

Problem Set 1

*Instructor: Prof. Bruce Maggs**Computer Science Department, Duke University*

This problem set has four questions, each with several parts. Answer them as clearly and concisely as possible. You may discuss ideas with others in the class, but your solutions and presentation must be your own. Do not look at anyone else's solutions or copy them from anywhere. (Please refer to the Duke University honor code).

Turn in your solutions in on **February 6, 2007** in class.

1 DNS

In the first part of this question, you will perform some hands-on DNS queries using `dig` and play with DNS lookups from various applications to understand more about the DNS. In the second part of this question, you will implement a variation on a stub DNS resolver.

RFC 1035 may be helpful for answering some of these questions.

1. In this question, we'll warm up by learning a few things about Duke University DNS setup.
 - (a) What are the authoritative nameservers for `duke.edu`? How long will your resolver cache the records pointing to these nameservers? What are the Computer Science Department authoritative nameservers (*i.e.*, , for the domain `cs.duke.edu`)? Give two benefits of topologically diverse authoritative nameservers. Why do NS records return names, rather than IP addresses?
 - (b) What is another "canonical name" for the Duke University Web server?
 - (c) What is the primary mail exchanger for `cs.duke.edu`?
2. Now that you've had some experience playing with `dig`, in this part of the problem, we'll implement a stub resolver that performs iterative DNS queries. Most of the time, stub resolvers send queries with the "RD" (Recursion Desired) bit turned on. In this problem, you are *not* allowed to use the recursion bit. Of course, you are welcome to solve this problem any way you like (C,C++,JAVA,Phyton,Perl,shell script, etc). If you prefer, you may use the Ruby skeleton code provided at <http://www.cs.duke.edu/courses/spring08/cps214/hw/ps1/dns-resolv-rb.tgz>. This may save you the trouble of figuring out which modules to use, instrumenting your own performance measurements, etc.
 - (a) Why do stub resolvers typically set the RD bit?
 - (b) Implement a stub resolver that performs only iterative queries to resolve A records. To answer the next question, you'll want to make it possible to provide an option to your program to control the root nameserver.

Your resolver need not do anything special as far as caching, etc., but you should handle timeouts (*e.g.*, , querying the next preferred authoritative nameserver if the first does not respond).

Just make sure you can (1) point it at different root nameservers and (2) measure the time taken to resolve a query (the skeleton code is instrumented for this).

- (c) Use your query to resolve (1) `www.cs.duke.edu` and (2) `www.nytimes.com` at the following nameservers.
- `a.name-servers.net` (198.41.0.4)
 - `f.name-servers.net` (192.5.5.241)
 - `m.name-servers.net` (202.12.27.33)
 - `a.gtld-servers.net` (192.5.6.30)
- (i) Through what sequence of nameservers was each query referred? How long did each referral step take? Based on this, what fraction of DNS query time is saved by caching at local resolvers?
- (ii) What is the first referral when you send a query `www.cs.duke.edu` to `a.gtld-servers.net`? Is the answer the same everytime? Why or why not?
- (iii) How do stub resolvers typically choose root nameservers?

Please hand in your code to this problem as well. The code should be commented. Also, include a README file with instructions for compiling/executing your code.

2 Network Operator for a Day (with *rcc*)

To work on this problem, you will need the following three resources:

- The *first* routing table dump (or dumps, if you need them) on January 23, 2008 from the Internet2 backbone network. BGP table dumps are available at: <http://ndb2-blmt.abilene.ucaid.edu/bgp/STTL/2008.01/RIBS/>. You should choose the first dump made on January 23, 2008 (rib.20080123.0153.gz).
1. To parse rib* files you may use Marco d'Itri zebra parser at <http://www.linux.it/~md/software/zebra-dump-parser.tgz>
 - (a) Other than the sessions to private AS numbers, what are the ASes with the most number of eBGP sessions?
 - (b) At what routers does Microsoft have eBGP sessions to Internet2? (*Hint*: You will first have to figure out Microsoft's AS number!)
 - (c) Note that Microsoft is corporate, but Internet2 is supposedly a research and education network; why might Microsoft have eBGP sessions to Abilene?
 - (d) What prefixes that are advertised by Microsoft are reachable from Internet2? Which routing table did you look at to answer this question (and does it matter)?
 2. Observe an output of running rcc verifier at: <http://www.cs.duke.edu/courses/spring08/cps214/hw/ps1/rcc-html/>
 - (a) Click on "IS-IS Errors" and then on "MTU Mismatch Checks". What is an MTU mismatch, and why could it cause a problem? The pair of interfaces in question start with `ge-*`, which typically stands for "gigabit ethernet". Which value is likely the correct value for the MTU?

