Computer Networks CSE 5344
Packet Sniffer Grading Rubric

Instructor: Mike O'Dell
GTA: Jees Augustine

April 2015

"The reason I can't figure out why it fails is because I don't understand how it ever worked to begin with." ***Van Jacobson***

Problem Statement

Packet sniffer applications is the Swiss Knife in a network engineers toolkit. This helps an engineer to analyze the packets, isolate the network issues, detect the intrusions, monitor the resource utilization, test and troubleshoot the protocol implementations, gather network statistics and many more.

As a Computer Networking student, all are now familiar with the network protocols. The only way to deepen the understanding of network protocols is by seeing them in action. This project will you give you a chance to reinforce the learning by putting into practice your knowledge on network protocol stack and packet headers. The expected outcome of the project is to give you a first class hands-on experience on real networking elements. Henceforth we define the Problem to be stated as follows,

To design and implement a full fledged packet sniffing application in Python[1]*, which enables a user to capture the packets and to gain insights about the underlying network by providing the user with the results of a detailed analysis*

Protocols Explored

The set of protocols that you will explore during the course of the project are,

Application Layer: HTTP

Transport Layer: TCP, UDP

[1] This is our best choice; however, other options can be negotiated

The Network Layer: ICMP

The Link Layer: ARP

The availability of all these information about packets is a reasonable assumption to make as the sniffer (your application), sniffs out packets at the lowest possible level in a computer, the raw socket. Once you have the packets available at your disposal(both ingress and egress), the low level socket library will enable you to obtain substantial amount of information by unwrapping each packet.

After having correctly identified and categorized the packets, we have access of the multitudes of information specific to a host, destination, route it had taken and underlying network from the packet. To mention some of them will be,

- *Source IP (IPV4)*
- *Destination IP (IPV4)*
- *Total Size of Data*
- *Time-to-Live*
- *Source Port*
- *Destination Port*
- *Sequence No (Protocol Dependant)*
- *Acknowledge No (Protocol Dependant)*
- *window size (Protocol Dependant)*
- *MAC address*

Expectation

This section is used predominately to synchronize the thought process of assignor and the student.Your packet sniffer should have the following functionality if it needs to be considered for evaluation.

Your Applications should,

- Capture packets from *'the wire'*
- Have the ability to listen to multiple interfaces (Ethernet and Wireless, 802.11)
- Capture the *'live'* packets (It should be real time)

- Display necessary details about a captured packet
- Have the ability to filter (dynamically) the packets based on protocol (by user request)[2]
- Calculate the throughput of the connections [3] and render the results as a graph using any plotting utility. (Verify your results with the graph given in the text book as a part of the analysis.)
- Collect different congestion window sizes from the TCP packets and plot a graph against time. (Verify your results with the sawtooth graph given in the text book as a part of the analysis)
- Calculate the average Packet/Frame sizes over a capture session[4]
- Calculate the diameter of the network that you have probing [5]
- Have a user interface[6] to issue commands (Start, Stop, Filter, Display throughput, Display Congestion Window Vs Time Graph)

You may use any operating system of your choice for the completion of the project.

Submission Details

Each team has to make a unique submission towards the final assessment. A team will collect all the project related files and documents in a single folder and zip/rar the file with a name,

PA3_Packet_Sniffer_LastName1_LastName2_LastName3_LastName4.zip

and send the zip/rar to us (List the names in alphabetical order) by email. Add a *readme.txt* file which mentions all the necessary information about your project. Send the zip/rar file copying all your teammates with a subject line,

[Submission] Programming Assignment 3 : Packet Sniffer

Please click on this if you are in a hurry and I have embedded the address and subject line for you.

After the submission you will have to sign up for a project demonstration session. This session will be brief one spanning 30 minutes and will be scheduled the week following the submission and slots will be filled based on first-come-first-serve basis. This will be the link for choosing your slots. There are twelve

[2] You can do filtering based on interfaces as well, this will be a feature for additional credit

[3] There are multiple ways of doing it, Pick yours or Pique your brain. Refer Appendix A for more details

[4] See Appendix B for details

[5] See Appendix C for details

[6] Command Line or Window based or Browser based

slots provided and sent me email mentioning the Slot number after you pick a slot so as to freeze the slot for any further edits. If you face any issues in booking a slot or for any general queries please email me here (Subject line is already embedded for you).

