Homework 1: Packet Delay + Measurement

Muyuan Zhang

2023-01-23

Question 1: Internet Delay

What is the average queuing delay experienced by the 1000 byte packets?

- (11 + 10 + 2.8 + 3.0 + 5.5) / 5 - 2.8 = 3.66
- Therefore the average queuing delay experienced by the 1000 byte packets is 3.66 ms.

What are the reasonable estimates of transmission and propagation delays experienced by a packet of size 600 bytes sent through the link K?

- The transmission delay is $ d_{trans} = L/R $. Therefore, the transmission delay experienced by a packet of size 500 bytes is 2.8 - 2.4 = 0.4 ms. Then we can get that the transmission delay experienced by a packet of size 600 bytes is 0.4 * 600 / 500 = 0.48 ms.
- The propagation delay experienced by a packet of size 600 bytes should be same as that of size 500 bytes, which is 2.4 - 0.4 = 2 ms.

Question 2: Traceroute

a Execute the traceroute command to one destination of your choice, at least 10 hops away. Try to pick a destination for which you get as complete data as possible and that the traceroute program terminates normally. Collect the traceroute data in a file (using output redirect on the Unix-shell).

Write a program to parse each line of this file and compute the average delay (averaged over the three delay values) for each hop. Your program should save the IP address of each hop (the address within parentheses in each line of traceroute data) and the corresponding average delay in another file such that each row of this new file contains the IP address of the hop and the average delay.

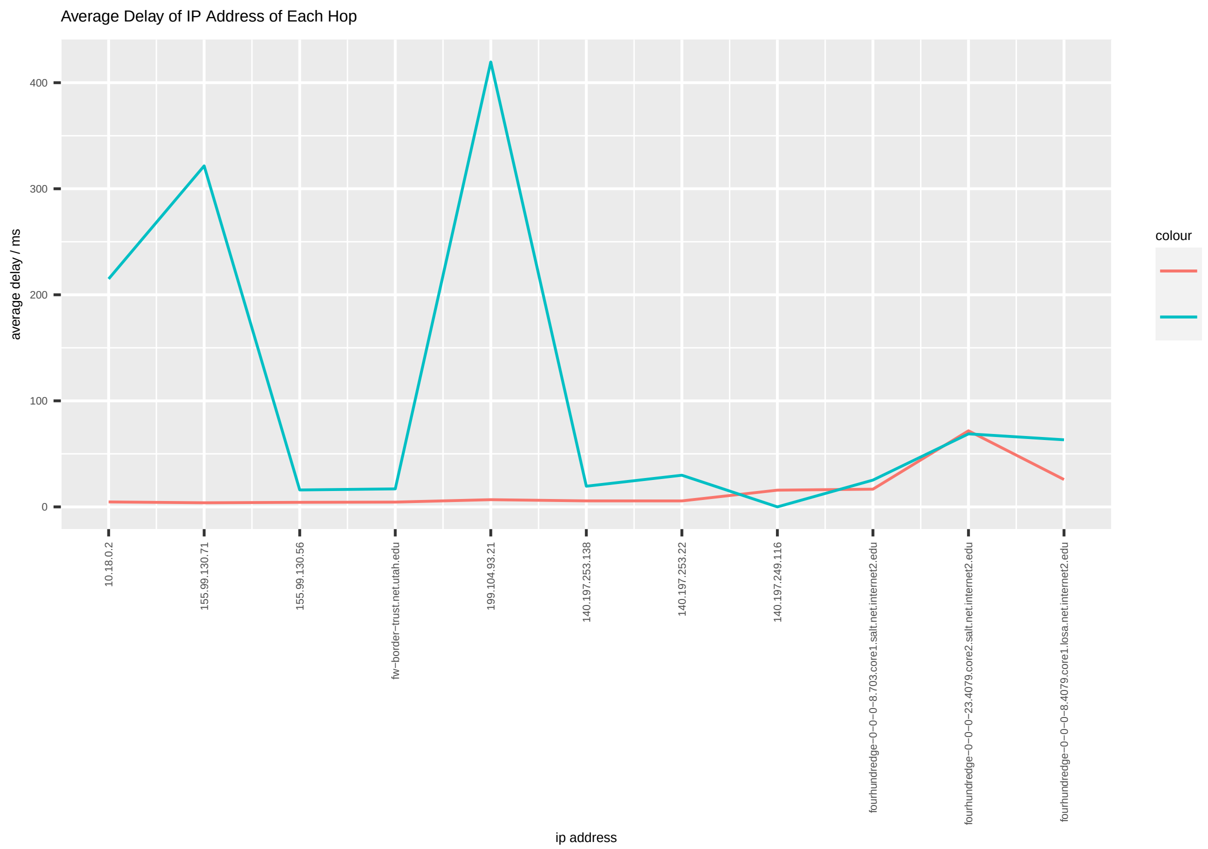
Use this new file to plot a graph with the x-axis showing the IP address of the hop and the y-axis the average delay corresponding to that hop. Run the same experiment a few hours later and show the new results on the same graph. Thus, your graph should have two curves. Submit the traceroute outputs, your code, and a pdf version of your graph on github.

