Carnegie Mellon
Computer Science Department.
15-744 Spring 2008 Theory-ish Problem Set 4

This problem set has a few short questions. Answer them as clearly and concisely as possible. You may discuss ideas with others in the class, but your solutions and writeup must be your own. If you do discuss at length with others, please mention in your solution for the problem who you collaborated with, and optionally, the amount of time you discussed at length. I bet most of you aren't even reading this. Do not look at anyone else's solutions or copy them from anywhere.

This assignment is due by **3:00pm, Monday, April 21st** in class or to the course secretary in Wean Hall 8018.

Glossary

BGP: The Border Gateway Protocol
DHT: Distributed Hash Table (like Chord)
DNS: The Domain Name System
TTL: Time To Live values (how long a DNS record may be cached)
NS Records: Name Server records; those records that point to more specific authoritative servers for a DNS name.
RIAA: The Recording Industry Association of America. Defn (1) An industry group that attempts to ensure that artists are fairly compensated for their work. Alternate meaning: (2) An industry group that believes that launching a campaign of fear and intimidation by indiscriminately suing its customers and seven year old children is the most effective way to boost sales and ensure goodwill.

If you are in doubt of any terms used here, Google is your friend.

Solution: Total: 100 points.

A DNS Redirection

Harry Bovik is working on a web site that has multiple replicated servers located throughout the Internet. He plans on using DNS to help direct clients to their nearest server replica. He comes up with a hierarchical scheme. Harry has divided his server replicas into three groups (east, west and central) based on their physical location. A typical query occurs as follows:

A. When a client makes a query for `www.distributed.hb.com`, the root and `.com` name servers are contacted first. It returns the name server (NS) record for `ns1.hb.com`. The TTL of this record is set to 1 day.

B. The `ns1.hb.com` name server is then queried for the address. It examines the source of the name query and returns a NS record for one of {`east-ns, central-ns, west-ns`}`.distributed.com`. The choice of which name server is based on where `ns1` thinks the query came from.

C. Finally, one of {`east-ns, central-ns, west-ns`}`.distributed.com`. is contacted and it returns an address (A) record for the most lightly loaded server in its region. Answer the following 3 questions based on this design.

1. Harry's name server software has only two choices for TTL settings for A and NS records - 1 day and 1 minute. Harry chooses the following TTLs for each record below:

   1. NS record for {`east-ns, central-ns, west-ns`}`.distributed.com` - 1 day TTL.
   2. A record for {`east-ns, central-ns, west-ns`}`.distributed.com` - 1 day TTL.
   3. A record returned for the actual Web server - 1 minute TTL.

   Briefly explain why Harry's choices are reasonable, or why you would have made different choices.

   Solution: 25 points. The name server for a client is based on the region and hence probably does not change very often. Therefore, Harry sets the NS and A records for the name server address to 1 day.

   Harry wants the name server to direct clients to lightly loaded web servers. To do this, Harry must be able to control which web servers each client goes to. Therefore, Harry sets the TTL of the web server A record to 1 minute, so that clients won't cache the record for very long and will ask the name server which web server to use for subsequent requests. If the TTL was 1 day, each client would cache the first A record it got and then continue using the same web server for the entire day even if it becomes overloaded.

2. Harry notices that many clients are directed to servers in the wrong region (i.e. the client is not in the same region as the Web server chosen). He tracks down the problem and sees that the clients appear to be in some other region during name resolution. Why is this happening? Hint: a client may choose a local name server that is not geographically local!

   Solution: 25 points. One common situation where this occurs is when the client's local caching name server is located outside his or her region (e.g., if you're in Pittsburgh but your ISP's name server is in San Francisco). Remember that Harry's name servers don't actually see the DNS request from you; you send it to your local name server and your local name server performs a request to Harry's name servers on your behalf. Thus, Harry's name servers must assume that you are located where your local name server is.

3. Harry's Web site is especially popular among CMU students. The CMU network administrator estimates that there is one access from CMU every 3 minutes. Each access results in the application resolving the name `www.distributed.hb.com`. Assume the following:

   - No other DNS queries are made in CMU
   - All CMU clients use the same local name server.
   - This local name server is mapped to the `east-ns` region.
   - Web browsers do not do any caching on their own.