Evaluation and Weightage

Your packet sniffer will be tested against real traffic during the demo session and evaluation will be based on the points mentioned in the Expectation section. There are two separate evaluation sections for your project,

Design and Implementation (60%) In this section all the design, implementation and parsing efforts are accounted.

- Capturing the Packets from raw sockets(15%)
- Parsing through the packets and extracting information (15%)
- Display of necessary information obtained from packet (15%)
- Filtering of the information based on protocols (5%)
- Capturing from Multiple interfaces (5%)
- Using the flexibility to derive any attributes of your choice [7] and use of novelty in implementation(5%)

Analysis (40%) Grading on this section will be based on the analysis that you have performed with the available data (Throughput, Congestion Window size, Diameter of the network and the Average File Size measures). The accuracy of the measurements, concurrency of your graphs with the expected behaviour, GUI Design are all covered in this section. The finer split will be as follows,

- Graphs (20%): 10% (Throughput), 10% (Congestion Window)
- Average Packet/Frame size (5%)
- Average Diameter of the network and Maximum Diameter of the network (5%)
- Accuracy and Concurrency (5%)
- GUI (5%)

Appendices

A How to calculate the Throughput

In addition to delay and packet loss, another critical performance measure in computer networks is end-to-end throughput. There are multiple ways of cal-

[7] This is just make you think creative, and to derive any network attribute from the information available or how creative are you presenting the results

culating this; however, We will mention two ways of doing it here,

Proactive Mode: In this mode you will try to send a file of known size of the same size multiple times (1000 times or more) to a known server or any application that establish a TCP connection. Repeat the experiment for different file sizes (500 or more different file sizes). Don't fall pray for the fallacy of sending different files, even of same file sizes during the estimation of throughput for a single file size[8]. You may repeat same set of experiments with a single file of different sizes. You know the time you have taken for transferring the file. Now you can calculate the throughput based on these data available. You *may* calculate either ***Instantaneous Throughput*** or ***Average Throughput*** . Refer section 1.4.4 of the textbook for more details.

Reactive Mode: You have maximum segment size *MSS* available in a packet. In each connection you can estimate the Round Trip Time (RTT) by measuring the time difference between the *(SYN)* and *(SYN, ACK)* [9] for the same TCP connection. Now throughput could be calculated by the formula given below. You get the throughput of a connection faster by this and with lesser effort.

$$Throughput = MSS/RTT$$

Choose your own method: Pique your brain for a method to measure the throughput of the connection. This is an excellent opportunity for you to explore a bit more and propose any novel approach to find thoruput and you could get a paper published in this area.

You may choose of the methods mentioned above to find the throughput of the underlying network.

B Average Packet/Frame Size Over a Capture Session

Once you have all the information available about a packet, you can derive some additional attributes like average packet and frame size by collecting details over large number packets. You have all the necessary information available to calculate this. You may know the total packet size and frame size from each packet. Collect all of them in some data structure and produce the average packet and frame size in the network as the output.

[8] Give a thought on why is this wrong in terms of probability. Whats the random variable if you vary files each time ?

[9] This is the best estimate of the network conditions at this point of time

C Diameter of the Network

You can really check how far do you navigate in a network in your usual network activities. This is an interesting estimate and a vindication of frugal allocation in *8 bits* for TTL. Look at your *Time To Live (TTL)* values to get a insight about this idea. Find and publish the maximum and the average values of it. The maximum defines the Total Diameter of the network and the average defines *'your small world'* that you interact with. Remember that you have to publish not the TTL itself, but the complement of TTL (We hope you understand what we mean by complement)

Sorry for allocating this Project late, If you were completing the project in 1990's your whole package would have been procured by any networking company by a hefty tag not less than $1500. Add a 20% inflation from then and you could have been a CEO ☺