**Project 1**: Compute per-hop Traceroute times given a tcpdump text trace  
  
*Type of Project*: Only Individual  
*Deadline*: 2012-01-16, 11:59pm  
*Language*: Java  
*Points*: max 10 points  
  
*Submission Guidelines*: Submit through nike.cs.uga.edu, as usual. Name the directory project as "LastName\_FirstName-tcpdump\_traceroute\_analysis" (e.g., "Perdisci\_Roberto-tcpdump\_traceroute\_analysis"). Submit ONLY the source code (source code must be inside project dir) using   
  
$ submit   LastName\_FirstName-tcpdump\_traceroute\_analysis   cs4760  
  
The program must be named "TracerouteAnalysis.java", and needs to take the file name of the tcpdump trace as the first argument. For example, the program needs to run as  
  
$ javac  TracerouteAnalysis.java  
  
$ java  TracerouteAnalysis  Project\_1-example\_traceroute\_TCP\_trace.txt  
  
and the results of the analysis must be written on ***standard output*** (basically, on screen).  
  
NOTE: project submissions that do not follow the guidelines will be discarded wihtout consideration (i.e., 0 points).  
 *Project Description*: In this project, you are required to write a program that takes in input a textual tcpdump trace of traffic generated by Traceroute and computes the time between a TCP packet sent by the client and the related ICMP "Time exceeded in-transit" message.  
  
As an example, consider the two packet logs reported below:  
 **1291654312.963163** IP (tos 0x0, **ttl 1**, **id 9067**, offset 0, flags [none], proto **TCP** (6), length 60) **128.192.76.177.47212 > 137.138.144.168.80**: S, cksum 0xc4d6 (correct), 1135826272:1135826272(0) win 5840 <mss 1460,sackOK,timestamp 2152510109 0,nop,wscale 2>  
  
**1291654312.963644** IP (tos 0xc0, ttl 255, id 2503, offset 0, flags [none], proto **ICMP** (1), length 56) **128.192.76.129** > 128.192.76.177: **ICMP time exceeded in-transit**, length 36  
    IP (tos 0x0, ttl 1, **id 9067**, offset 0, flags [none], proto **TCP** (6), length 60) **128.192.76.177.47212 > 137.138.144.168.80**:  tcp 40 [bad hdr length 0 - too short, < 20]  
  
as we can see from the highlighted fields, the two packets are related to each other.   
The fields you should check to match a sent TCP packet with the related ICMP response are: **id 9067, TCP, 128.192.76.177.47212 > 137.138.144.168.80**  
Notice that these fields are replicated in the body of the ICMP message (in practice the IP ID field should be sufficient to correctly correlate the two packets).  
  
From the two packets above, the output should be:  
  
**TTL 1  
128.192.76.129  
0.481 ms**  
where *128.192.76.129* is the IP addresses of the router that generated the ICMP response, and *0.481 ms* is computed as (1291654312.963644 - 1291654312.963163) \* 1000 and rounding to obtain only three digits after the dot.  
  
As another example the output related to the following packets  
  
**1291654312.963267** IP (tos 0x0, **ttl 3**, **id 9075**, offset 0, flags [none], proto **TCP** (6), length 60) **128.192.76.177.56812 > 137.138.144.168.80**: S, cksum 0x5834 (correct), 2778675862:2778675862(0) win 5840 <mss 1460,sackOK,timestamp 2152510109 0,nop,wscale 2>  
  
**1291654312.963655** IP (tos 0x0, ttl 62, id 47385, offset 0, flags [none], proto **ICMP** (1), length 56) **128.192.254.49** > 128.192.76.177: **ICMP time exceeded in-transit**, length 36  
    IP (tos 0x0, ttl 1, **id 9075**, offset 0, flags [none], proto **TCP** (6), length 60) **128.192.76.177.56812 > 137.138.144.168.80**:  tcp 40 [bad hdr length 0 - too short, < 20]  
  
would be  
  
**TTL 3  
128.192.254.49  
0.388 msTTL 3  
128.192.254.49  
0.388 ms  
0.401 ms  
0.398 ms**  
  
test1 [[tcpdump\_output](http://www.cs.uga.edu/~perdisci/CSCI4760-S12/Assignments/Project_1/bbc.txt), [analysis\_output](http://www.cs.uga.edu/~perdisci/CSCI4760-S12/Assignments/Project_1/bbc.out)]  
test2 [[tcpdump\_output](http://www.cs.uga.edu/~perdisci/CSCI4760-S12/Assignments/Project_1/cern.txt), [analysis\_output](http://www.cs.uga.edu/~perdisci/CSCI4760-S12/Assignments/Project_1/cern.out)]  
test3 [[tcpdump\_output](http://www.cs.uga.edu/~perdisci/CSCI4760-S12/Assignments/Project_1/italia.txt), [analysis\_output](http://www.cs.uga.edu/~perdisci/CSCI4760-S12/Assignments/Project_1/italia.out)]  
test4 [[tcpdump\_output](http://www.cs.uga.edu/~perdisci/CSCI4760-S12/Assignments/Project_1/japan.txt), [analysis\_output](http://www.cs.uga.edu/~perdisci/CSCI4760-S12/Assignments/Project_1/japan.out)]  
  
  
To verify your results, I will compare your output to the correct one using the diff command, for example  
  
$ diff  correct\_output.txt  your\_output.txt

if any difference is found, your output will be considered wrong.  
  
Your software will NOT be tested on the example traces. To make sure your software works as required on other traces, you can generate them on your own by running, for example  
  
# tcpdump -tt -i eth3 -p -n -nn -v icmp or tcp  
  
# traceroute -T -p 80 -n www.cern.ch  
  
but make sure to use the following versions of traceroute and tcpdump (e.g., on Ubuntu 10.10)   
  
$ traceroute --version  
Modern traceroute for Linux, version 2.0.14, May  9 2010  
Copyright (c) 2008  Dmitry Butskoy,   License: GPL v2 or any later  
  
$ tcpdump -h  
tcpdump version 4.1.1  
libpcap version 1.1.1