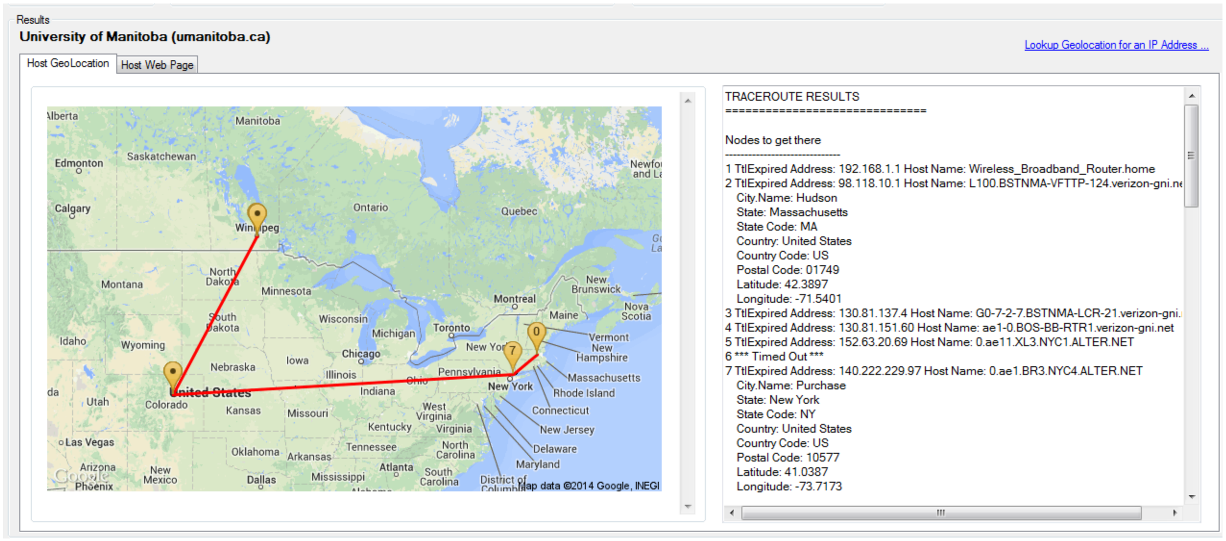
Typical network path for Yahoo.com


Results
Yahoo (www.yahoo.com)
Host GeoLocation
Host Web Page
Lookup Geolocation for an IP Address ...
Map data ©2014 Google, INEGI

```
TRACEROUTE RESULTS
================================

Nodes to get there
-----------------------------
1 TtlExpired Address: 192.168.1.1 Host Name: Wireless_Broadband_Router.home
2 TtlExpired Address: 98.118.10.1 Host Name: L100.BSTNMA-VFTTP-124.verizon-gni.ne
   City.Name: Hudson
   State: Massachusetts
   State Code: MA
   Country: United States
   Country Code: US
   Postal Code: 01749
   Latitude: 42.3897
   Longitude: -71.5401
3 TtlExpired Address: 130.81.137.4 Host Name: G0-7-2-7.BSTNMA-LCR-21.verizon-gni.
4 TtlExpired Address: 130.81.151.60 Host Name: ae1-0.BOS-BB-RTR1.verizon-gni.net
5 TtlExpired Address: 152.63.20.69 Host Name: 0.ae11.XL3.NYC1.ALTER.NET
6 TtlExpired Address: 152.63.19.201 Host Name: 0.xe-10-0-0.BR1.NYC1.ALTER.NET
7 *** Timed Out ***
8 TtlExpired Address: 4.69.138.222 Host Name: vlan51.ebr1.NewYork2.Level3.net
9 TtlExpired Address: 4.69.201.45 Host Name: ae-47-47.ebr1.NewYork1.Level3.net
10 TtlExpired Address: 4.69.134.78 Host Name: ae-91-91.csw4.NewYork1.Level3.net
11 TtlExpired Address: 4.69.155.210 Host Name: ae-4-90.edge4.NewYork1.Level3.net
12 TtlExpired Address: 4.28.130.94 Host Name: YAHOO-INC.edge4.NewYork1.Level3.ne
13 TtlExpired Address: 216.115.100.74 Host Name: ae-2.pat2.bfz.yahoo.com
   City.Name: Sunnyvale
   State: California
```

