**Lab 2: Enterprise Networking**

**PRELIMINARIES**

In this lab you will be a detective. There is a bad guy in your network. Something is going wrong with your routing. How do you figure out what is going on? In this lab you will use the real tools and mechanisms that operators use to troubleshoot traffic, analyze topologies, and trace bad guys in their networks. In addition to helping you understand these methodologies, this lab can also help you better understand routing and protocols that run in common network environments.

For this lab, it may help to familiarize yourself with basic linux text processing commands. Do some google searching to get a sense of how the following commands work. Make a small text file on your computer and try them out:

1. cat

2. grep

3. cut

4. sort (eg. sort -u)

5. awk (eg. to check if a column has some desired value)

It is possible to complete this entire lab on a Windows machine, but if you can do it on Linux or MacOS. Some of the tools we use may be easier to get working in those environments.

This lab will have a number of questions for you to investigate. Please include answers to all of these questions in your report.

**PART 1: Debugging exterior routing with BGP Updates**

So, suppose your network is working fine, but your customer are having trouble connecting to destinations outside your network. You've got three transit providers. What do you do? It would be nice if you could localize the problem to one of your providers, or to one of their providers, or some location in the Internet, so you can get in touch with them and work with them to troubleshoot the problem. But, how can you figure out what is going wrong outside your network?

To do this, operators analyze BGP advertisements. They can analyze updates they directly receive, as well as "looking glass" services (for example [RouteViews](http://www.routeviews.org/)). They can then analyze the contents of these updates to localize problems. For example, if many routes going through a particular AS are experiencing problems, perhaps that AS may be involved in the problem.

Let's discuss things in more detail. A network *topology* consists of the interconnection and arrangement of equipment (routers, links) that makes up the network. The complete topology of the Internet is unknown. Yet knowing the structure of the Internet is crucial in designing simulation/evaluation environments to test new protocols, in understanding the nature of how different networks interconnect, in searching for weak points and analyzing resilience of the Internet, and in provisioning for future growth patterns. To deal with this problem, researchers perform measurements to infer the structure of the Internet graph on various levels. In this problem we will infer the AS-level (ISP-level) topology of the Internet graph. In particular, for each AS in the Internet, we will determine which other ASes are connected to it. This is a simpler problem than determining how many times ASes peer with one another, or determining the router-level Internet topology, or inferring what routing policies are used at each router, etc. (we will discuss these harder problems in lecture).



**Figure 1: RouteViews Architecture.** *The RouteViews project runs a collector to collect BGP updates from multiple ASes. This could have been done by writing a script to ssh into routers at various ASes and doing show ip route, but for added security, another approach is followed. In particular, the RouteViews Collector peers and receives BGP feeds from each of a number of volunteer "vantage point" routers. This allows RouteViews to see the best route from multiple places in the Interdomain routing system.*

The University of Oregon Route Views project (www.routeviews.org) maintains eBGP peering sessions with routers in a number of ASes. Route Views maintains historical records of BGP routing updates logged on each of these sessions. Analyzing these routing updates can provide insights into stability of the global routing system, distribution of addresses, and the structure of the Internet topology. Here, we will study the structure of the AS-level topology.

Routing updates (UPDATES) and BGP table snapshots (RIBS) are located at <http://archive.routeviews.org/bgpdata/> . Documentation for file contents may be found at http://www.routeviews.org/data.html.

To convert the contents of those files to human-readable format, compile https://bitbucket.org/ripencc/bgpdump/downloads to get a bgpdump executable. (Instructions for how to compile provided at: <https://bitbucket.org/ripencc/bgpdump/wiki/Home>. If you get errors saying "... not found", try installing autoconf, zlib1g-dev, and libbz2-dev before you run the .sh file again). When you run this file consider using the "-m" flag, which puts all the output on one line, which makes it easier to play around with using sed/grep/cut and other shell commands.

Please investigate and answer the following questions in your report:

1. Pretend you are an AS directly peered with vantage point 12.0.1.63. Process all updates for this vantage point for the day of January 29, 2018 (all updates for that day, not just one snapshot file - consider using wget and check out the "-A" flag and how to make wget ignore robots.txt and recurse just one level). Which routes (to prefixes) are the most unstable? What is going on with those routes, why are they unstable, what do you observe?

2. Now pretend you are an AS directly peered with all vantage points. Plot the number of updates per minute you receive from each peer (have a line chart, with one line per vantage point, and one data point on each line per minute - x axis is minutes, y axis is # of updates in that minute -- consider using a map data structure). Include the plot in your report. What do you observe? If you were going to choose a peer to purchase service from, which would you choose?

