Tutorial on HTTP2 and Wireshark

Chao Zhu, chao.1.zhu@aalto.fi

This tutorial covers

- HTTP/2
  - A brief introduction on HTTP/2
  - How to implement your own HTTP/2 client & server
- Wireshark
  - How to use Wireshark to monitor network packets
  - How to deal with the encrypted packets

HTTP/2

- The next version of HTTP/1.x
- Documented in RFC 7540
- Features
  - Binary protocol
  - Multiplexing
  - Header compression
  - Server push
  - Request prioritization
- HTTP/2 outperforms HTTP/1.x in many aspects[1][2], read the papers for details
- Demo page

[1] HTTP/1.1 pipelining vs HTTP2 in-the-clear: Performance comparison, NOTERE ' *16*
[2] How Speedy is SPDY? NSDI ' 14

HTTP/2 Clients & Servers

- Java
  - Netty, Tomcat
- C/C++
  - Nginx, H2O, Nghttp2
- Python
  - ***Hyper-h2***
- More
  - https://github.com/http2/http2-spec/wiki/Implementations

Hyper-h2 Installation (For Linux)

- Install Python 3
  - sudo apt-get install python3.6
- Install Hyper-h2
  - pip install h2
- Install Hyper
  - pip install hyper

A Simple Hyper-h2 Server

- A simple server implementation
- Demo
  - Check HTTP/2 request ‘special headers’

HTTP/2 in Chrome

- By default, Chrome enables HTTP/2
- Openssl and Libapr required
  - For macOS, *brew install apr-util openssl*
  - For Ubuntu, *apt-get install libssl-dev libapr1-dev*
- Demo
  - Clone code from Github
  - Visit server from chrome
- Developer panel

Implement Client & Server

- Notes:
- Free to use any language (C++, Java and Python)
- Hyper-h2 does not provide I/O
- You need to implement your own client and server (not just browser)
- Just focus on file sending, receiving and packet decrypt (we do not care about GUI and rendering)

Wireshark

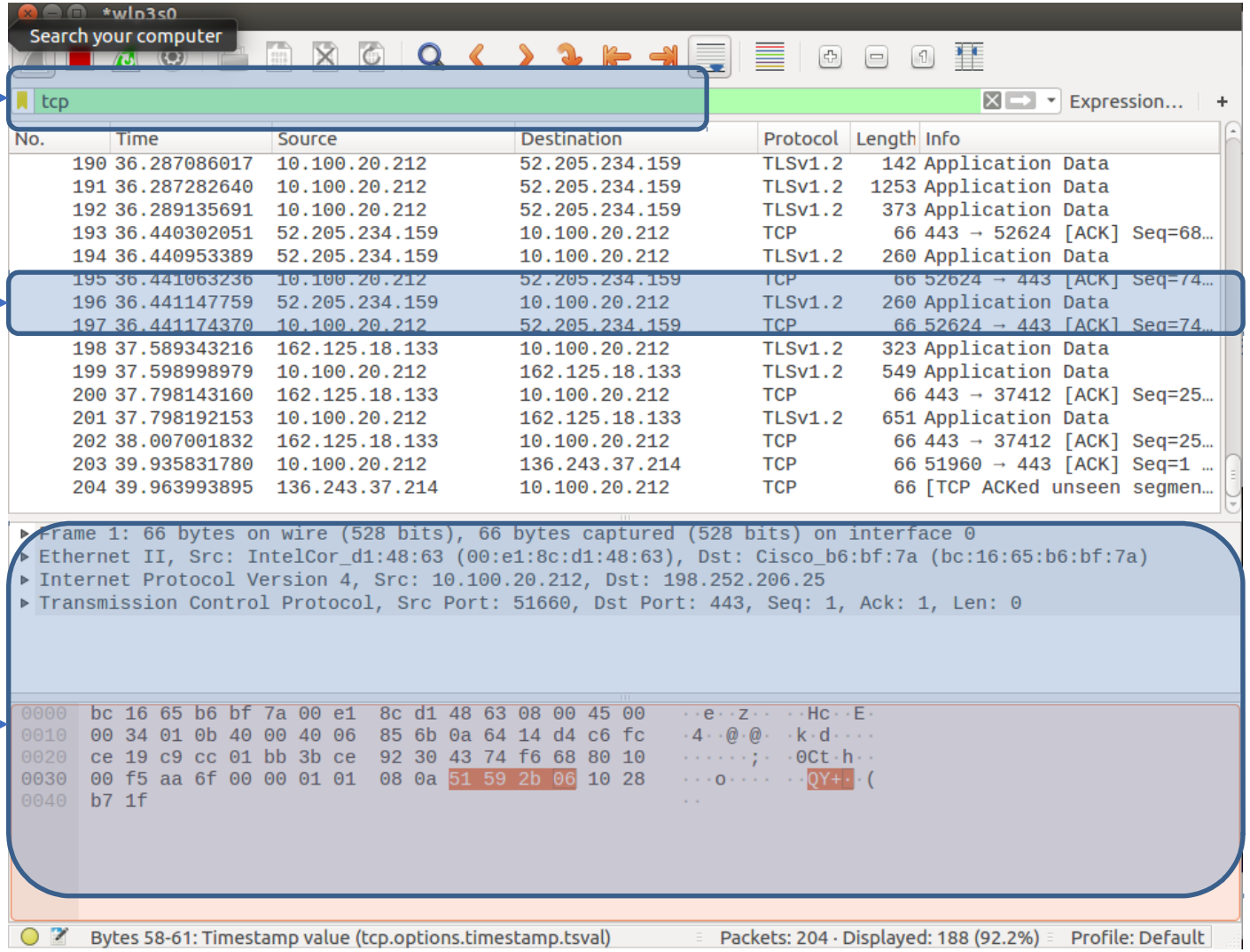
- A packet analyzer that captures data packets flowing over the network
- Installation
  - Downloads

Wireshark

Filter

Packets

Packet Detail



Wireshark Demo

- Wireshark Tutorial for Beginners, https://www.youtube.com/watch?v=TkCSr30UojM

Hints on Assignments

- Assignment 1, implement server and client
  - Read the protocol specification
  - Design test cases for evaluating the protocol design
    - Access a webpage
    - Post an image
    - Post tags (both ‘POST’ and ‘PUT’ are okay)
    - Receive notification from server

Hints on Assignments

- Assignment 2, test HTTP/2 new features
  - Select 2 out of the 4 new features
    - Flow control
      - local_flow_control_window(***stream_id***)
      - increment_flow_control_window(***increment***, ***stream_id=None***)
    - Stream prority
      - prioritize(self, stream_id, weight=None, depends_on=None, exclusive=None)
    - Multiplexing
      - Sending multiple requests over a communications link without response
    - Server push
      - Server pushes a resource directly to the client without the client asking for the resource.

Other Hints

- HTTPS packets decryption with Wireshark
- Netty-based Implementations

Reference

M. Jiang, X. Luo, T. Miu, S. Hu and W. Rao, "Are HTTP/2 Servers Ready Yet?," 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, 2017, pp. 1661-1671. doi: 10.1109/ICDCS.2017.279