15-441: Computer Networks
Assignment 3

Lead TA: Seunghwan Hong (seunghwa@andrew.cmu.edu)

Due Date: November 13, 2007

Instruction: Please type or *neatly* handwrite a solution to each of the following questions. For full credit, please explain how you derived an answer: don't just give the final result. Papers are due at the beginning of the class on Tuesday, November 13, 2007.

In this assignment, we provide additional problems at the end of the assignment. These questions will not be graded and provided only for the purpose of exercise. But we will provide the solutions for the practice problems as well.

1 Domain Name Service (DNS)

The Andrew Linux machines provide a program **dig** that allows you to query Domain Name Service (DNS) servers around the Internet (some documentation is avilable if you type **man dig**). When running dig for the purposes of this question, you should use the following format:

- *dns server name* is the hostname of the DNS server you wish to query.
- *record type* is the type of DNS record you wish to retrieve, such as ANY and MX.
- *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 reponsible 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 elegated 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 can discver by querying the second-level servers.

(a) In order to discover the chain of delegation in use at CMU, run a series of NS queries for UX3.SP.CS.CMU.EDU. You may start with any of the root servers, 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). Delegation chain for: AOL.COM

```
Server queried                      NS delegations to
---------------                     ------------------
A.ROOT-SERVERS.NET                  A.GTLD-SERVERS.NET, K.GTLD-SERVERS.NET
K.GTLD-SERVERS.NET                  DNS-01.NS.AOL.COM, DNS-02.NS.AOL.COM
DNS-01.NS.AOL.COM                   DNS-01.NS.AOL.COM, DNS-02.NS.AOL.COM
```

This was produced by running the following commands:

% *dig +norecurse @a.root-servers.net NS aol.com*
% *dig +norecurse @k.gtld-servers.net NS aol.com*
% *dig +norecurse @dns-01.ns.aol.com NS aol.com*

Generate the delegation chain for UX3.SP.CS.CMU.EDU. 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.

(b) The DNS is also used 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., UX3 in UX3.SP.CS.CMU.EDU), but the reverse is true of IP addresses (i.e., in 128.2.203.134, 128 is ”top-level”, 128.2 is CMU.EDU, and 128.2.203 belongs to CS.CMU.EDU). Thus, address-to-name mapping is handled by reversing the bytes of the IP address and making queries in a special domain. To turn 134.203.2.128 into a hostname, various servers are sent queries seeking PTR records for 134.203.2.128.in-addr.arpa. The first query would be:

% *dig @a.root-servers.net* PTR 134.203.2.128.*in-addr.arpa*

You will know you are doen when your query gives you back a PTR record in addition to (or instead of) NS records. Fill in a table like the one above showing a query chain for the IP address 64.91.109.37.

```
Server queried                     NS delegations (or PTR record)
---------------                    ------------------------------
...                                ...
```

2 Tools like Route and Ifconfig

In this section, you will learn a couple of practical tools: **route**, and **ifconfig**, and **netstat**. Below is a very brief description of what they do. For more detailed information, check the man pages (Note on unix.andrew route and ifconfig are located under /sbin/. Either add this to your PATH or use the full path to run the commands).

route The `route` command can be used to view and manipulate the IP routing table.

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

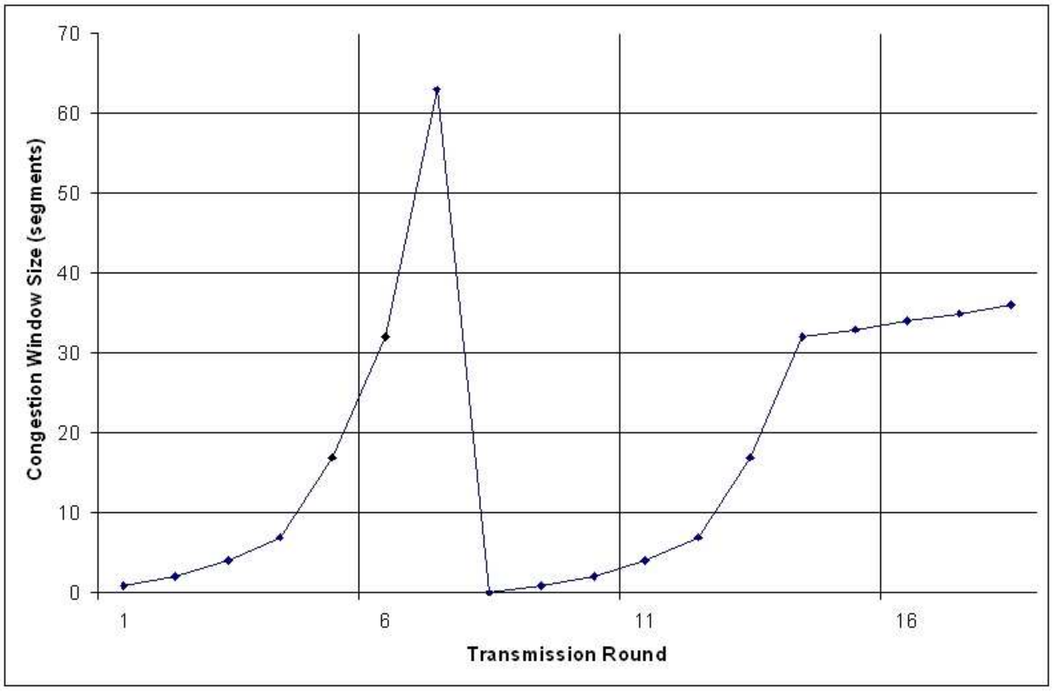
ifconfig The `ifconfig` command is used to configure a network interface and to display the status of the currently active interfaces.