Note, this is a good opportunity to try a new programming language that's suited to these types of short text processing programs. Python is a good choice. Also note, that it's difficult to get complete data. Justify whatever choice you make for how to deal with incomplete data.

```
options(warn=-1)

get_ips <- function(file_name) {
    file_str <- paste(readLines(file_name), collapse="\n")
```

```
    str_list <- as.list(strsplit(file_str, '\\s+')[[1]])
    i <- 1
    str_len <- length(str_list)
    len_range <- 1:str_len
    ip_vec <- c()

    for (j in len_range) {
        if (str_list[j] == as.character(i)) {
            i <- i + 1
            ip_vec <- append(ip_vec, str_list[j + 1])
        }
    }

    return(ip_vec)
}

parse_file <- function(file_name) {
    file_str <- paste(readLines(file_name), collapse="\n")
    str_list <- as.list(strsplit(file_str, '\\s+')[[1]])
    num_vec <- as.numeric(unlist(str_list))
    num_vec <- na.omit(num_vec)
    i <- 2.000
    avg_num <- 0.000
    avg_list <- c()
    cur_avg_list <- c()

    for (cur_element in num_vec) {
        if (cur_element == i) {
            i <- i + 1
            if (length(cur_avg_list) > 0) {
                avg_num <- mean(cur_avg_list)
            }
            else {
                avg_num <- 0
            }
            avg_list <- append(avg_list, avg_num)
            cur_avg_list <- c()
        }
        else if (cur_element != 1) {
            cur_avg_list <- append(cur_avg_list, cur_element)
        }
    }

    if (length(cur_avg_list) > 0) {
        avg_num <- mean(cur_avg_list)
    }
    else {
        avg_num <- 0
    }
    avg_list <- append(avg_list, avg_num)

    return(avg_list)
}
```

```r
library(ggplot2)
avg_list_1 <- parse_file("output1")
avg_list_2 <- parse_file("output2")

ip_list <- get_ips("output1")

len_1 <- length(avg_list_1)
len_2 <- length(avg_list_2)
lens <- c(len_1, len_2)
min_len <- min(lens)

hop <- c(1 : min_len)
avg_list_1 <- avg_list_1[1 : min_len]
avg_list_2 <- avg_list_2[1 : min_len]
ip_list <- ip_list[1 : min_len]

timingdata <- data.frame(hop, avg_list_1, avg_list_2)

write.csv(timingdata, file = "timing_data.csv")

ggplot() + geom_line(data = timingdata, aes(x = hop, y = avg_list_1,
    colour = "1")) + geom_point() +
    geom_line(data = timingdata, aes(x = hop, y = avg_list_2,
    colour = "2")) +
    scale_x_continuous(breaks = hop,
    labels = ip_list, guide = guide_axis(angle = 90)) +
    theme(text = element_text(size = 5)) + labs(
    x = "ip address",
    y = "average delay / ms",
    title = "Average Delay of IP Address of Each Hop")
```



b Suppose one of the three traceroute delay values between the source and a given router hop turns out to be unusually high. What are two possible causes for this unusually high delay?

- Network congestion.
- Slow path queuing delay at routers.

Question 3: Ping

```
parse_file <- function(file_name) {
    file_str <- paste(readLines(file_name), collapse="\n")
    str_list <- as.list(strsplit(file_str, '\\s+')[[1]])
    time_list <- c()

    for (cur_element in str_list) {
        if (startsWith(cur_element, "time=")) {
            cur_time <- substr(cur_element, 6, 12)
            time_list <- append(time_list, cur_time)
        }
    }

    return(time_list)
}
```

```r
time_list <- parse_file("ping")
num_vec <- as.numeric(unlist(time_list))
sum_time <- sum(num_vec)
last_delay <- tail(num_vec, n = 1)
len_vec <- length(num_vec)
avg_delay <- sum_time / len_vec - last_delay
cat("The average round trip queuing delay experienced by the packets is", avg_delay, "ms.")
```

```
## The average round trip queuing delay experienced by the packets is -1.797012 ms.
```