15744: Computer Networking

Homework 3 - TCP/NS + DNS

Date Assigned : 11/12/04
Date Due : 11/19/04 (in class)

1 Tools for querying NSs

In the following you will learn two useful tools, `nslookup` and `dig`.

1.1 nslookup

The `nslookup` command is a program that is used to query Internet domain name servers (DNS). Entering the command nslookup will give you the name of the name server your system knows and its IP address. The list of name servers used by a Unix machine can usually be found by looking at the file */etc/resolve.conf*. Read the man page for nslookup and answer the following questions.

Question 1: On which machine is the school's web server (www-2.cs.cmu.edu) running on? How many IP addresses does it have? Why is there more than one address and how are these addresses used ?

Question 2: It is possible to query the default name server to ask it to display all the records it contains for a particular domain. Log into one of the sun andrew machines (Some of these are: unix3.andrew.cmu.edu, unix4.andrew.cmu.edu, unix1, unix2, unix5, unix6, unix[12, 13, 14, 15]), use the nslookup program at /usr/sbin/nslookup (/usr/local/bin/nslookup does not work), and change the server to a nameserver for the cs domain. Then, at the prompt enter

```
ls cs.cmu.edu > output.txt
```

Approximately, how many DNS entries are there in the cs.cmu.edu domain?

Question 3: DNS also provides reverse mappings. Find out the domain name of 128.2.198.250.

Question 4: When you send mail to <somebody>@acm.org, which machine does the mail go to? What are the machines used to process mail sent to <somebody>@cmu.edu?

Question 5: How many name servers are there in the andrew.cmu.edu domain? What are their names and IP addresses?

1.2 dig

`dig` is another program that allows you to query DNS servers. For the purpose of this question, you should use the following format to invoke dig

```
dig +norecurse @<name.of.dns.server> <record-type><domain-name>
```

where

- `<name.of.dns.server>` is the hostname of the DNS server you wish to query such as A.ROOT-SERVERS.NET
- `<record-type>` is the type of DNS record you wish to retrieve, such as ANY, MX, etc.

- `<domain-name>` is the name of the host or domain you seek information on.

The DNS is a distributed architecture that uses hierarchical delegation. At the top of the system are the root name servers, who know which DNS server is responsible for each second-level domain (such as cmu.edu). If you send a root server a query for a particular machine, you will receive a reply listing the servers that have been delegated authority for that machine's second-level domain. It is common for a large domain such as cmu.edu to further delegate to departmental or workgroup DNS servers, which you discover by querying the second level servers.

In order to discover the chain of delegation in use at Akamai, run a series of NS queries for A121.G.AKAMAI.NET. You may wish to start with any of the 13 root servers ([a-m].root-servers.net), and you should continue your sequence of queries until you stop getting new delegations (in some domains this is indicated by a DNS server returning you a delegation pointing to itself, and in other domains this is indicated by a DNS server returning you a SOA record instead).

As an example here is the delegation chain for AOL.COM:

```
Server queried                           NS delegates to

B.ROOT-SERVERS.NET                       A.GTLD-SERVERS.NET
A.GTLD-SERVERS.NET                       DNS-01.NS.AOL.COM, DNS-0[267].NS.AOL.COM
DNS-01.NS.AOL.COM   DNS-01.NS.AOL.COM
```

Question 6: Generate the delegation chain for A121.G.AKAMAI.NET. Present your results in the table form shown above. Each NS query will typically return two or more answers: choose among them at random. If you query a server and get a timeout, choose an alternate server.

DNS is also used for reverse lookup, i.e. to translate IP addresses into hostnames. Again, the database is distributed in a hierarchical fashion, with a wrinkle. The most-specific part of a domain name is on the left (i.e. ux1 in ux1.sp.cs.cmu.edu), but the reverse is true of IP addresses (i.e. in 128.2.198.101, 128 is top-level, 128.2 is cmu.edu, and 128.2.198 belongs to cs.cmu.edu). Thus, address-to-name mappings is handled by reversing the bytes of the IP address and making queries in a special domain.To turn 128.2.198.101 into a hostname, various servers are sent queries seeking PTR records for 101.198.2.128.in-addr.arpa. The first query would be:

```
dig @a.root-servers.net PTR 101.198.2.128.in-addr.arpa
```

You will know when you're done when your query gives you back a PTR record in addition to (or instead of) NS record.

Note that you have to reverse the bytes in the address, i.e. start with the lowest level byte, e.g.,

```
dig @a.root-servers.net PTR 36.0.26.18.in-addr.arpa
```

```
Server Queried                NS delegates to

A.ROOT-SERVERS.NET            STRAWB.MIT.EDU
STRAWB.MIT.EDU                returns the PTR record
```

The hostname is mintaka.lcs.mit.edu

Question 7: Fill in a table like the one above showing a query chain for the IP address 128.2.209.197.

2 DNS

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:

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

Question 8: Harry's name server software has only two choices for TTL settings for A and NS records - 1 day and 1 minute. What are reasonable TTLs for the following records? Briefly explain you choice.

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

Question 9: 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?

Harry's Web site is especially popular among CMU students. The CMU network administrator estimate that there is one access from CMU every 6 minutes. Each access results in the application resolving the name www.distributed.hb.com. Assume the following:

- No other queries are made in CMU
- All CMU clients use the same local name server.
- Web browsers do not do any caching on their own.

