

* High Level Design

The project consists two parts: client and server.

The client and server are able to support the HTTP GET request under two different options of connection: HTTP 1.0 and HTTP 1.1.

- client

The client will accept input of multiple URLs. For each URL, the client will transform the hostname to IP address by DNS lookup, and then setup connection to the server, send requests, and receive requested files. Under the HTTP 1.0 protocol, the client will setup a new connection to the server for each request. Under the HTTP 1.1 option, if the next URL (with port number) remains unchanged, the client will keep the previous connection. If the URL and port number changed, the client will close the the previous connection and setup a new one.

- server

The server accepts the requests from the client. It will accumulate received data until finding "\r\n\r\n", meaning it receives a full header. Server then checks if the request is valid, and build the corresponding status line: 400 bad request, 404 Not found, and 200 OK. If the request is valid, the server will build the header, attach the data, and send it back to the client. Under the HTTP 1.0 option, the server will end the current connection after sending the message; under the HTTP 1.1 option, the server will set the timer and wait for the incoming data until timeout.

* Problem

We spent time to learn the new API.

* Test

We tested our code under two circumstances: first, we have one client send request to download file from server at local host; second, we have multiple clients to download files simultaneously. Both cases worked.

* Contribution

Client and httpmessage header file: Yunong Jiang [204466452]
Server: Hanjun Li [504425747] Tian Ye [504270284]

- * extra credit
 - timeout
 - client timeout for connection without response.
 - server timeout for incomplete header message.
 - HTTP/1.1
 - client keep previous connection if current URL and port number stays the same.
 - server waits on a connection to see if there are follow up message from client.