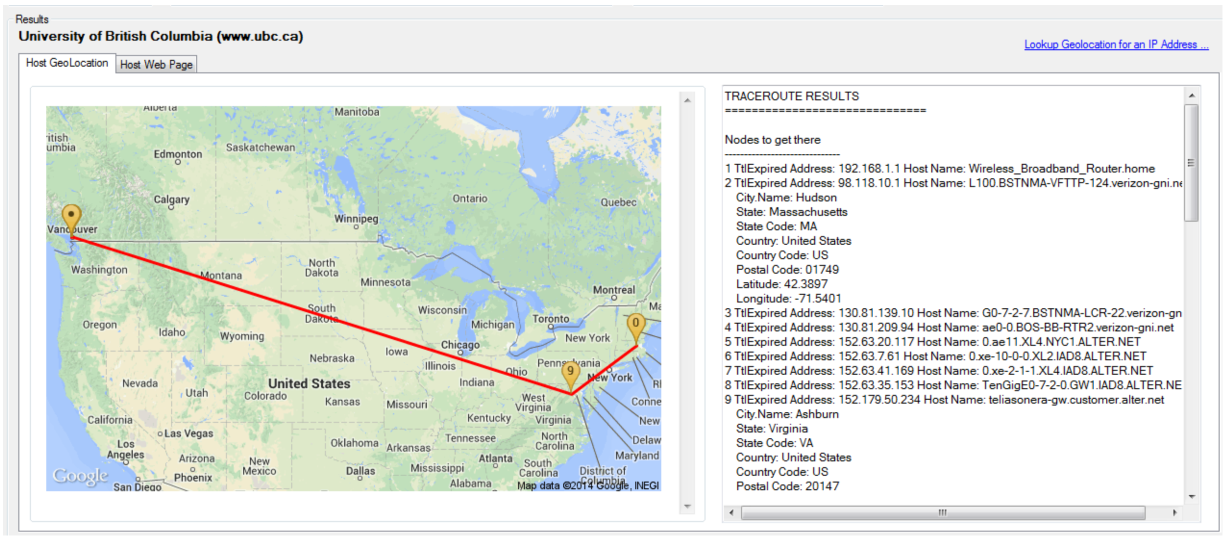
University of Manitoba, Canada:

The network path for the University of Manitoba, CA goes through Purchase, NY and Englewood, Colorado.


Results
University of Manitoba (umanitoba.ca)
Host GeoLocation
Host Web Page
Lookup Geolocation for an IP Address ...
Map data ©2014 Google, INEGI

```
TRACEROUTE RESULTS
================================

Nodes to get there
-----------------------------
1 TtlExpired Address: 192.168.1.1 Host Name: Wireless_Broadband_Router.home
2 TtlExpired Address: 98.118.10.1 Host Name: L100.BSTNMA-VFTTP-124.verizon-gni.ne
   City.Name: Hudson
   State: Massachusetts
   State Code: MA
   Country: United States
   Country Code: US
   Postal Code: 01749
   Latitude: 42.3897
   Longitude: -71.5401
3 TtlExpired Address: 130.81.137.4 Host Name: G0-7-2-7.BSTNMA-LCR-21.verizon-gni.
4 TtlExpired Address: 130.81.151.60 Host Name: ae1-0.BOS-BB-RTR1.verizon-gni.net
5 TtlExpired Address: 152.63.20.69 Host Name: 0.ae11.XL3.NYC1.ALTER.NET
6 *** Timed Out ***
7 TtlExpired Address: 140.222.229.97 Host Name: 0.ae1.BR3.NYC4.ALTER.NET
   City.Name: Purchase
   State: New York
   State Code: NY
   Country: United States
   Country Code: US
   Postal Code: 10577
   Latitude: 41.0387
   Longitude: -73.7173
```

