Assignment 4: Traffic analysis (15%)

Fall 2023 – CPSC 441

Due at 23:59, Dec. 6 on D2L

Assignment policy:

- This is an **individual** assignment, so the work you hand in must be your own. Any external sources used must be properly cited (see below).
- Extensions will not be granted to individual students. Requests on behalf of the entire class will only be considered if made more than 24h before the original deadline.
- Some tips to avoid plagiarism in your programming assignments:

  1. Cite all sources of code that you hand in that are not your original work. You can put the citation into comments in your program. For example, if you find and use code found on a web site, include a comment that says, for example:

     ```
     # the following code is from https://www.quackit.com/hello_world.cfm.
     ```

     Use the complete URL so that the marker can check the source.
  2. Citing sources avoids accusations of plagiarism and penalties for academic misconduct. However, you may still get a low grade if you submit code that is not primarily developed by yourself.
  3. Discuss and share ideas with other programmers as much as you like, but make sure that when you write your code that it is your own. A good rule of thumb is to wait 20 minutes after talking with somebody before writing your code. If you exchange code with another student, write code while discussing it with a fellow student, or copy code from another person's console, then this code is not yours.
  4. Collaborative coding is strictly prohibited. Your assignment submission must be entirely your code. Discussing anything beyond assignment requirements and ideas is a strictly forbidden form of collaboration. This includes sharing code, discussing code itself, or modelling code after another student's algorithm. You cannot use another student's code, even with citation.
  5. We will be looking for plagiarism in all code submissions, possibly using automated software designed for the task.
  6. Remember, if you are having trouble with an assignment, it is always better to go to your TA and/or instructor to get help than it is to plagiarize.

Background:

In this assignment, you will be exploring the dynamics of Internet video streaming traffic and its impact on network performance. You will be writing a short program to simulate a router queue, and then using this to make an informed decision about how best to improve network quality of service (QoS).

In this scenario, you are the head network administrator at Hoppy Homes. Everyone at the company loves fighting games, and today happens to be the day of the EVO championships. Both you and your boss Fred know that many of the employees who actually came to work today will spend most of their time watching the matches, doing so using their wireless laptops. Your boss is worried about the impacts of the video streaming traffic on the QoS for work-related Internet traffic (or perhaps the impact of work-related traffic on the QoS for the streaming!). For this reason, your boss has generously offered use of the boardroom for watching the tournament, so that only one video stream is needed.

To be honest, Fred is quite the miser, and has made minimal investment in the company's Internet infrastructure. Specifically, there is a 100 Mbps (Megabits per second) external link for incoming Internet traffic, but only a 5 Mbps Wireless LAN (WLAN) internally for distributing Internet traffic to the employees. The capacity of the WLAN is certainly going to be the performance bottleneck. Furthermore, the access point on the WLAN has only B=100 buffers available to store incoming IP packets that are waiting for transmission over the WLAN.

You are definitely worried about the QoS on your network during the game. Fortunately, you were prescient enough to collect a Wireshark packet trace of last year's tournament when you were watching at home. You will use this trace of IP packet timestamps and sizes (in bytes) to learn about video streaming traffic, and what it might do to your company's network. You also have a Zoom trace from a recent work-related Zoom meeting that you can use for further experiments and comparison.

Instructions:

Your task is to write a program in C or C++ that can estimate the QoS for incoming video streaming traffic on your network. The two QoS metrics of interest are the **average queueing delay** for packets in the AP buffer, and the **packet loss percentage** at the wireless AP. There is a simple tradeoff between these two metrics. That is, with a tiny buffer, there will be minimal queueing delay, but there might be really high packet loss. Conversely, with a huge buffer, there will be minimal packet loss, but the packet queueing delays might be really large. Your goal is to find the "sweet spot" for the buffer size that provides a good tradeoff between loss and delay.

You have been provided with two trace files on D2L for this purpose. One contains packet timestamps and sizes for video streaming traffic, and the other contains the same data for a Zoom call. You will need to create a program that reads in the data from a trace file, and then tracks how many packets are in the queue at each point in time.