3. Level 3 Communications (AS 3549) provides two feeds. Suppose you are peered with both of them. Are they doing consistent export (i.e. Either one prefix is advertised to an AS by all the BGP peers, or none of the BGP peers)? (Hint: "sort -u",”diff” may provide a simple way to solve this problem)

4. Let's find some ASes that are not doing a good job. Find the top ASes that are advertising the most prefixes. Could they aggregate? Give an example.

5. Do you see any ASes advertising bogons? How would you protect your network against those? (Hint: go to bgp.he.net and search for a few ASes)

6. Suppose you are an enterprise that has traditionally done default routing. But you are getting bigger and now want to participate in BGP. Do some performance analysis on these updates to get a sense of (a) your CPU needs (i.e. How fast the CPU is?) - hint, consider # updates per second you'd need to process (b) your TCAM needs (i.e. How large your TCAM will be? Consider the routing table size). Keep in mind overflows/overload are really bad (why?) - so would you add headroom? How much?

7. Now suppose you are an employee that has access to the BGP routers at your company. You don't like Facebook and want to take them offline. How would you do it? Note Facebook is a distributed service.

8. Now suppose you are the US Department of Defense. You send a lot of traffic to US Kadena Air Force Base, Japan. Does that traffic go through networks owned by any other foreign countries that might be considered adversaries of the United States, and that might want to snoop on that data?

9. Think of something else interesting to analyze in this data - analyze it, and report what you find.

10. Suppose you find out that AS 46479 is the source of a lot of problems. How would you get in touch with them?

11. In addition to inferring the Internet’s AS-level topology, it is also useful to infer the way that traffic flows over that topology. The manner in which an AS advertises a route in the Internet can very commonly be classified into one of several categories: (a) *provider-customer* relationships, where a customer pays a provider money (typically) for service. In this case, the provider advertises all routes it receives to the customer, and advertises all routes from the customer to all its neighbors. (b) *peer-peer* relationships, where two ASes agree to peer out of their own mutual benefit, and typically no money is exchanged. ASes almost always advertise all their routes from customers to peers. However, they almost always prevent provider/peer routes from being advertised to other provider/peers.

Instead of performing this inference yourself, we will provide a file containing these relationships for you: <http://www.cs.illinois.edu/~caesar/courses/cs436/19980501.as-rel.txt.bz2>.

This file was originally downloaded from <http://www.caida.org/data/active/as-relationships/>, and contains an AS-level graph annotated with inter-ISP relationships. The format of the file is explained in comments at the start of the file; 0 means the ASes are peers, -1 means the first AS is a provider of the second.

Suppose an IP packet is sent from a computer in UIUC (AS 38) to Instagram (hosted by AWS, AS 6250). Using this file, name one possible sequence of ASes it may traverse.

**PART 2: Characterizing traffic aggregates with Netflow**

Characterizing Internet traffic is an important challenge. Studying Internet traffic allows ISPs to determine how to provision capacity, to perform more effective traffic engineering, and to localize sources of attacks and unwanted traffic (e.g., spam) flowing through their networks. While maintaining logs of every packet traversing an ISPs network would provide the greatest degree of resolution into network activity, doing so is not tractable. The largest ISPs transit hundreds of terabytes of traffic every day, creating storage and analysis problems that seem intractable.

To deal with this, Internet routers run a traffic characterization system known as *Netflow*. Netflow collects information about IP *flows*, as opposed to complete packet-level traces. A flow consists of a sequence of packets between a source IP/port and destination IP/port, that are observed within a fixed window of time at the router collecting the Netflow trace. To further improve scalability, Netflow relies on *sampling*, i.e., where instead of looking at every packet, the router only looks at every *n*th packet. While Netflow traces do not provide complete information about data traffic traversing a router, these traces are sufficient to observe a wide range of Internet traffic patterns. Further information about Netflow is available at <http://www.ietf.org/rfc/rfc3954.txt> .

In this problem, you will analyze Netflow traces to compute statistics about traffic characteristics and utilization on the Abilene Internet2 ISP network.

Download the Netflow trace located at <http://www.cs.illinois.edu/~caesar/courses/cs436/ft-v05.2005-11-26.001500+0000.gz>. This trace was originally downloaded from Abilene’s Internet2 Observatory web site (<http://www.internet2.edu/observatory/>). The Abilene Internet2 network is an ISP run by a non-profit consortium, which provides high-bandwidth connectivity between universities, government agencies, and some corporations. Abilene only provides connectivity between these entities, and not to the global Internet – so these traces only contain flows between Abilene-connected institutions. The provided trace was acquired from the router where UIUC connects to Abilene. For future reference, you can download traces from a larger set of routers and dates from the Abilene web site. To do this, you will need to sign up for an account by filling out the proposal process form on that site.