1. Run the `route` command. What is the use of the entry with netmask 0.0.0.0?

2. Suppose a malicious attacker runs ”route del ” to delete the routing entry corresponding to 128.2.13.128/26 on one of the unix.andrew.cmu.edu machines. Now you try to run ”ping www.cnn.com”. Do you expect the ping to get through? Harry Bovik, the local networking guru argues that ”Well, there is still a default route, so the ping should go through!”. Harry’s pessimistic alter ego suggests otherwise, but does not have a reason. Should Harry believe his alter ego or dismiss it as a case of unjustified pessimism? Give a short 1-2 line answer why.
3. What does the command netstat -a show you? Explain the two parts of the output.
4. 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?
5. How can you use netstat to find out what the network interfaces of your machine are? What is the MTU of your Ethernet interface?
6. Run ifconfig. What information does the field ”Mask” give you?
7. What happens if you run ifconfig and configure an interface to be in promiscuous mode?

3 Congestion Control

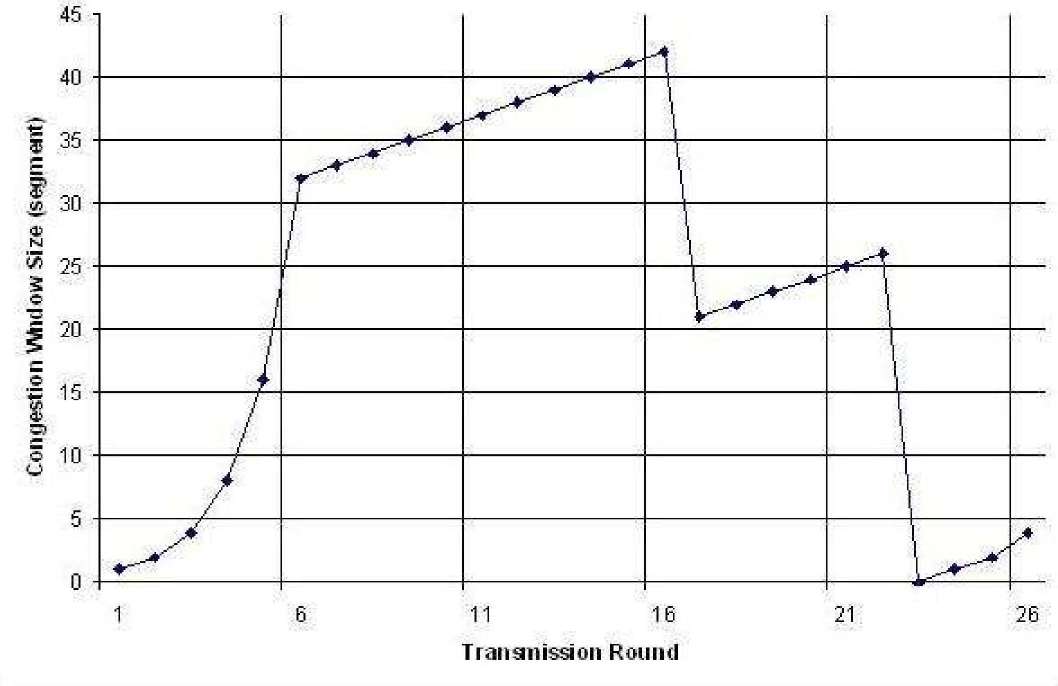
Transmission Control Protocol uses a method called **congestion control** to regulate the traffic flow. Often, a congestion control can be represented as a graph in which the x-axis indicates the time, and the y-axis indicates congestion window size. For each part, identify each of the following, if present:



(a) For the above figure, identify the intervals of time when TCP slow start is operating.