This assignment is essentially a discrete-event network simulation problem (although you could also model the system dynamics approximately using even simpler approaches, such as a time-driven simulation, a fluid flow model, or some differential equations). You'll

want to implement some sort of loop that models the arrival and/or departure of each packet, and then use this loop to track the current occupancy of the router buffer. Packets arrive at certain times, based on the trace file, and depart at certain times, based on the packet's size and the wireless link transmission rate. After running the simulation, your program should output (write to terminal and/or a file) a statistical summary that shows the number of incoming packets, the number of delivered packets, the number of lost packets, the packet loss percentage, and the average queueing delay for (delivered) packets. You will be changing the values of B (the buffer size) and the WLAN capacity in your simulations, so make sure these parameters are easy to change in your code.

You may assume that the queue is managed in a First-In-First-Out fashion. Please ignore all other WiFi details, such as MAC-layer ACKs and retransmissions. You may also assume that only one video stream is required, since all employees can watch on the big screen in the boardroom.

Use your program to answer the following "what if" questions about the network QoS:

1. For the 5 Mbps WLAN, what would the QoS be if the buffer size was B packets? Vary B over some reasonable range of values, such as from 0 to 1000 in steps of 100 (starting at B = 1). Show a graph of these results, and suggest a suitable value of B that provides a good tradeoff between delay and loss. Do this separately for both the video streaming trace and the Zoom call trace to see which one is more challenging for your network to handle. Show both results, and comment on this in your writeup.

2. If the buffer size was fixed at B=100 packets, how would the WLAN capacity affect the QoS? What would the QoS be for each of these two individual video streams if the WLAN capacity was 5 Mbps, 6 Mbps, 8 Mbps, and 10 Mbps? Use a tabular format for reporting these results, with one row for each of the four cases, and do this for both traces.

From these results, you should be able to determine whether it is better to buy buffer space or bandwidth.

When you are finished, submit your solution in electronic form to the dropbox on D2L. You should include the code file(s) for your simulation and a brief report presenting the results (2–3 pages should be sufficient, but there are no strict requirements). Your report should as a minimum include the following:

- Your name and student ID
- A brief summary of how your simulation works (up to 1 page), including code snippets where appropriate. The goal is not a line-by-line walkthrough, instead, you should answer some broad questions about your approach. For example, what does your main loop iterate through? Are you modelling packet arrival or departure? Where do you track packet loss? Where is the WLAN bandwidth used in your code?
- Your answers for both key questions above (including the graph/tables) and a short comment on the results.

- A conclusion that clearly states whether it would be better to purchase additional buffer capacity or WLAN bandwidth to best improve QoS.

You do not need to upload the trace files, since the TAs already have these.

Rubric (40 points total):

- **16 marks** for design, implementation, and functionality of a properly documented simulation program.
- **8 marks** for your graph of the results from your buffer size experiments. Both axes of the graph must be clearly labelled, including quantity measured, numeric values, and units as appropriate.
- **8 marks** for your table of QoS results for B=100.
- **8 marks** for a well-written, clearly structured, readable report, with relevant comments and a justified conclusion.

Tips:

- The trace files are quite large, so you might want to use just a small snippet of the trace at first as you learn how to read in and process the file.
- Remember that the packet sizes are in **bytes**, and the bandwidth is in **Mbps** (megabits per second), so don't be off by a factor of 8.
- I recommend creating a dummy input file with easy values (up to four or five entries) that you can test by hand to verify that your simulation is working properly.
- Your main task will be figuring out how to model the buffer occupancy. Make sure you are clear on what you need to do here before you start coding.
- Your program does not need to be perfectly accurate. It suffices to have good estimates of the results so that you can compare the configurations quantitatively.
- If you have never written a simulation program before, then you should plan to attend the tutorials on simulation methods. These will take place on Nov. 20–21.
- Start on this early. As a minimum, make sure you understand clearly what you need to do and that you have some pseudocode for the main simulation loop. Things always get busy at the end of semester, so make sure you get help with anything you need while you can.

---

author: jcleahy@ucalgary.ca