ECE 158B Spring'23 Project Assignment 1

University of California San Diego

Submission deadline: 6pm, Wednesday 05/15/2024.
Instruction: Choose one of the following two project topics. For each topic, you can work together in a group with up to 3 people. Feel free to post messages on the Canvas discussion forum to find project a partner. Each team only needs to submit one version of your report and deliverables. Include the names of all team members in the report. Alternatively, you can propose your own topic; If so, the workload must be similar to Topic 1 or 2, and you must get the instructor's consent in advance.

Topic 1: In this project, you will investigate the traffic patterns of different application-layer protocols. You can capture protocol-specific packets using Wireshark, and then parse the captured packets to examine their properties. Follow the steps below to complete the project, and submit your source code along with your report.

(1) Run Wireshark to capture packets generated by four different protocols: HTTP, FTP, VoIP, and BitTorrent. Each capture should last for at least 1 minute. During the capture, try to stop all other irrelevant applications that may access the Internet. Save the captured packets in *.pcap* files (the default file format in Wireshark).

- HTTP: capture the traffic when you visit a website, e.g., news.yahoo.com, and its sub-pages. Try to mimic the click behavior when you normally browse that website.
- FTP: run an FTP file-downloading session for 1 minute, using an FTP client such as FileZilla or WinSCP. Google "public ftp server" and you will find many free ftp servers online, e.g., https://cs.brown.edu/about/system/services/ftp/

  You can use other similar ftp sites.
- VoIP: start a VoIP call, e.g., using Skype or Zoom, with some remote host, and capture the traffic. Again, make it a real VoIP call, in which speech and silent periods interleave.
- BitTorrent: run a BitTorrent session (you'll need to install a BitTorrent client software first. Google "BitTorrent" to figure out where to download a client.). Download files that are NOT copyrighted, e.g.,

  http://www.ubuntu.com/download/alternative-downloads

(2) (30pt) Extend the python dpkt file parser library: https://github.com/kbandla/dpkt so that the payload size and arrival time of all packets can be parsed based on the .pcap file you saved. The *examples* folder in the dpkt library shows how to extract such statistics. dpkt has been well documented here: https://dpkt.readthedocs.io/en/latest/ Visit the following website if you want to understand the format of the packet captured by Wireshark. http://www.tcpdump.org/pcap.html

(Note: Relevant information starts from "The pcap_pkthdr structure is defined in…")

In your report, explain how you get the arrival time and payload size of a specific application that you want to focus on (HTTP, FTP, VoIP, BitTorrent). Make sure your code is well commented. **Submit your report in pdf and your code in a separate zip file**.

(3) (70pt) For each of the above 4 applications, plot:

- The cumulative distribution function (CDF) and probability density function (PDF) of the captured payload sizes. Matlab is a convenient tool for plotting CDF and PDF.
- The cumulative distribution function (CDF) and probability density function (PDF) of the packet inter-arrival time, i.e., temporal separation between two consecutive packets.

Answer the following questions:

- How does the payload size distribution of different applications differ from each other (e.g., you may see different peaks in the PDF)? Explain what causes the difference.
- Similarly, explain your observations of the inter-arrival time distribution.

Topic 2: Investigation of BitTorrent through a simple simulator.

Build a simple simulator to simulate a BitTorrent network with hundreds of nodes. Follow the example BitTorrent network topology in lecture 8. Use the simulator to investigate the following properties of BitTorrent.

1) Resilience: How effective can BitTorrent handle peer churns?
2) Effectiveness under peer heterogeneity (different peers have different access network speed)
3) Scalability: As more peers join the network, how does per-peer downloading rate improve?
4) Effectiveness against free-riders.
5) Fairness (It is up to you to define fairness. One simple metric: a peer who contributes more uploading bandwidth should get higher downloading rate as well).

Hint: The above are open-ended problems and it is up to you to perform the simulation and analyze the results under various possible network settings. Use your best judgment to decide what are the practical network settings.

It does not matter how you build the simulator, e.g., what programming language you use --- the conclusion from your investigation is the only important thing in this project. Your simulator does not have to simulate every aspects of a network (e.g., you do not have to simulate TCP/UDP/IP). Try to make assumptions to simplify your task. For example, instead of following the BitTorrent protocol exactly, just assume a centralized algorithm is running the key modules on behalf of each peer, e.g., tit-for-tat peer selection and chunk selection.

Grading criteria: Correctness and completeness of the simulation code (20pt); investigation of the 5 metrics (16pt each).