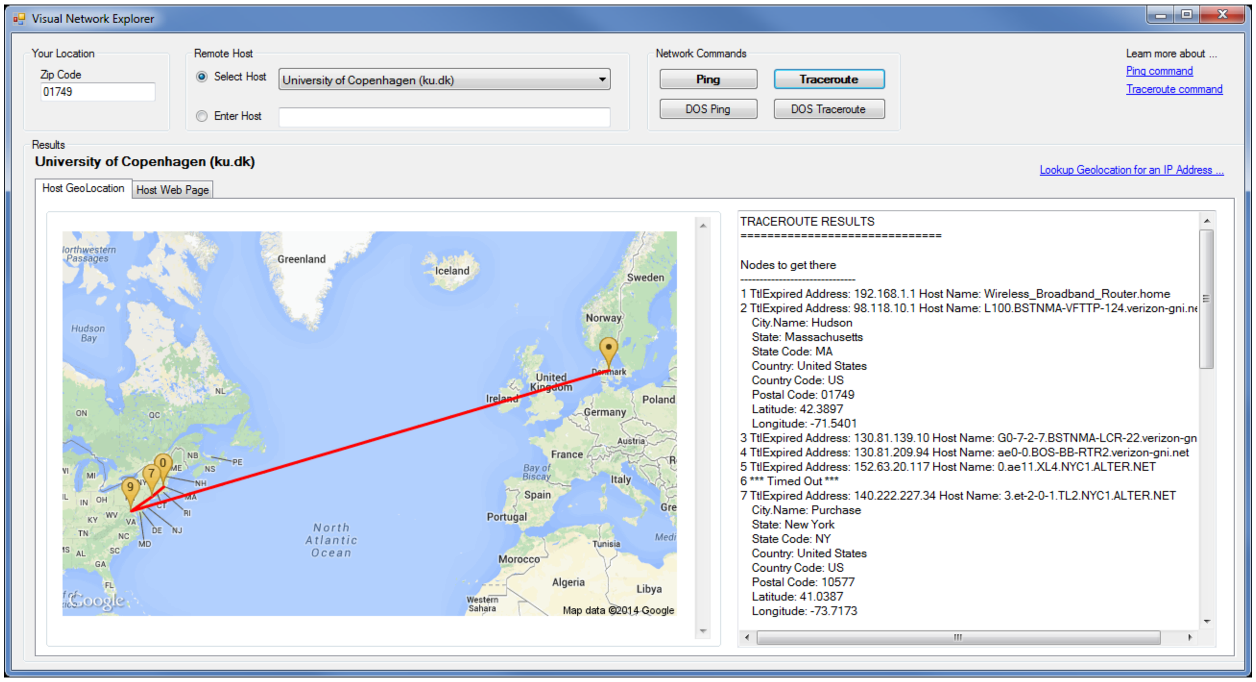
University of British Columbia, CA


Results
Lookup Geolocation for an IP Address ...
Host GeoLocation
Host Web Page
Map data ©2014 Google, INEGI

University of British Columbia (www.ubc.ca)

```
TRACEROUTE RESULTS
================================

Nodes to get there
------------------------------
1 TtlExpired Address: 192.168.1.1 Host Name: Wireless_Broadband_Router.home
2 TtlExpired Address: 98.118.10.1 Host Name: L100.BSTNMA-VFTTP-124.verizon-gni.ne
   City.Name: Hudson
   State: Massachusetts
   State Code: MA
   Country: United States
   Country Code: US
   Postal Code: 01749
   Latitude: 42.3897
   Longitude: -71.5401
3 TtlExpired Address: 130.81.139.10 Host Name: G0-7-2-7.BSTNMA-LCR-22.verizon-gn
4 TtlExpired Address: 130.81.209.94 Host Name: ae0-0.BOS-BB-RTR2.verizon-gni.net
5 TtlExpired Address: 152.63.20.117 Host Name: 0.ae11.XL4.NYC1.ALTER.NET
6 TtlExpired Address: 152.63.7.61 Host Name: 0.xe-10-0-0.XL2.IAD8.ALTER.NET
7 TtlExpired Address: 152.63.41.169 Host Name: 0.xe-2-1-1.XL4.IAD8.ALTER.NET
8 TtlExpired Address: 152.63.35.153 Host Name: TenGigE0-7-2-0.GW1.IAD8.ALTER.NE
9 TtlExpired Address: 152.179.50.234 Host Name: teliasonera-gw.customer.alter.net
   City.Name: Ashburn
   State: Virginia
   State Code: VA
   Country: United States
   Country Code: US
   Postal Code: 20147
```

