Feedback on Assignment I

ELEC-E7320 Internet Protocols, 2019

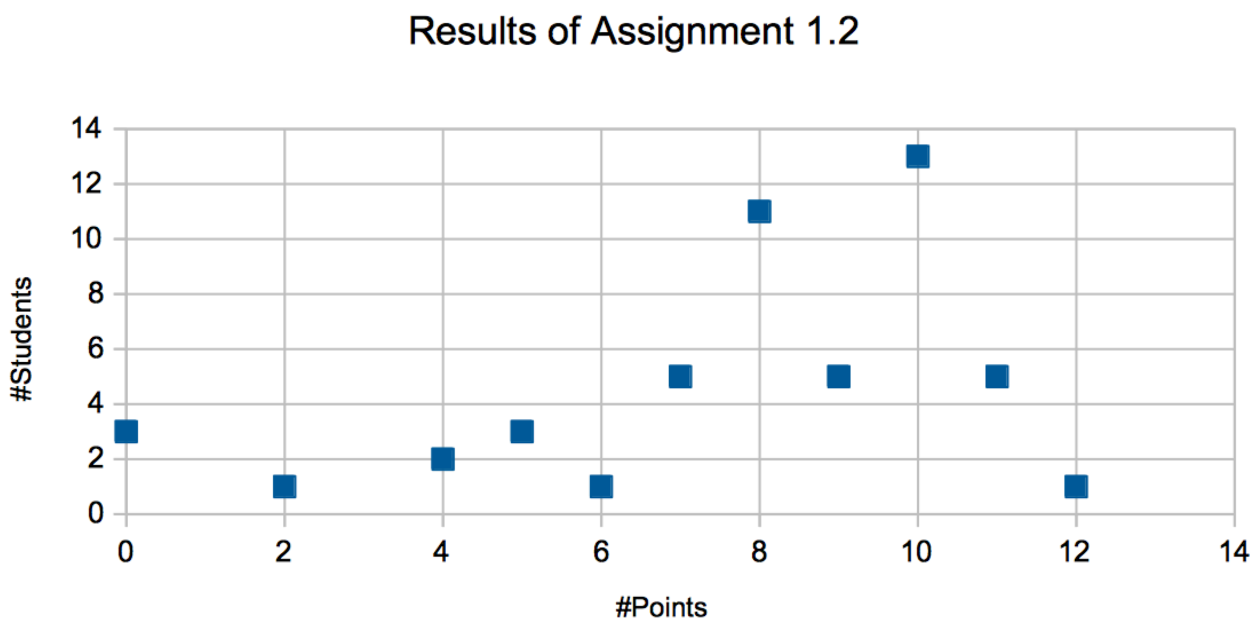
Aalto-yliopisto
Sähkötekniikan korkeakoulu

