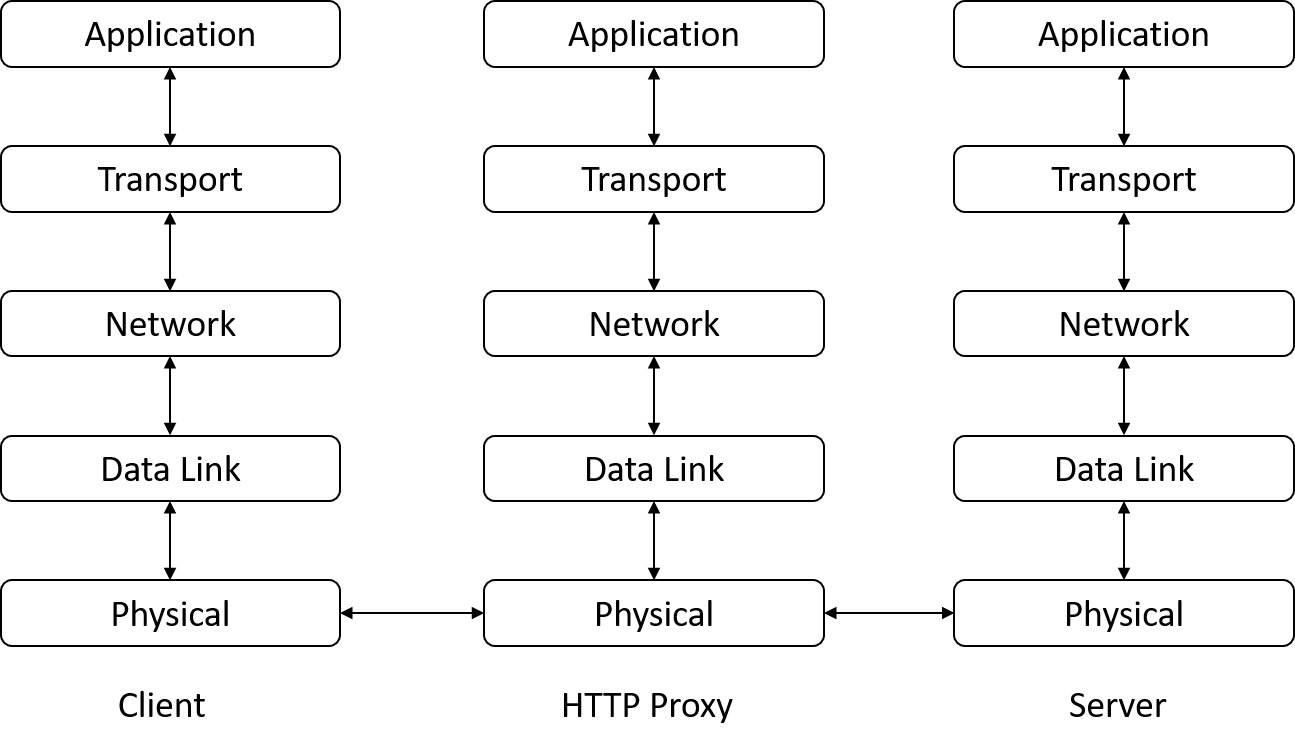
**Project2: Proxy Server and Network Address Translation (NAT)**

M2608.001200 Introduction to Data Communication Networks (2021 Fall)

Name: Student ID:

1. Explain how packets are exchanged between a client to a server with the layering architecture shown below when using HTTP proxy and NAT (DNAT+SNAT configuration), respectively. (Draw arrows for request and response packets on the figures below to describe the path of packets and use these figures for your explanation). (30pts)