Question 10: How many accesses will be made to the following name servers each HOUR to resolve these queries? Explain your calculation.

1. ROOT
2. ns1.hb.com
3. one of {east-ns, central-ns, west-ns}.distributed.com.

3 More tools .. *Extra Credit*

In the following you will learn two more useful network tools, `netstat` and `tcpdump`.

netstat is a tool that can be used to display network connections, routing tables, interface statistics and many other things.

Question 11: What does the command netstat -a show you? Explain the two parts of the output.

Question 12: What is the command to view the routing table of your machine using netstat? What is the command to only show IP addresses and not host names in the routing table?

Question 13: How can you use netstat to find out what the network interfaces of your machine are? What is the MTU of your Ethernet interface?

Question 14: How can you use netstat to find out how many TCP and how many UDP packets your machine has received and sent (since its last reboot)? What type of ICMP messages has your machine sent?

tcpdump enables you to see packets leaving and coming into your network interfaces. tcpdump may or may not already be installed on your machine. If it is not, you can obtain binaries as well as source from http://www.tcpdump.org. You may need to be root to run tcpdump.

Question 15: Run tcpdump and look at the output. Is all the traffic originating from or destined to your machine? Where does it come from?

Tcpdump allows you to apply some filters to the output. For example, commands ‘tcpdump ip‘ and ‘tcpdump tcp src port 80‘ tell tcpdump to show only IP packets, and to show only TCP packets from port 80.

For the next question run the following two commands one after another and compare the amount of output they produce.

```
tcpdump | grep arp
tcpdump arp
```

Now do the same for these two commands:

```
tcpdump | grep domain
tcpdump port domain
```

Question 16: Notice that the output of the first two commands is similar in volume. The output volume of the last two commands is quite different from each other. Why is this so?

Question 17: What does the -n option in tcpdump do? Can you think of a situation when this option might be useful?

Question 18: In this question you will see how tcpdump can be used to capture one's login name and password by monitoring an insecure ftp session. Go to the class homepage and download the trace of an ftp session, which was obtained using the tcpdump -w option. Read the man page of tcpdump to find out how to replay the trace and how to make tcpdump display the contents of the saved packets. Now use tcpdump and the trace file to find out which two machines are involved in this ftp session and what the username and passwd of the user running the ftp is.

4 More ns ...

In the following questions you will use ns to verify some of the things you have learned in class about TCP. You will also see how choosing parameters in your simulation setup can affect results.

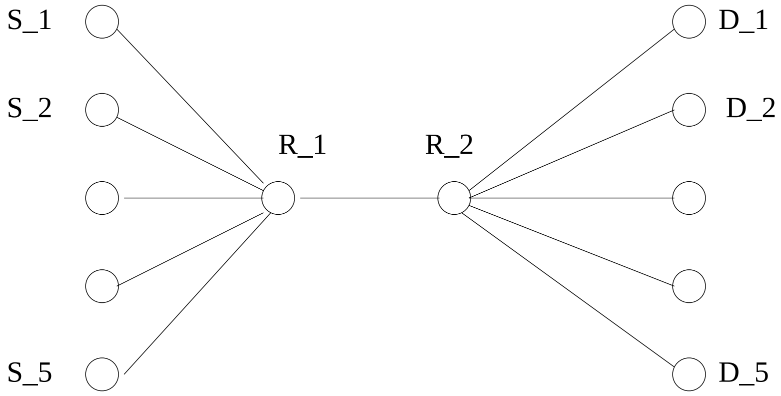

Figure 1: *The topology*

For the next question download the ns script Q11.tcl from the class web page. This script sets up the following ns simulation.

- It creates a dumbbell topology (as shown in Figure 1) with 5 nodes on each side. The bandwidth of the bottleneck link is 5Mbps and the bandwidth of all other links is 100Mbps. The delay on the bottleneck link is 100ms and the delay on all other links is 5ms.
- During the first 10 sec of the simulation, at random instants of time it starts one FTP source on each of the S_i nodes.
- It starts tracing the bottleneck link 20 sec after simulation begin.
- It stops the simulation after 120 sec.

Question 19: Run the script and compute the utilization on the bottleneck link. You will find that the link is not fully utilized. What do you think are the reasons? Change your script to increase the link's utilization as much as possible (without reducing the link speed). Describe the changes and give the resulting link utilization.

In the following question you will confirm the throughput equation for TCP, and at the same time learn how to use ns error models.

The error model in ns simulates link-level losses or errors by either marking the packet's error flag or dumping the packet to a drop target.

Question 20: Create an FTP source (on top of TCP Reno) and a receiver and connect them with a 10Mbps link with 100ms delay. Set the loss rate to 1%, and have the FTP source firing for 100 sec. Measure the achieved throughput. Repeat the same experiment for loss rates of 2, 3, 4 and 5% and plot the per-connection throughput vs the loss rate. In the same graph plot the theoretical curve. Compare the two curves and explain the result. (It helps to use a log-log plot and look a the slope of the resulting graphs).

Question 21: Repeat the experiments from the previous question with a link bandwidth of 100Kbps. Does your throughput follow the simple proportional to $\frac{1}{\sqrt{p}}$ relationship? It shouldn't! Why not?