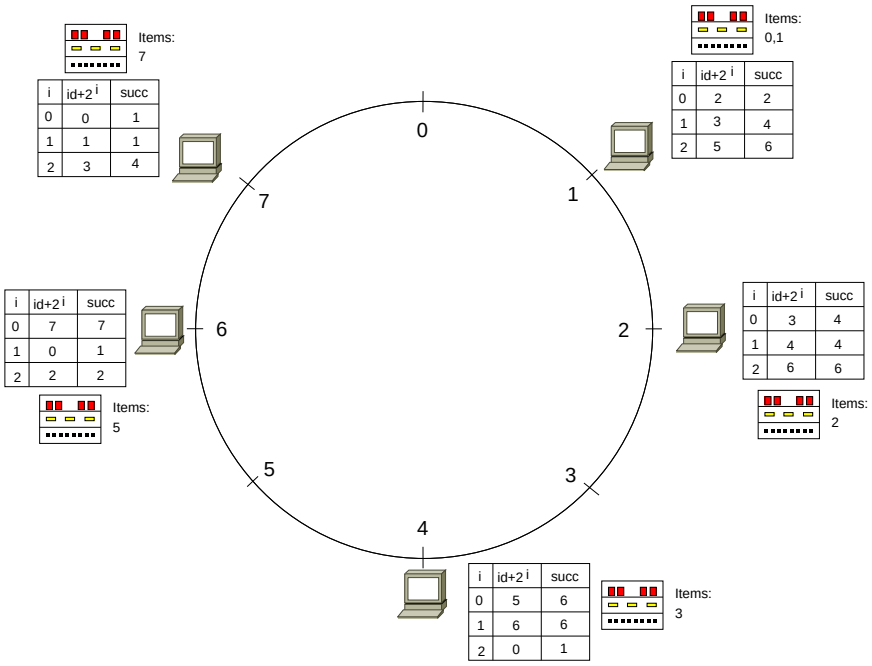
   How many accesses per hour will be made to the following name servers to resolve these CMU queries? Explain your calculation.

   1. The Root Servers
   2. `ns1.hb.com`
   3. `east-ns.distributed.com`

Solution: 25 points.

1. The Root Servers - 1/24 requests/hour
2. `ns1.hb.com` - 1/24 requests/hour
3. `east-ns.distributed.com` - 20 requests/hour

4. Vijay, in fear that the RIAA/MPAA/ASPCA will shut down his centralized P2P server like Napster, sets up a Chord for lookups and routing in his peer to peer network. Unfortunately (or fortunately, for you), Vijay's P2P network is not very popular (apparently nobody really wants to have 3TB of lolcats images[1]) and only consists of four (!) peers at the current moment with successor tables and items illustrated below. For example, *node 4* has *item 3*.



(a) List the nodes that will receive a query from *node 1* for *item 7*.

Solution: 10 points The query will begin at node 1, go to the highest node in the successor table which does not exceed the the number of the item which is node 6. Node 6 will then see that it has an entry in its table for 7 and forward it on to 7.

(b) List the nodes that will receive a query from *node 2* for *item 5*.

Solution: 5 points Node 2 does not have an entry for item 5, so the query is forwarded to the highest success in the table that does not exceed 5 which is node 4. Node 4 has an entry for 5 in its table which is at node 6, so it will now be forwarded to 6.

[1] http://icanhascheezburger.files.wordpress.com/2007/05/jesus_christ_its_a_lion.jpg

(c) In his uncontrollable Hulk-like rage over the unpopularity of large lolcats datasets[2], Vijay decides to arbitrarily DDoS www.srini.com and accidently mixes up the IP address 10.1.0.2 of www.srini.com with 10.1.0.1, which points to *node 4*. His attack is successful, albeit misguided, and Node 4 is no longer in the network :(. Soon after, *node 7* queries for *item 5*. List the nodes that will receive this query, assuming the the tables have had time to converge. Hint: keep ranges in mind.

Solution: 10 points The key here is that the entry at node 7 for i=2 will be (3,6), however 6 is greater than the item ID we are looking for, therefore we cannot immediately forward the query to 6. The query would go to the highest successor that does not exceed the item number which is 1. Node 1 has an entry for item 5 which is with successor 6. The query goes 7-¿1-¿6.

[2]http://icanhascheezburger.files.wordpress.com/2007/01/1159587965947.jpg