Homework #1

OBJECTIVE

This homework assignment is designed to give you some hands on expertise with some basic networking tools. You will learn about traceroute, ping, dig and whois, all of which should give you some good insight into the operation of the network from the application level. You'll also exercise a bit of your analytic skills.

GUIDELINES

This is not a group assignment. You should work on this homework individually and write a report on your own. It is not acceptable to just look over your classmate's shoulder and gather the answers for your report. It is fine to discuss the homework with others, but you must do all the work involved in such a manner as to understand the exercises. In particular, if your answers are substantially identical to those of another student's answers, both of you will be considered for academic integrity violations.

The points associated with each problem are listed after the problem. Most will be graded in an "all-or-nothing" manner, so show all your work for full credit.

Page 2:	_____	(16 possible)
Page 3:	_____	(10 possible)
Page 4:	_____	(6 possible)
Page 5:	_____	(36 possible)
Page 6:	_____	(24 possible)
Page 7:	_____	(8 possible)
Total:	_____	(100 possible)

Write a report and submit it using the *Assignment* section of Blackboard before your class on the date of your lecture. Late submissions will not be accepted. No exceptions. The report should be in PDF format in a single file.

PART 1: ICMP TOOLS -- TRACEROUTE AND PING

In lecture 1 and textbook chapter 1.6.3, you learned about **`traceroute`**, a program used to reconstruct the path taken from source to destination, along with measurements of round-trip delays between the source and each intervening router. **`Ping`**[1] is another tool that issues ICMP "ECHO_REQUEST" datagrams. For this part of the homework, you don't need to understand all the technical details behind **`traceroute`** or **`ping`**, just the output it displays.

Follow these steps:

- Login to unix.andrew.cmu.edu using **`ssh`**.
- Read the manual page for traceroute and ping, i.e. **`man traceroute`**.
- Run **`traceroute`** without extra options, i.e. **`/bin/traceroute`** hostname.
- Find a domain name that is more than 15 hops away from you. Perform **`traceroute`** between unix.andrew.cmu.edu and this domain at 3 widely different hours of the day. At nearly the same time, perform a **`ping`** between unix.andrew.cmu.edu and the domain. Copy the output into your report (with timestamp on each sample).

Answer the following questions:

1. Find the number of **routers** in the path at each of the 3 runs. Did the paths change between runs or even during a single run? (5 points)
2. Try to identify the number of ISP networks that the traceroute packets pass through from source to destination. Routers with similar names and/or similar IP addresses could be considered as part of the same ISP. (2 points)
3. Approximately how long did it take to run the entire traceroute command? Why so much longer than the round-trip-time indicated by ping? (2 points)
4. What is the relationship between ping and traceroute? (2 points)
5. Simulate the first 3 steps of a traceroute query from unix.andrew.cmu.edu to your chosen domain, but only using ping (show your work). You won't have timing information, but do you get the same 3 intermediate machines as your

[1] Once you understand how ping works, check out this book review: http://www.amazon.com/review/R2VDKZ4X1F992Q/ref=cm_cr_rdp_perm.

previous traceroute tests? Just to be extra clear, for this question I'm asking you to use ping (perhaps with some command line options) such that the packets getting sent are substantially similar to those that traceroute would have sent. (5 points)

PART 2: DIG

During lecture 6, you learned about the Domain Name Service -- the internet's directory service.

The program **`dig`** (Domain Information Groper) submits DNS queries to name servers and presents the results in a human-readable format. It is gaining popularity because of its flexibility, ease of use and clarity of output. It was initially developed for gathering performance data and testing DNS servers. It is similar in functionality to **`nslookup`**, however, it provides much more detailed information. Many of you might be familiar with **`nslookup`** and might desire to use it for this homework. However, **`nslookup`** is a deprecated tool and does not always conform to the DNS protocol standard, so we will not be using it here.

Another good reason to use **`dig`** is that it is installed on CMU's Unix servers. Login to unix.andrew.cmu.edu using **`ssh`**, and you can read dig's manual page there (i.e. **`man dig`**).

You may also find the DNS RFC to be of use: www.ietf.org/rfc/rfc1035.txt for this exercise.

You will also need to learn a few things about reverse DNS. Check out RFC 2317 or the much, much more readable Wikipedia article (en.wikipedia.org/wiki/Reverse_DNS_lookup).

Answer these questions:

6. A DNS response message comprises four different sections: question, answer, authority and additional. Explain briefly the information each of the sections contains. (2 point)

7. What is the command-line option in `dig` to directly specify the name server to query? (1 point)

8. What does the "`-x`" command-line option in `dig` do? If you do not use the "`-x`" option, how would you achieve the same query? Use an example to illustrate. Why does this work? Will it work for EVERY IP address? Why or why not? (4 points)

9. Use the +trace option to query the CNAME record for www.cmuj.jp. Also, make sure "Additional Information" isn't displayed. Copy the output into your report. Then, write several sentences interpreting the various parts of the output, commenting on why each line was included and what it means. (3 points)

10. In lecture 6, I made the following claims. Use dig to verify each one. Make sure to show the command(s) you used and the output produced for each. Then, write a description or annotate the output to show how you know the answer supports or refutes each claim. (6 points)

    1. www-cmu-prod-vip.andrew.cmu.edu is our campus webserver

    2. you@andrew.cmu.edu email gets sent to andrew-mx-0[1-6].andrew.cmu.edu

    3. Email to you@cmu.edu was handled by cmu-mx-0[1-3].andrew.cmu.edu

PART 3: WHOIS

The whois database contains registration details of IP addresses and AS numbers originally allocated by the Regional Internet Registries (Each RIR maintains a database of their regional assignment). It shows the organizations that hold the resources, where the allocations were made, and contact details for the networks. The organizations that hold those resources are responsible for updating their information in the database.