University of Copenhagen (ku.dk)



Zip Code: 01749

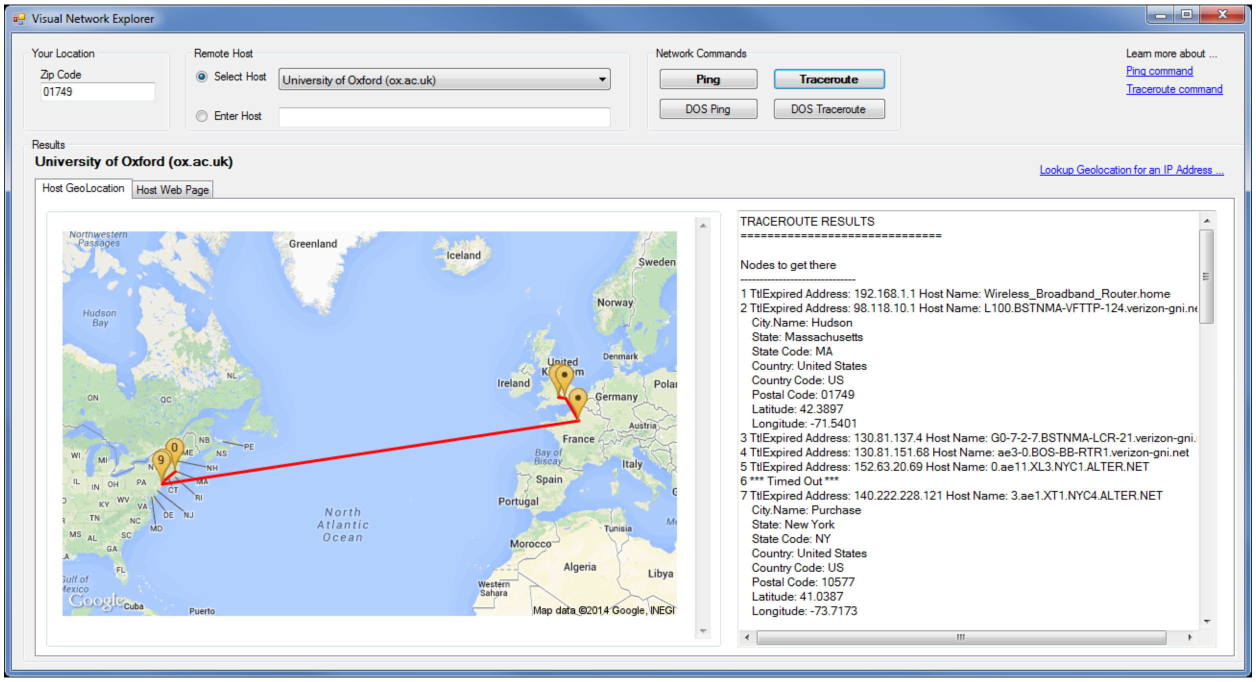
(●) Select Host: University of Copenhagen (ku.dk)

( ) Enter Host:

University of Copenhagen (ku.dk)

```
TRACEROUTE RESULTS
================================

Nodes to get there
------------------------------
1 TtlExpired Address: 192.168.1.1 Host Name: Wireless_Broadband_Router.home
2 TtlExpired Address: 98.118.10.1 Host Name: L100.BSTNMA-VFTTP-124.verizon-gni.ne
   City.Name: Hudson
   State: Massachusetts
   State Code: MA
   Country: United States
   Country Code: US
   Postal Code: 01749
   Latitude: 42.3897
   Longitude: -71.5401
3 TtlExpired Address: 130.81.139.10 Host Name: G0-7-2-7.BSTNMA-LCR-22.verizon-gn
4 TtlExpired Address: 130.81.209.94 Host Name: ae0-0.BOS-BB-RTR2.verizon-gni.net
5 TtlExpired Address: 152.63.20.117 Host Name: 0.ae11.XL4.NYC1.ALTER.NET
6 *** Timed Out ***
7 TtlExpired Address: 140.222.227.34 Host Name: 3.et-2-0-1.TL2.NYC1.ALTER.NET
   City.Name: Purchase
   State: New York
   State Code: NY
   Country: United States
   Country Code: US
   Postal Code: 10577
   Latitude: 41.0387
   Longitude: -73.7173
```