1. List the pros and cons of HTTP proxy and NAT proxy and compare them. (20pts)
2. Describe how you implemented HTTP proxy server and how the addresses are changed by iptables rules with a screenshot of *result.pcap* opened by Wireshark. (30pts)
3. One of the applications of the HTTP proxy is web page caching. The cache proxy saves the response file when the file is requested for the first time. Then, when the file is requested again, the proxy sends the stored file to the client without requesting it again from the server. By doing so, clients can receive the file faster. However, files in the server can be updated such that the files in the proxy are not up-to-date. To solve this problem, we can use HTTP conditional requests called ‘if-modified since’. Based on the concept above, provide your pseudocode for implementing HTTP cache proxy which can always send up-to-date files to the client. (20pts)  
     
   Skeleton for Pseudocode

Struct HTTPRequest {

string method; /\* e.g, GET \*/

string path; /\* e.g, /index.html \*/

string version; /\* e.g, HTTP/1.1 \*/

Dict<String, String> headers; /\* e.g, Content-Type: text/html \*/

/\* use : req->headers[“Contet-Type”]\*/

}

Struct HTTPResponse {

string version; /\* e.g, HTTP/1.1 \*/

int result\_code; /\* e.g, 200 \*/

string result\_message; /\* e.g, OK \*/

}

main() {

open\_socket\_and\_listen();

while True {

sock\_client = accept\_from\_client();

HTTPRequest req = parse\_http\_req(recv(sock\_client));

forward(req, sock\_client);

}

}

forward(struct HTTPRequest req, int sock\_client) {

sock\_server = open\_socket\_to\_server();

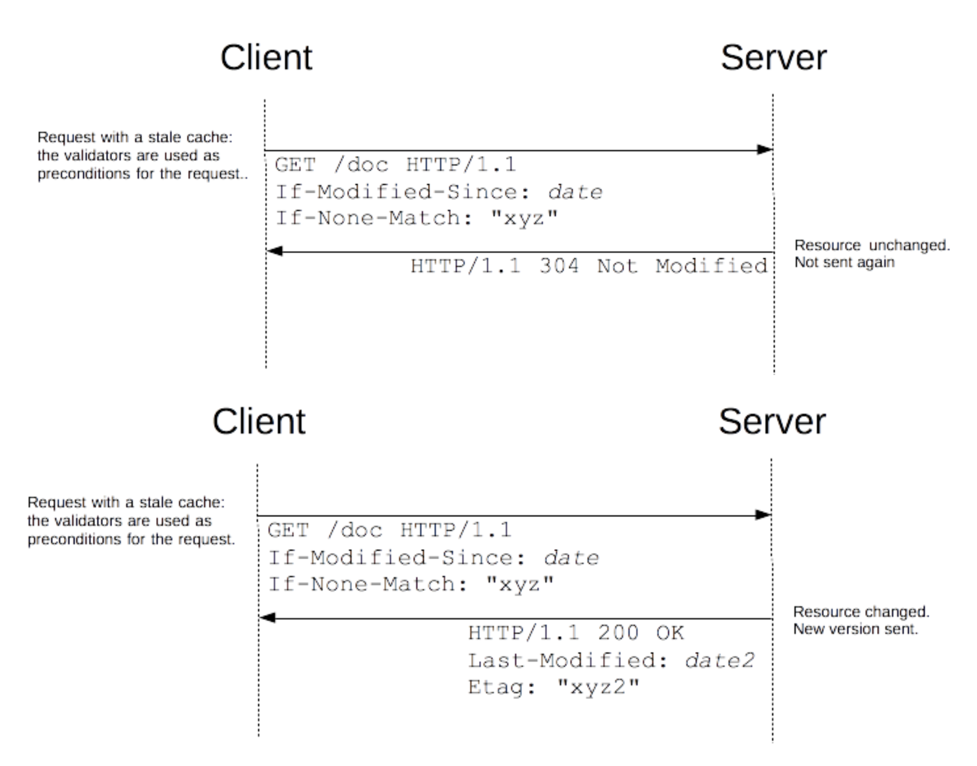
(file\_exist, cache\_file, last\_update\_time) = open\_file(req.path);

**/\* TODO : write your pseudo code for HTTP cache proxy \*/**

close(sock\_server)

}

Reference for conditional request: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Conditional_requests>



<Example of using If-Modified-Since Request>