- (b) Under “BGP Errors”, click on “Information Flow”. These warnings indicate places where an import or export policy was configured in different ways on different routers for the same neighboring AS. What is a reasonable explanation for why “anomalous import” (*i.e.*, , different import policies on different neighboring routers) might be a reasonable thing for an operator to do?
- (c) Under “BGP Errors”, click on “iBGP Signaling”. What is meant by an “iBGP Signaling Partition”, and why is it bad?

3 Understanding BGP using table dumps

For this question, you will need to download the Routeviews routing table from <http://www.cs.duke.edu/courses/spring08/cps214/hw/ps1/oix-full-snapshot-2008-01-20-1800.dat.bz2> This file contains a Cisco BGP4 routing table snapshot, taken at Oregon Route Views (<http://www.routeviews.org/>) on January 20, 2008. (*Beware:* This is a text file that is 13MB, compressed. You should be able to analyze it without uncompressing it using, for example `bzcat`, `grep`, `less`, `searching into the file` - be patient when searching this is a really huge file !!!)

If you are curious about what other snapshots look like, you can find daily snapshots at <http://archive.routeviews.org/>

1. Find the routing table entry for Duke University network.
 - (a) What is the IP address of the best next hop from this router to Duke? How does this router know how to reach that next hop IP address?
 - (b) From the routing table file, what is the AS number for Duke?
 - (c) How many routes are there to get from this router to Duke?
 - (d) What is the best route to Duke? Why was this route selected as the best route?
 - (e) How many ASes must a packet traverse between the time it leaves the router and the time that it arrives at Duke?
 - (f) What are the AS numbers of all Duke’s upstream providers? What ISP does the above AS correspond to? (*Hint:* You can discover this information using a whois query.)
 - (g) In paths where Duke University uses Time Warner Telecom (AS4323) as an upstream, the AS path ends with two instances of the same AS number. Why? What is the likely relationship between this AS number and Time Warner Telecom?
 - (h) Use `traceroute` to measure route from some machine at Duke to the router that took the snapshot. Please include the output of your traceroute with your problem set. Is the sequence of ASes from Duke to the router the same as the reverse route in the trace data? Why might the reverse path differ? (Please list reasons other than the fact that your traceroute was performed at a different time as the table snapshot!)
2. Look at the routing table entry for 12.108.254.0/24. This entry has several routes marked with a “d”, for “damped”. Give a short, one-to-two sentence explanation for (1) why routers damp routes and (2) why routers keep damped routes. To answer this question, you may want to look at RFC 2439.

3. Several of the IP prefixes in the table are formatted as $w.x.y.z/m$. The mask field, m , specifies the length of the network mask to use when matching input destination addresses to entries in the table.
 - (a) Write down the bit-wise operation to determine whether a destination address, A_i , matches a prefix A/m in the routing table. A_i and A are 32 bits each.
 - (b) Find the first “Class C” CIDR address in the table (address prefix $\geq 192.0.0.0$). How many class C networks does this address correspond to? What is the maximum number of routing table entries that this single CIDR address saves? Why is it that we can only infer the maximum, and not the actual, number of addresses that this CIDR address saves?
 - (c) In the table, there are examples of groups of prefixes that have the same advertised AS path, but show up as separate entries in the routing table.¹
 - (i) Provide an example of non-contiguous prefixes (and the corresponding AS path) for which this is true. Why might non-contiguous prefixes have the same AS path?
 - (ii) Provide an example of contiguous prefixes (and the corresponding AS path) for which this is true. This practice is often called *deaggregation*. Why might this be done?
4. RouteViews makes available table snapshots from 1997 to present. Suppose you had access to all of these snapshots, as well as some routing table snapshots from pre-CIDR. For each of the following pieces of information available in the table snapshot, what information might you be able to infer about the evolution of the Internet?
 - (a) Only the destination addresses.
 - (b) Only the lines marked $*>$.
 - (c) Only the paths, with best next-hops marked.

4 Understanding IS-IS Using Packet Traces

Obtain the IS-IS packet traces from the Abilene network for January 2, 2007. For example, the trace from the Atlanta router is located at <http://ndb2-blmt.abilene.ucaid.edu/isis/2007.01/ATLA/isisd.20070102.gz>. Seven of the 11 Abilene backbone routers capture such traces. You will need all seven IS-IS traces for this day to answer this question.

1. In the trace, list the different types of IS-IS messages that you see and the purpose of each message.
2. From the traces, what is the LSA refresh interval? What are the advantages of setting this value to a small value? What are the disadvantages?
3. Looking and accounting for failures: How many occurred on this day? What type?
4. Compute the average propagation time between each pair of routers (there are 7 routers). To compute the average propagation time use at least 10 values.

¹For both parts of this problem, it’s sufficient to find the existence of one AS path that is advertised more than once. It is *not* necessary to find two prefixes for which *all* advertised paths are the same.