University of Oxford (ox.ac.uk)



```
TRACEROUTE RESULTS
==============================

Nodes to get there
------------------------------
1 TtlExpired Address: 192.168.1.1 Host Name: Wireless_Broadband_Router.home
2 TtlExpired Address: 98.118.10.1 Host Name: L100.BSTNMA-VFTTP-124.verizon-gni.ne
    City.Name: Hudson
    State: Massachusetts
    State Code: MA
    Country: United States
    Country Code: US
    Postal Code: 01749
    Latitude: 42.3897
    Longitude: -71.5401
3 TtlExpired Address: 130.81.137.4 Host Name: G0-7-2-7.BSTNMA-LCR-21.verizon-gni.
4 TtlExpired Address: 130.81.151.68 Host Name: ae3-0.BOS-BB-RTR1.verizon-gni.net
5 TtlExpired Address: 152.63.20.69 Host Name: 0.ae11.XL3.NYC1.ALTER.NET
6 *** Timed Out ***
7 TtlExpired Address: 140.222.228.121 Host Name: 3.ae1.XT1.NYC4.ALTER.NET
    City.Name: Purchase
    State: New York
    State Code: NY
    Country: United States
    Country Code: US
    Postal Code: 10577
    Latitude: 41.0387
    Longitude: -73.7173
```

Text of Traceroute for University of Oxford (ox.ac.uk)