The 3 important objects that these databases maintain are:

- Contact Information: A responsible administrative and technical contact for a network.
- Authentication Information: Contains authentication information about who can modify contents of a registration.
- And registered IP address space: the range of numbers, status, and responsible contacts.

Different RIR's store and maintain registration information differently. However the information obtained from whois servers will have similar basic information containing IP Addresses, AS numbers, organization name, contact information, etc.

If you are interested in further details about the RIR databases you can refer the RIPE database reference, Section 1, found at http://www.ripe.net/ripe/docs/

databaseref-manual.html and ARIN database reference at http://www.arin.net/reference/database.html.

For the purpose of the exercise in part 3 we will only be querying the whois servers provided by the Regional Internet Registries and the default whois server on CMU's Unix servers – whois.crsnic.net. Login to unix.andrew.cmu.edu using **`ssh`**, and you can read whois's manual page there (i.e. **`man whois`**).

Answer these questions:

11. List the 5 Regional Internet Registries (RIRs) along with the general geographic locations they are associated with. Also, list their whois server names. (3 points)

12. What is the command-line option in whois to directly specify the server to query? Why do you need to use this command? (2 points)

PART 4: PUTTING IT ALL TOGETHER

Using the knowledge of the tools you've just gained, complete the following questions. Cut-and-paste the command(s) you used to answer the questions, as well as the complete output of the command(s), to your report.

13. What is the IP address of the default local DNS server for CMU (from viewpoint of unix.andrew.cmu.edu)? (3 points)

14. Find the names and IP addresses of all root name servers. (2 points)

15. Using **`dig`**, make an educated guess as to what service Google is running on server(s) with IP address of **`8.8.4.4`**. (3 points)

16. What is the hostname of the default (local) nameserver? (3 points)

17. I mentioned in class that the root name servers do not support recursive requests. Prove it. Explain why or why not. (3 points)

18. Find the top-level name servers for the **`.beer`** domain. Which organization owns these name servers? What is the technical point-of-contact? (If you use multiple steps to find the answers, show each step in your report). (5 points)

19. List the IP addresses and names of all name servers and mail servers for the ECE department at CMU. (ECE's domain name is ece.cmu.edu). (5 points)

20. Find out details about Autonomous System Number 8. Include the OrgName, OrgID, ASName and the ASHandle in your answer. (4 points)

21. Find the IP address range assigned to CMU. (3 points)

22. Find the IP address of ece001.ece.cmu.edu

    a. Ask the C root server for the address of ece001 without recursion. (1 points)

    b. Go through the hierarchy from the root without recursion and following the referrals manually until you have found the address of ece001.ece.cmu.edu. Show all your work. (3 points)

    c. What is the address? (2 points)

    d. How many iterations did it take? (2 points)

PART 5: A FEW OTHER QUESTIONS

23. Refer to Lecture 01 (Networking Introduction), slide 30 for this problem. Suppose users share a 1Mbps link. Also suppose each user requires 100Kbps when transmitting, but each user transmits only 10 percent of the time. (8 points)

    a. When circuit switching is used, how many users can be supported? Let's call this number $n_{circuit}$.

    b. For a packet switching network, what is the probability that a given user is transmitting?

    c. Suppose there are 40 users. Find the probability that at any given time, exactly n users are transmitting simultaneously.

    d. Again, suppose there are 40 users. Find the probability that there are ($n_{circuit}$+1) or more users transmitting simultaneously.

24. Let's explore propagation delay and transmission delay. Suppose there are two hosts, A and B, connected by a single link with bandwidth R bps. The physical link's length is d meters. Propagation speed in the link is s m/s. At time t=0, Host A begins to send a packet of L bits to Host B. (8 points)

    a. Express the propagation delay, d_{prop}, in terms of d and s.

    b. Determine the transmission time of the packet, d_{trans}, in terms of L and R.

    c. At time $t = d_{trans}$, where is the last bit of the packet?

    d. At time $t = d_{trans}$, where is the first bit of the packet?

    e. Suppose $s = 2.8 \times 10^8$ m/s, L = 150 bits and R = 48 kbps. Find the distance d such that d_{prop} is equal to d_{trans}.

25. A webserver receives an average of 950 HTTP requests per second. 740 of them are simple GET requests for static pages, each of which takes, on average, 100 μS to reply to. 124 of them are GET requests for dynamic pages. Because responding to these requests requires at least one database lookup, they take 1 mS to process. The remaining are PUT requests that require database writes -- they take 2 mS to process. Model the webserver as an M/M/1 system. (8 points)

    a. Find λ, μ, ρ, L, Lq

    b. What percentage of the time is the webserver idle?

    c. How often are there more than 6 requests in the webserver's queue?

    d. What is the average waiting time for a request?

    e. What is the average total time for a request?