Q: Where can I find open source implementations of HTTP/2 client and server?

You can find a list from https://github.com/http2/http2-spec/wiki/Implementations

Q: Do you provide tutorials on Java-based implementations?

This year we only provide tutorial on Python-based implementation. However, our course assistants developed examples of HTTP/2 client/server based on Netty in previous year. If you would like to take it as reference, you can download the code from https://github.com/johnson-li/H2-TA.git.

- Run the server
  - Right click on Http2Server.java, click **Run Http2Server.main()**
  - If you use the Jetty-ALPN
    - Click on **Edit Configurations...**
    - Fill **VM options** with: **-Xbootclasspath/p:/path/to/jetty/alpn/alpn-boot.jar**
  - Open Chrome, visit https://127.0.0.1:8443/
- Run the client
  - Follow the same procedure to run the client, use **Http2Client.java** instead
- Parameter class: example.Parameters
- Static parameter variables
  - SSL, whether to use HTTPS or not
  - PORT, server port
  - SERVER_HOST, server host name or IP
  - SERVER_URL,
  - CLIENT_ENABLE_HTTP2, whether to use HTTP/2 in the client side
  - SERVER_ENABLE_HTTP2, whether to use HTTP/2 in the server side

Detailed instruction of Netty setup can be found from https://netty.io/wiki/requirements-for-4.x.html

- Compile Netty by your self
  - Git clone https://github.com/netty/netty.git
  - Git checkout netty-4.1.17.Final
  - mvn install -Dmaven.test.skip -Denv=release
  - //Change some code
  - mvn install -Dmaven.test.skip -Denv=release
  - Compile your HTTP/2 client and server, the change should take effect

Important classes & methods that may be related to new features in HTTP/2.

io.netty.handler.codec.http2.DefaultHttp2FrameReader
- readHeadersFrame: Parse the HEADERS frame (https://tools.ietf.org/html/rfc7540#section-6.2), stream priority is defined here.

- readWindowUpdateFrame: Parse the WINDOW_UPDATE frame (https://tools.ietf.org/html/rfc7540#section-6.9), it controls the behaviro of flow control.

io.netty.handler.codec.http2.DefaultHttp2FrameWriter
- writePushPromise: Generate the PUSH_PROMISE frame (https://tools.ietf.org/html/rfc7540#section-6.6), it is used to init a server push.
- writeData: Generate the DATA frame (https://tools.ietf.org/html/rfc7540#section-6.1), and the resource content is wrapped in the DATA frame.

io.netty.handler.codec.http2.WeightedFairQueueByteDistributor: This class schedules the concurrent stream and implements a fair sharing policy.