Yu Xiao

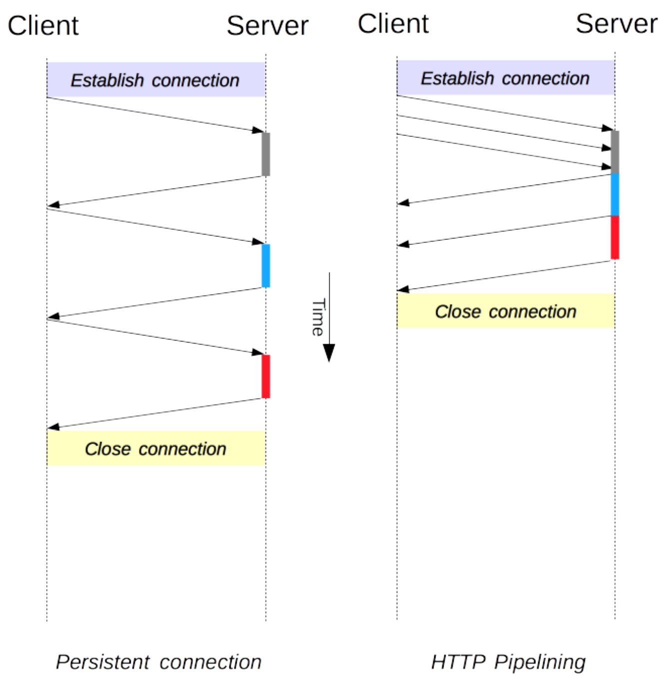
18.2.2019

Overview

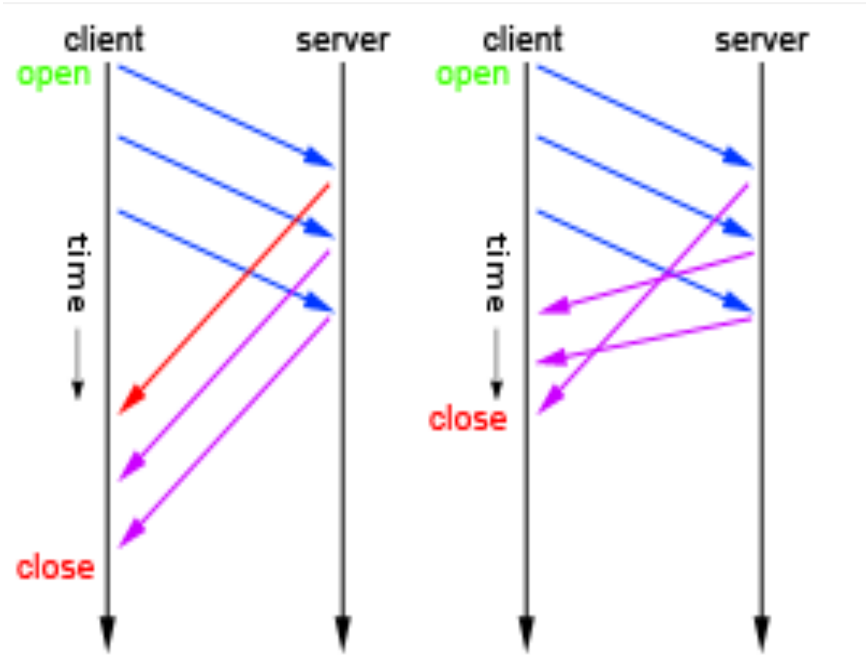
- **50 submissions, median: 9p**



Multiplexing



HTTP/1.1



HTTP/1.1 Pipelining vs. HTTP/2 multiplexing

Multiplexing

- HTTP/1.1: supports persistent TCP connection and pipelining. No need to establish one connection per request.
- HTTP/1.1: most browsers support concurrent TCP connections (typically max 6)
- One slow or large response would delay the progress in case of HTTP/1.1

Experiment Design

- **Not fair to assume one TCP connection per request (in sequence)**
- **Important details are often missing from reports:**
  - HTTP/1.1 pipelining enabled or not?
  - HTTP/1.1 N TCP connections?
  - When multiplexing was enabled, how many streams were in use?
  - What did the clients download from server?
  - Were other features like server push disabled?
  - Lossy networks? Variation in measurements?

Server Push

- HTTP/2 server push enabled vs. HTTP/2 server push disabled
- If you simply compare HTTP/2 and HTTP/1.1, you will be evaluating the joint effects of many features including server push, multiplexing, flow control and etc. How would you solve this problem through a smart test case? (e.g. limit #request, size of files to download, network link state)
- #requests you observe from Chrome dev tool vs. #requests observed from Wireshark

Flow Control

- **A flow-control scheme ensures that streams on the same connection do not destructively interfere with each other**.
- Flow control is used for both individual streams and for the connection as a whole.
- **Default HTTP/2 vs. HTTP/2 with flow control disabled**
- **How to disable HTTP/2 flow control?**

Experiment Design

“Deployments that do not require this capability can advertise a flow-control window of the maximum size (2^{31}-1) and can maintain this window by sending a WINDOW_UPDATE frame when any data is received. This effectively disables flow control for that receiver. Conversely, a sender is always subject to the flow-control window advertised by the receiver.”

RFC7540 Chapter 5.2.2 Appropriate Use of Flow Control

Experiment Design

- **Flow control is directional, and is hop-to-hop**

"HTTP/2 defines only the format and semantics of the WINDOW_UPDATE frame (Section 6.9). This document does not stipulate how a receiver decides when to send this frame or the value that it sends, nor does it specify how a sender chooses to send packets. Implementations are able to select any algorithm that suits their needs."

RFC7540 Chapter 5.2.1

- **#streams**
- **What kind of data to transmit in each stream?**
- **How to handle WINDOW_UPDATE frame in your implementation?**
- **Network conditions?**

Stream Priority

- How to disable ‘stream priority’?

- #stream
- What to download?
- Default HTTP/2 vs. HTTP/2 with stream priority disabled

Protocol Analysis

- **Use experimental results to explain how the new design change the way of communication or other factors in order to improve performance**

- **Common problems:**
  - No explanation about experimental results
  - X-axis and y-axis were not defined. Units were missing.
  - Lacking analysis of metrics like #requests, amount of data, order of data, shape of traffic