Problem Statement: Design and prototype a complete packet-sniffing tool to analyze packets.

Objective: To put into practice your understanding of the essential network protocols and packet structures discussed in class.

Outline: Throughout the course we have used a packet analyzer, Wireshark, which is a useful tool kit for network administrators, protocol implementers and novice computer networking students. In your homework assignments so far you have viewed the functional capabilities of Wireshark as a black box. This programming project will give the student an opportunity to create a functional packet-sniffing tool that analyzes packet directly from the network interface of the host computer. To successfully complete this project, you must build a basic packet tracer utility, capable of displaying different protocols and encapsulations.

Challenges
1) Capturing packets from native interfaces. For the purpose of this implementation you can focus on the wired Ethernet interface and the wireless interface.
2) Processing the packets and extracting the details. You should identify the protocol headers and application data separately.
3) Starting and stopping packets based on command. This task may seem trivial but remember you have multiple interfaces to handle simultaneously.
4) Analyzing the throughput.
5) Displaying the filtered results. Packet details, rates and other information about packet flow must be displayed a graphed in a visually appealing and useful way.

Prerequisites
1) Good understanding of socket programming in Python 2.7
2) Good understanding of the network protocol stack (OSI)
3) Good understanding of packet encapsulation protocols and technologies
4) Good understanding of the network interfaces (Ethernet, IEEE 802.11)
5) Ability to leverage the power of regular expressions for parsing the sniffed packets.

Nature of Project, Tips for Success and Team Composition
This project will require significant time and effort to conceptualize, design, code, test and iterate through prototypes. It expected that it will require several weeks of effort from a typical team of 4 students. The earlier you start the better the resulting program will be. After the successful completion of the basic
functionality you should enhance your program to incorporate visual elements to display the packets, throughput variations, congestion window size variations, etc.

Start by analyzing and understanding Wireshark. You should identify some of the drawbacks of the packet sniffing tool used in your homework assignment and try to improve on them. This will be a great opportunity to address the capabilities of a commercial software program while giving you a more complete exposure to the protocol stack that we have studied in class.

Taking into account the complexity of this assignment, you must select a team of no less than three (3) and no more than four (4) students in your section to work with on this project. All students on a team will receive the same grade for this assignment, so work should be apportioned accordingly.

If you have any questions about this assignment, please contact Jees Augustine, 5344-001 GTA, for clarification.

Submission Details

1) Select your team and submit names of teammates to your GTA by email (copy to Professor O’Dell) no later than Thursday, March 19, 2015.
2) Submit your completed project source code to your GTA by email no later than Tuesday, April 21, 2015.
3) Include a brief write-up with your submission that discusses the highlights of your project, execution details and sample results that you have obtained during final testing.
4) Be prepared to demonstrate the operation of your application if requested.
5) Maximum credit cannot be achieved on this assignment unless your program operates as specified and handles typical error conditions properly.