To convert Netflow traces into human-readable format, you will need to install the *flow-tools* package. For example, on Ubuntu/Debian, run *sudo apt-get install flow-tools*, on other systems download and install <ftp://ftp.eng.oar.net/pub/flow-tools/flow-tools-0.66.tar.gz>. In rare cases (e.g., if you’re using your home machine and messed up your Ubuntu setup), when installing flow-tools, some dependencies/kernel problems may arise, and the solution is to reinstall Ubuntu completely (in VM, for example). After installing, update & upgrade the kernel and then repair broken packages (look up how to do so). Other possible errors (such as package reconfiguration) can be resolved using online help. Then, to convert the file to human-readable format, you can run the following command:

~: > zcat ./ft-v05.2005-11-26.001500+0000.gz | flow-cat | flow-export -f2 > netflowOutput.txt

Please investigate and answer the following questions in your report:

1. From a machine within UIUC, perform a traceroute to a machine at IBM. Give the IP address of the first hop router (Which is not in UIUC or UI AS) which connects to UIUC.

2. What fraction of traffic is BitTorrent traffic (ports 6881-6999), and what fraction is web request (port 80) traffic (consider both source and destination ports)?

3. Plot a CDF of flow lengths measured in number of bytes per connection, and again in terms of number of packets per connection.

4. What are the top five /16’s owned by Abilene-connected institutions to which AS 680 sends traffic? Use whois to give the ASCII names of the companies/entities which own these prefixes. e.g., whois -h whois.cymru.com “-v AS10000”

5. What are the top five /16s owned by Abilene-connected institutions which send traffic to AS 680?

**PART 3: Analyzing packet-level traffic with tcpdump**

While Netflow-level traces are useful for studying Internet traffic on large scales, sometimes it becomes necessary to study traffic on a finer grain. For example, debugging protocol operation, or analyzing network attacks may require studying packet-level traces. Here, we will use packet-level traces to analyze behavior of the CodeRed Internet worm. This worm infected over 359,000 computers in less than 14 hours. In order to find a patch to the vulnerability, to study the worm’s propagation behavior, or to trace back to the source of a network’s infection, packet traces may be collected and manually analyzed using the approach you will perform below.

Here, we will use the *tcpdump* tool to record and play back traces of packets. tcpdump allows a host to sniff network packets. It supports a flexible filtering interface to limit the number of displayed packets, as well as the ability to log packets to a trace file, and to replay contents of traces.

As an alternative, you can also install Wireshark, and use the tshark tool to view the trace (e.g., *tshark -r*).

Please investigate and answer the following questions in your report:

1. Next, we will use tcpdump to analyze the contents of an existing trace. Download <http://www.cs.illinois.edu/~caesar/courses/cs436/CodeRedTraces.tgz>. This file contains a trace of a small network infected by the Code Red worm. More details about the trace are given at <http://www.bofh.sh/CodeRed/index.html>. You can display the contents of a trace by executing a command similar to *tcpdump -n -r CRed.07-19-01.dump*. The Code Red worm’s behavior is divided into stages: after initially infecting a machine, it first attempts to infect other machines for a period of 20 days. How many remote hosts did the infected machine (192.168.1.105) attempt to infect on July 19 2001? Also, how many hosts are infected per second (average), at which the infected machine attempted to infect remote hosts?

2. Find the first TCP syn sent to a host (any host) in the trace CRed.07-19-01.dump. How long did it take to receive an ack?

3. 20 days after infection, the Code Red worm begins to DDoS the Whitehouse’s web server (198.137.240.91). On July 21 2001, How many hosts are infected per second (average),at which the infected machine sent packets to the Whitehouse’s web server?

4. On July 30, 2001, what do you observe about the worm’s behavior? Is it performing DDoS or infecting hosts at the same rate it did previously?

5. If you wanted to protect your machine from being infected by Code Red, what sort of filters might you install in your firewall?

**PART 4: Analyzing reachability with packet probing and querying**

The previous sections described how you passively analyze your network; collecting and analyzing logs, receiving BGP updates, etc. In this section, we will conduct active analysis through sending packet probes, querying DNS, etc. Include the output from the commands you run in your report.

**4a. Testing connectivity with *ping***

Ping is a networking utility used to test reachability and latency. It works by sending [ICMP](https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol) messages to a particular IP address. The receiver, if running a proper implementation of ICMP, will reply back to the source.

Please perform the following steps, and answer the corresponding questions in your report:

1. Using ping, study the latency between where you are currently, and www.cs.illinois.edu. Run ping for a while and study how latency changes over time. What do you observe?

2. Next, run ping to test latency to (a) an interface on the same physical LAN as yourself (b) an interface on [www.tsinghua.edu.cn](http://www.tsinghua.edu.cn). Compare the average latency you find to what you discovered in the question above. What do you observe?

