COSC 417 – Topics in Networking
Lab 2 – Fall 2025

In this lab, we'll be using the same techniques of performing a traceroute, and then an ASN lookup, but we'll be automating it so that we can perform lookups quite a bit faster. We will use Python for this lab.

The iplist File

The iplist in our program is a simple text file, with each line containing one destination IP address. These will be the target IPs for our traceroute. You should come up with 20 IP addresses for your iplist file. It would be a good idea to try to use IP addresses from distant, fixed sources, so that we can see longer routes with more inter-AS routing.

Some ideas of good IP addresses to use:

- DNS servers, like Google's (8.8.8.8)
- University/College websites, since they are often hosted locally
- Time servers, such as the NIST time server in Boulder, Colorado (132.163.96.1)
- Web pages for scientific or government institutions (such as the Government of Canada website, 167.40.79.24)

Each IP address should be in standard IPv4 or IPv6 format and take up one line in our iplist.txt file. We can loop through this easily using code like that shown below:

```
ipList = open("iplist.txt", "r")
for ip in ipList:
     print(ip)
```

Performing the Traceroute

Now, we will need iterate over the iplist.txt file, and for each IP address we perform a traceroute using the Windows utility `tracert <ip>`. This can be done with the ***os*** library in Python as it has been done in the code. The example code below shows how to run a single IP, you will need to make it into a loop that checks each IP in the iplist.txt.

```
import os
ip = '8.8.8.8'
myData = os.popen('tracert -4 ' + ip).read()
print(myData)
```

Once we've performed the traceroute, we'll need to parse the data output. This can be done with a simple regular expression (RegEx), that looks for IPv4-formatted IP addresses. We remove the first IP in the list,

as it is the target IP (and appears twice - once at the beginning of the traceroute output, and once at the end). An example is given below:

```
import re
route = re.findall('(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})', myData)[1:]
print(route)
```

The output from the regular expression should be a Python list object of each IP address, beginning with your local address (i.e. 192.168.1.254) and ending with the destination address (i.e. 8.8.8.8). Once you have that output, you're ready to move on to the next step.

Using the ipwhois Module

Now that we have a list of IPs in our traceroute, we can begin performing lookups to determine what autonomous system is associated with each IP. We can do this using the ***ipwhois*** module. This can be installed easily using pip: ***pip install ipwhois***. This library has a function to perform a lookup on an IPv4 or IPv6 address and return the associated ASN and ASN description.

We will want to iterate over our route variable, which contains the list of IPs in the route:

```
for ip in route:
      print(ip)
```

We will need to declare each IP, currently a string, as a proper network resource to the ipwhois module:

```
import ipwhois
ip = '8.8.8.8'
myIp = ipwhois.net.Net(ip)
```

And then we'll need to pass this to the IPASN function to return an object, and then call the lookup function to get our results:

```
myObj = ipwhois.asn.IPASN(myIp)
results = myObj.lookup()
print(results)
```

The result set includes a large amount of data, including what ASN registry was used, the ASN number, CIDR block prefix, country, creation date, and description (often a company or institution name - 8.8.8.8 returns "GOOGLE, US")

This whole thing will need to be wrapped in a ***try/except group***, as certain IP addresses (such as local IPs, 192.168..) will cause ***ipwhois*** to return an error (can't look up the ASN of a private IP address):

```
try:
      #Your ipwhois code here
except:
      print("Non-lookupable IP address.")
```

Finally, you should write out each IP, ASN, and AS Description to a text file. The following example code shows opening a file, and then performing a write (don't forget the newline):

```
with open("asnoutput.txt", "a") as file:
      outputData = [ip, results['asn'], results['asn_description]
      file.write(str(outputData) + "\n")
```

You should print out a divider line every time you move onto the next IP in your iplist.txt file, so that you have a single output file of all the traceroute ASN information, split into blocks with newlines. If you forget to do this, it will be very hard to identify where one route begins and another route ends.

Once you've completed this step and you're able to produce reliable output, you're all done! We'll use this data file in the next lab to generate maps of the ASNs that we have looked up (with the help of a much larger iplist file).

Grading

When you're done, submit your ***Python script file***, as well as your ***iplist.txt*** file on Moodle.