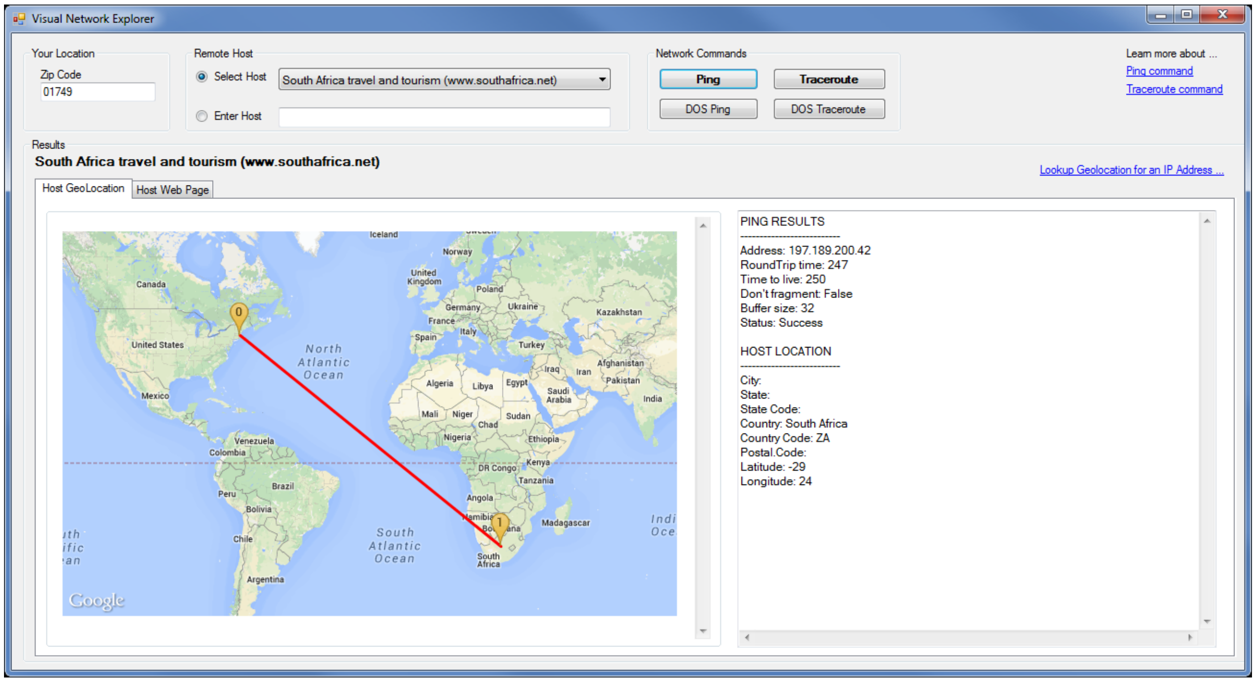
Path goes Hudson, MA > Purchase, NY > Brooklyn, NY > UK but shows a point in France even though not listed.

```
TRACEROUTE RESULTS
==============================

Nodes to get there
------------------------------
1 TtlExpired Address: 192.168.1.1 Host Name: Wireless_Broadband_Router.home
2 TtlExpired Address: 98.118.10.1 Host Name: L100.BSTNMA-VFTTP-124.verizon-gni.net
    City.Name: Hudson
    State: Massachusetts
    State Code: MA
    Country: United States
    Country Code: US
    Postal Code: 01749
    Latitude: 42.3897
    Longitude: -71.5401
3 TtlExpired Address: 130.81.137.4 Host Name: G0-7-2-7.BSTNMA-LCR-21.verizon-gni.net
4 TtlExpired Address: 130.81.151.68 Host Name: ae3-0.BOS-BB-RTR1.verizon-gni.net
5 TtlExpired Address: 152.63.20.69 Host Name: 0.ae11.XL3.NYC1.ALTER.NET
6 *** Timed Out ***
7 TtlExpired Address: 140.222.228.121 Host Name: 3.ae1.XT1.NYC4.ALTER.NET
    City.Name: Purchase
    State: New York
    State Code: NY
    Country: United States
    Country Code: US
    Postal Code: 10577
    Latitude: 41.0387
    Longitude: -73.7173
8 TtlExpired Address: 152.63.21.61 Host Name: TenGigE0-4-0-0.GW8.NYC4.ALTER.NET
9 TtlExpired Address: 152.179.72.122 Host Name: tinet-gw.customer.alter.net
    City.Name: Brooklyn
    State: New York
    State Code: NY
    Country: United States
    Country Code: US
    Postal Code:
    Latitude: 40.6501
    Longitude: -73.9496
10 TtlExpired Address: 89.149.185.233 Host Name:
11 TtlExpired Address: 141.136.103.210 Host Name:
12 TtlExpired Address: 146.97.33.2 Host Name:
    City.Name: London
    State: England
    State Code: ENG
    Country: United Kingdom
    Country Code: GB
    Postal Code:
    Latitude: 51.5142
    Longitude: -0.0931
13 TtlExpired Address: 146.97.37.206 Host Name:
    City.Name: London
    State: England
    State Code: ENG
    Country: United Kingdom
    Country Code: GB
    Postal Code:
    Latitude: 51.5142
    Longitude: -0.0931
14 TtlExpired Address: 193.63.108.129 Host Name:
15 *** Timed Out ***
16 TtlExpired Address: 193.63.109.110 Host Name:
    City.Name: Wantage
    State: Oxfordshire
    State Code: OXF
    Country: United Kingdom
    Country Code: GB
    Postal Code:
    Latitude: 51.5833
    Longitude: -1.4
17 TtlExpired Address: 192.76.21.11 Host Name:
    City.Name: Oxford
    State: Oxfordshire
    State Code: OXF
    Country: United Kingdom
    Country Code: GB
    Postal Code:
    Latitude: 51.75
    Longitude: -1.25
18 TtlExpired Address: 192.76.22.200 Host Name:
    City.Name: Oxford
    State: Oxfordshire
    State Code: OXF
    Country: United Kingdom
    Country Code: GB
    Postal Code:
    Latitude: 51.75
    Longitude: -1.25
19 TtlExpired Address: 192.76.32.62 Host Name: boucs-lompi1.sdc.ox.ac.uk
    City.Name: Oxford
    State: Oxfordshire
    State Code: OXF
    Country: United Kingdom
    Country Code: GB
    Postal Code:
    Latitude: 51.75
    Longitude: -1.25
```

```
Host Destination
------------------------------
Node #20
Status:  Success
Address:  129.67.242.154
Host Name:
RoundTrip time:  87
City:  Oxford
State:  Oxfordshire
State Code:  OXF
Country:  United Kingdom
Country Code:  GB
Postal Code:
Latitude:  51.75
Longitude:  -1.25
```

