CS 4251: Computer Networking II March 8, 2008

Problem Set 2

Instructor: Prof. Nick Feamster *College of Computing, Georgia Tech*

This problem set has two 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 Georgia Tech honor code, posted on the course Web site).

Turn in your solutions in on **March 28, 2008** in class.

1 Physical Layer

1. Consider the 3-byte sequence `011011010100`.
   (a) Draw waveforms for this sequence using the following encodings: Non-return to zero, Non-return to zero invert on ones (NRZI), and Manchester encoding
   (b) What are the mimumum and maximum signal transition rates for each of these encodings (e.g., once per clock period, twice per clock period, etc.)? Give an example bitstream that would generate such a minimum or maximum transition rate.
2. Suppose that the spectrum of a channel is between 4 and 5 MHz and SNR is 24 dB. According to Shannon's formula, what is the capacity of this channel? Give two reasons why the channel capacity might not be achieved in practice.

2 Lookup Tables

Consider the routing table that you used for BGP analysis in the last assignment.

1. Suppose that this routing table were represented as a two-tier lookup trie, where the first level of the trie has a branching factor of 2^{24} and the second level has a branching factor of 2^8.
   (a) How many nodes will this trie require? Do not forget to count duplicate entries that will result when a prefix "shorter" than a /24 must be inserted into the table.
   (b) Suppose that each routing table entry (*i.e.,* , each prefix) is looked up once. What is the average number of memory accesses that will be required to look up a table entry? (Hint: Each entry will require either one or two lookups.)
2. Suppose that, instead of having to represent routing table entries in a binary trie, you could represent routing table entries by ranges of numbers.
   (a) In what cases might this function reduce the overall number of entries in the table? Given one example (include the prefixes and the next-hop IP address(es) for the routing table entries for your example).

(b) Give the number of routing table entries in the routing table (1) in the current table; and (2) the number that would result if routes could be represented as ranges.

3 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.

Update: The first part of this problem only appears to work on Georgia Tech's Sun cluster machines.

1. In this question, we'll warm up by learning a few things about Georgia Tech's DNS setup. You may find `dig` helpful for completing this problem.

   (a) Run "`traceroute ai`" from some machine on the Georgia Tech campus network.[1] Now run "`traceroute ai.`" from the same machine. Include the output from each run in your problem set writeup. Are the two traceroutes running traces to different machines? Why or why not?

   (b) What are the authoritative nameservers for `gatech.edu`? How long will your resolver cache the records pointing to these nameservers?

   What are the College of Computing's authoritative nameservers (*i.e.,* , for the domain `cc.gatech.edu`)? Give two benefits of topologically diverse authoritative nameservers.

   Why do NS records return names, rather than IP addresses?

   (c) What is another "canonical name" for the College of Computing's Web server?

   (d) What is are the mail exchanges for `cc.gatech.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. If you prefer, you may use the Ruby skeleton code that I have provided at `http://www.cc.gatech.edu/classes/AY2006/cs7260_spring/psets/ps1/aux/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) Use `dig` to resolve (1) `www.cc.gatech.edu` and (2) `www.nytimes.com` at the following nameservers. *Trace the sequence of queries through the nameserver hierarchy.* That is, do not just let `dig` return an answer; mimick the behavior of an iterative resolver.

   - a.name-servers.net (`198.41.0.4`)
   - f.name-servers.net (`192.5.5.241`)

[1] Depending on where on the campus you run this experiment, you may see different results.

- 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.cc.gatech.edu` to `a.gtld-servers.net`? Is the answer the same everytime? Why or why not?

(iii) Several of the root DNS servers use a technology called *anycast*. (1) What is it? (2) How is it implemented? (3) Use the traceroute "looking glass" servers (www.traceroute.org), or some set of distributed nodes such as PlanetLab[2] to find a DNS root anycast server that is hosted in two different places. Give the traceroute logs that correspond to the two different locations in your traceroute.

[2]Contact Prof. Feamster if you need access to a set of distributed machines.