3. Disconnect yourself from the network while you are running ping. What happens? Connect to the network. What happens?

**4b. Studying IP-layer reachability using *traceroute***

Traceroute is a networking utility used to discover the IP-layer path between a pair of interfaces. It is also based on ICMP, but uses a clever trick - it repeatedly sends packets with increasing TTLs -- intermediate routers, upon TTL expiry, send an ICMP message back to the source. In so doing, traceroute can discover the set of IP-layer hops along the path to the provided destination.

Please perform the following steps, and answer the corresponding questions in your report:

1. Run traceroute to [www.cs.illinois.edu](http://www.cs.illinois.edu) from your current location. Then run it to government.ru. Which ISPs does your packet traverse? Can you figure out some cities that your path traverses? For the latter, which link is probably the trans-atlantic (or trans-pacific) link?

2. Perform a traceroute to www.youtube.com. Use Cogent's Looking glass (<http://www.cogentco.com/en/network/looking-glass>) from Paris, France to the same two sites ([www.youtube.com](http://www.youtube.com) and www.cs.illinois.edu) above. What do you observe?

**4c. Querying DNS with *dig***

dig is a tool that can query the Domain Name System (DNS) to obtain the domain name associated with an IP address or vice versa.

Please perform the following steps, and answer the corresponding questions in your report:

1. Use dig on your computer to find out the IP address of facebook.com.

2. Use <https://toolbox.googleapps.com/apps/dig/#A/> to find out the IP address of facebook.com. Why might this return a different answer?

3. Use dig to lookup [www.facebook.com](http://www.facebook.com) (instead of facebook.com -- add a "www" in front). Dig returns different information this time. What is the explanation of what is being returned? What is a CNAME record?

**4d. Checking interface properties with *ifconfig***

ifconfig is a command-line utility which prints and configures information about interfaces configured on your computer. Please use ifconfig to find out (and answer the corresponding questions in your report):

1. What are the layer 3 and layer 2 addresses of the interface your computer uses to send traffic to the public Internet?

2. How much traffic has been sent and received on that interface?

**PART 5: Putting it all together: Dealing with real problems**

In this section, we are going to get real. You are going to leverage the knowledge you acquired to deal with some real problems. Here are some links from the NANOG mailing list. Read the problem described in the post, provide a brief description of the problem that is going on, and answer how you would deal with the problem. Make sure you click "Next in Thread" to read the entire thread, not just the original post I link to below. Investigate and answer the following questions in your report:

1. [Attacks from poneytelecom.eu](https://mailman.nanog.org/pipermail/nanog/2018-January/093811.html). What exactly is the problem? If you were Dovid, how would you protect your network from these attacks?

2. [Spectrum prefix hijacks](https://mailman.nanog.org/pipermail/nanog/2018-January/093807.html). What exactly is the problem? What is Christopher Morrow's hypothesis about what is going on? What does "pure transit" mean?

3. [Packet loss through Level 3 in Southern California?](https://mailman.nanog.org/pipermail/nanog/2017-November/093094.html) Also see the followup [here](https://mailman.nanog.org/pipermail/nanog/2017-December/093196.html). What exactly is Greg's hypothesis? How do his results back that up? What do you think is going on? And how would you work around the problem to get better service to your customers if you were Greg?

4. [Contact info, AS4766 Korea Telecom](https://mailman.nanog.org/pipermail/nanog/2017-December/093256.html). What does Igor think happened to his network? How specifically could Korea Telecom have done that (what specifically would they configure)? Describe how could be due to a misconfiguration (what sort of mistake could Korea Telecom have made?) And also describe why this might be intentional (what sort of reasons might Korea Telecom have to block Igor's network?)

5. [Companies using public IP space owned by others for internal routing](https://mailman.nanog.org/pipermail/nanog/2017-December/093459.html). What is Robert talking about? Why might this be bad practice? Why might it be good practice? Later down that thread, what is Harald Koch talking about?

6. [Hurricane Maria: Summary of communication status - and lack of](https://mailman.nanog.org/pipermail/nanog/2017-October/092560.html) . What problems did Hurricane Maria cause to networks? How did operators deal with these problems? How would you make your network resilient to these problems?

7. Pick another email thread from the NANOG archives: <https://mailman.nanog.org/pipermail/nanog/> that describes a problem you find interesting. Then do the same thing, describe the problem and how you would solve it.

# **Post-Mortem/Wishlist**

1. In the code-red problem, the problem asks for the first outside host but no outside hosts in the trace. it then says the first host period, but the first host that replies replies after zero time. Would be nice to make problem more interesting. Better yet can we get a more recent trace?
2. Change Kadena to some other foreign site. Maybe identify some ISP in kadena and let them know that's the isp.