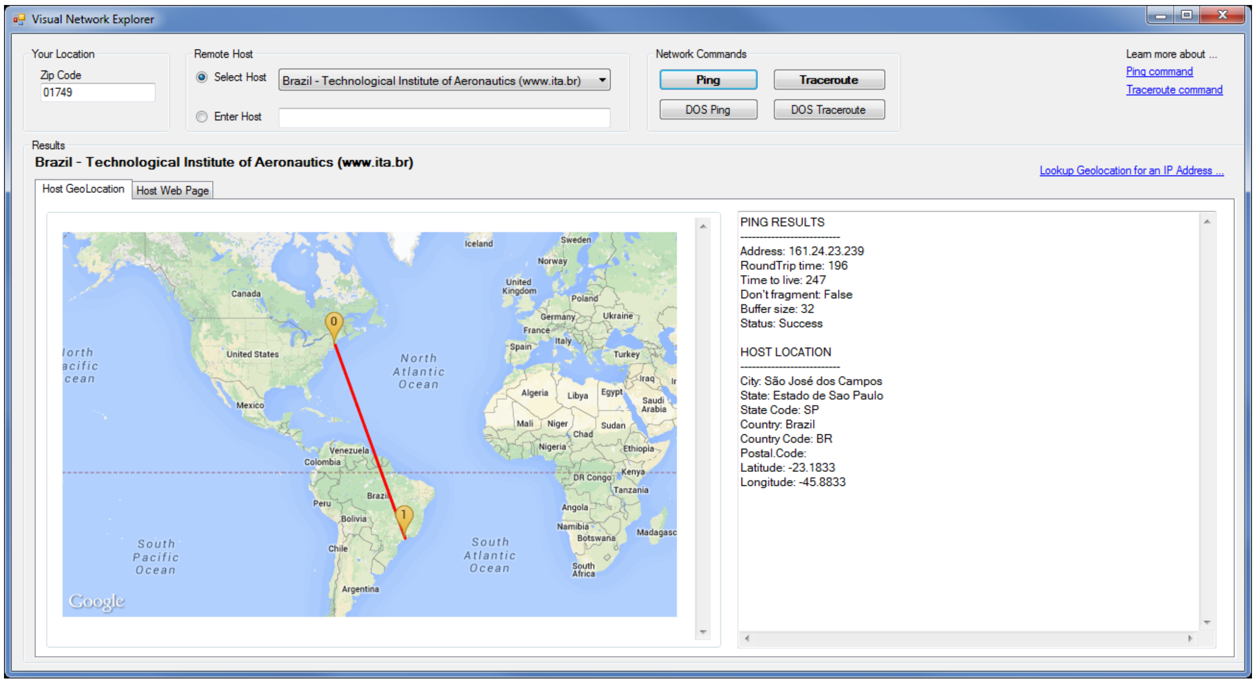
South Africa travel and tourism (www.southafrica.net)

Ping results:



```
PING RESULTS
-------------------------
Address: 197.189.200.42
RoundTrip time: 247
Time to live: 250
Don't fragment: False
Buffer size: 32
Status: Success

HOST LOCATION
-------------------------
City:
State:
State Code:
Country: South Africa
Country Code: ZA
Postal Code:
Latitude: -29
Longitude: 24
```

Brazil - Technological Institute of Aeronautics (www.ita.br)



Your Location

Zip Code: 01749

Remote Host

Select Host: Brazil - Technological Institute of Aeronautics (www.ita.br)