(b) After the 7th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?

(c) What is the ssthreshold during the 10th transmission round?



(d) For the above figure, identify all rounds during which TCP fast retransmits a packet. Assume TCP Reno.

(e) Identify the intervals of time when TCP is using additive increase congestion control.

4 Internet Service Provider (ISP)

The program *traceroute* is used to find a sequence of routers that a packet will follow from the source to a specific destination. The routers along the path are often identified by name, implying that you can learn the identity of the various ISPs that your packets travel through. You can type **man traceroute** to get more information.

The program *whois* contains information about various aspects of domain name and AS registration. For example, you can query the information:

whois -h radb.ra.net *IP ADDRESS*
whois -h whois.arin.net *ASN*

Now, consider the three hosts:

- www.cs.berkeley.edu
- www.alpinist.com
- www.ox.ac.uk

Using the two programs, you must answer the following categories for each question.

(a) Determine the AS number (ASN) associated with each of the routers along the path.

(b) Identify the name of the ISPs along the way.

(c) From the information, guess if each of the identified ISPs is local, regional, or backbone.

(d) What is the name of CMU's local ISP?

(e) Notice that CMU uses different backbone ISPs depending on the destination. Experiment with the routes taken by packets to different destinations (academic, commercial, national, international, etc). What observations can you make about which backbones traffic to different kinds of destinations is routed over?

5 TCP Forensics

You are the TCP specialist at the FBI. One day an FBI agent gives you a packet trace of a TCP connection between two machines on the Internet. The trace is believed to contain important information pertaining to national security.

This packet trace contains 63 packets and each line in the trace is one packet, identified by its packet number (from 1 to 63). The rest of the line in a sequence of bytes, represented as hex numbers. For example, consider the first line of the trace:

```
1                 45 00 00 3c fd b1 40 00 40 06 fd 45 80 02 8c ea 8c
                  d3 a6 04 8f 37 1a 0b b5 3c c0 85 00 00 00 00 a0 02
                  16 d0 88 c6 00 00 02 04 05 b4 04 02 08 0a 0c 6c
                  3d 34 00 00 00 00 01 03 03 07
```

The first number "1" (packet number 1) implies that this is the first packet received at the trace point.

The first byte of the packet is "45" in hex, which is "69" in decimal. As a hint, the first byte seems to indicate that this is an IPv4 packet, where the IP header contains 20 bytes. The fourth byte of the packet is "3c".

This trace, hw3.trace.txt, is available on the course web page under the Assignments link. To reduce the amount of work you have, we provide a template code tcptrace.c (also available on the course web page under the Assignments link) which parses the trace file into an array of bytes. You are free to use perl or any other tools (and even by hand if you wish) to analyze this trace file.

Please answer the following questions (and determine whether this is a threat to national security):

1. What is the client's IP address, the client's port number, the server's IP address, and the server's port number of this TCP connection? Based on the IP addresses of the client and server, what are their respective DNS names? You may assume that the computer which initiates the connection is the client, and the other computer is the server.

   Client IP: __________

   Client Port: ________

   Client DNS Name: __________________________

   Server IP: __________

   Server Port: ________

   Server DNS Name: __________________________

2. Which Internet application (i.e. web, FTP, gopher...) is running on top of this TCP connection?

   How do you arrive at this conclusion? (hint: information from your previous answers and Google can help!)

3. Using the TCP connection state diagram in Figure 6 of RFC 793, identify which packets (by their packet number) cause or result from the TCP state transitions of the TCP client? Your answer should be in the form: when a client receives packet X or user request Y, its TCP state is changed from A to B, and packet Z is sent (or some other action takes place). For example, when a client receives user request OPEN, the TCP state is changed from CLOSED to SYN-SENT, and a packet Z is sent.

4. Using the same format as used in (c), identify the state transitions with respect to the server.

5. Can you recreate the "crime scene" at the client's end user terminal? We do not want you to include the exact text in your solution, only a brief description. What is contained in packets 13 and 14's TCP payload, specific to the protocol? What was the client doing, who was the client looking for, and where was the client looking? (packets 32+)

6. BONUS(+2): Can you identify a piece of information in the trace that was clear text and would be considered insecure? Please state the piece of information verbatim.

7. Staple a printout of the source code, script, etc. that you used to answer this problem. This is evidence that you are indeed a qualified FBI TCP expert and did not "hire" someone else to perform this trace analysis.