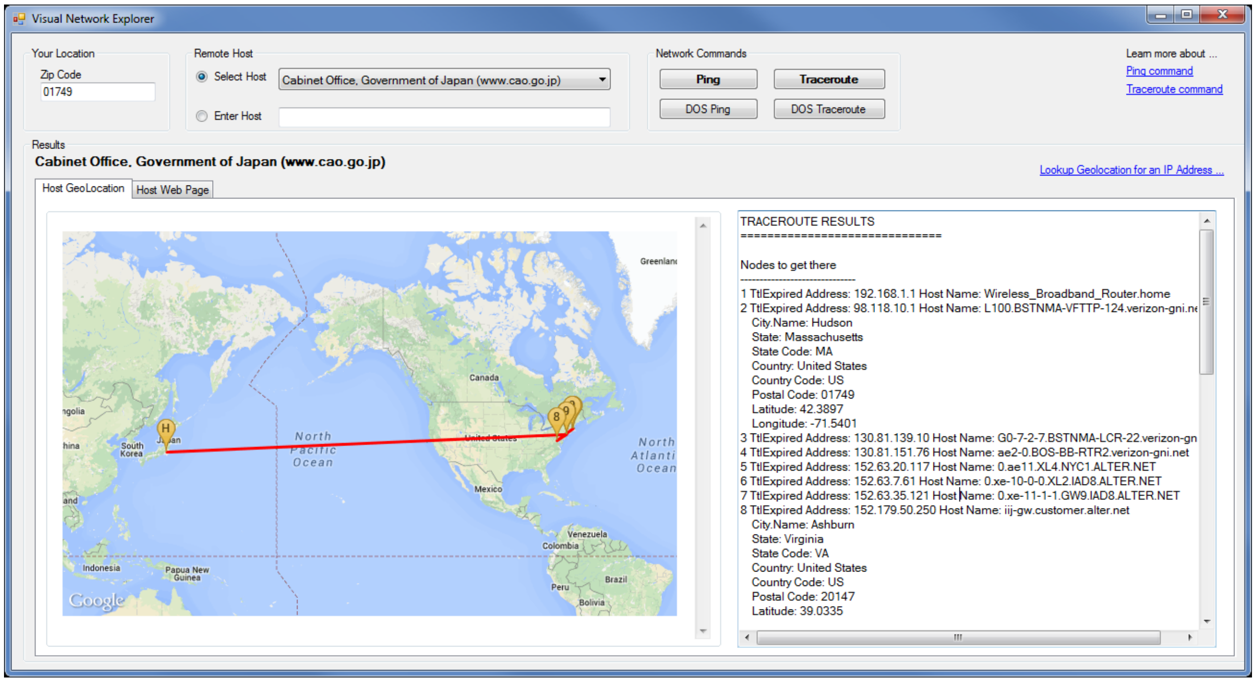
Enter Host:

Results

Brazil - Technological Institute of Aeronautics (www.ita.br)

```
PING RESULTS
------------------------
Address: 161.24.23.239
RoundTrip time: 196
Time to live: 247
Don't fragment: False
Buffer size: 32
Status: Success

HOST LOCATION
------------------------
City: São José dos Campos
State: Estado de Sao Paulo
State Code: SP
Country: Brazil
Country Code: BR
Postal Code:
Latitude: -23.1833
Longitude: -45.8833
```

Cabinet Office, Government of Japan (www.cao.go.jp)



Your Location

Zip Code: 01749

Remote Host

Select Host: Cabinet Office, Government of Japan (www.cao.go.jp)

Enter Host:

Results

Cabinet Office, Government of Japan (www.cao.go.jp)

```
TRACEROUTE RESULTS
==============================

Nodes to get there
------------------------------
1 TtlExpired Address: 192.168.1.1 Host Name: Wireless_Broadband_Router.home
2 TtlExpired Address: 98.118.10.1 Host Name: L100.BSTNMA-VFTTP-124.verizon-gni.ne
   City.Name: Hudson
   State: Massachusetts
   State Code: MA
   Country: United States
   Country Code: US
   Postal Code: 01749
   Latitude: 42.3897
   Longitude: -71.5401
3 TtlExpired Address: 130.81.139.10 Host Name: G0-7-2-7.BSTNMA-LCR-22.verizon-gn
4 TtlExpired Address: 130.81.151.76 Host Name: ae2-0.BOS-BB-RTR2.verizon-gni.net
5 TtlExpired Address: 152.63.20.117 Host Name: 0.ae11.XL4.NYC1.ALTER.NET
6 TtlExpired Address: 152.63.7.61 Host Name: 0.xe-10-0-0.XL2.IAD8.ALTER.NET
7 TtlExpired Address: 152.63.35.121 Host Name: 0.xe-11-1-1.GW9.IAD8.ALTER.NET
8 TtlExpired Address: 152.179.50.250 Host Name: iij-gw.customer.alter.net
   City.Name: Ashburn
   State: Virginia
   State Code: VA
   Country: United States
   Country Code: US
   Postal Code: 20147
   Latitude: 39.0335
```