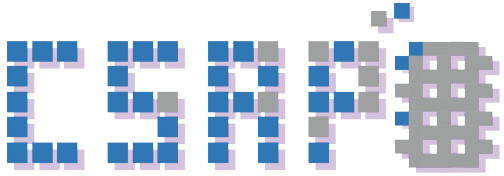
System Programming
Lab Session #6

Proxy Lab

2019/11/19

sysprog@csap.snu.ac.kr



Computer Systems and Platforms Laboratory
School of Computer Science and Engineering
Seoul National University

Content

- **Implementing an HTTP server**
  - HTTP server concept
  - HTTP server set up
  - HTTP server implementation guide
  - Tips

- **Implementing an proxy server**
  - Proxy server concept
  - Proxy server setup
  - Proxy server implementation guide
  - Tips

- **Caching and Logging**
  - Proxy cache concept
  - Proxy cache setup
  - Proxy server implementation guide
  - Proxy log implementation guide
  - Tips

Content

- **Implementing an HTTP server**
  - HTTP server concept
  - HTTP server set up
  - HTTP server implementation guide
  - Tips

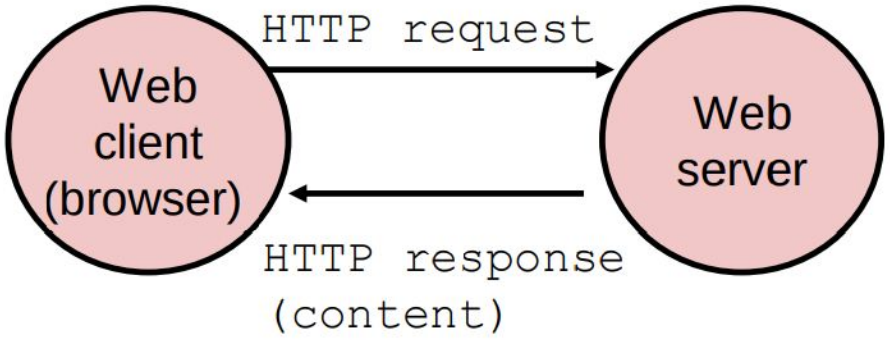
- **Implementing an proxy server**
  - Proxy server concept
  - Proxy server setup
  - Proxy server implementation guide
  - Tips

- **Caching and Logging**
  - Proxy cache concept
  - Proxy cache setup
  - Proxy server implementation guide
  - Proxy log implementation guide
  - Tips

HTTP servers concept

- **Clients and servers communicate using the HyperText Transfer Protocol (HTTP)**
  - Client and server establish TCP connection
  - Client requests content
  - Server responds with requested content
  - Client and server close connection (eventually)
  - HTTP 1.0 Specification: RFC 1945, May 1996

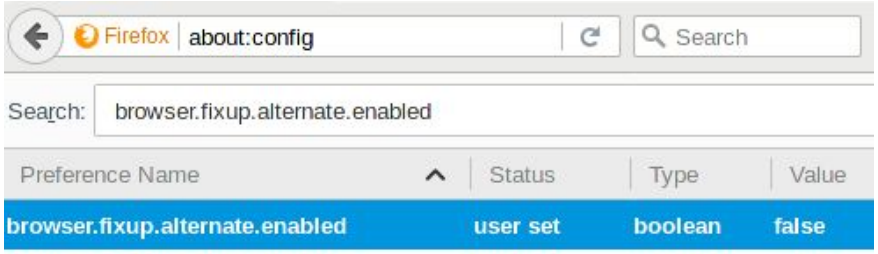
- **Later Versions of HTTP in use are 1.1 and 2.0**



HTTP servers setup

- Please only use the Firefox browser for this lab session

- To stop your browser from auto-correcting your requests if it doesn't get a response type: 'about:config' into your address bar.
- Press the ok button then search for the following two flags and set them to **false**
  - browser.fixup.alternate.enabled
  - network.captive-portal-service.enabled



HTTP server implementation guide 1

- **Your first task is to:**
  - Accept incoming connections from the browser
  - Pass this to the ‘doit’ function to handle it

- **Secondly parse the HTTP request in the format:**

  method URI version

  Host: <host>:<port>

- **Based on this information implement 3 types of HTTP error:**
  - 501 - method not implemented
  - 404 - file not found
  - 403 - permission denied (for directories)

- You should use the client_error function to handle this
- You can use the stat function to find info on the file

HTTP server implementation guide 2

- **Implement the serve static function**
  - This should send the file back to the browser as a HTTP response

version status-code status-message

Server: <serverName>

Content-type: <contentType>

Content-Length: <contentLength>

<content>

- **Add image files (png, jpg, gif) to the servable files**

- **Test your solution with files in the ./pages and <u>your own</u> testing files**

Tips

- **sscanf**
  - sscanf(sourceString, “Name: %s Age: %d”, name, &age);
  - Will read name=’TA’ and age=20 from the following string:
  - “Name: TA Age: 20”
  - You can use the string ‘%[^:]’ to read a string up until the : character

- **sprintf**
  - name=”TA”
  - intro=”My name is:”
  - sprintf(name, "%s %s ", intro, name);
  - Will write the string: ‘My name is: TA’ to the name string.

- **Newlines**
  - HTTP requests require that the characters **‘\r\n’** should represent a newline, just the character ‘\n’ is not sufficient

Content

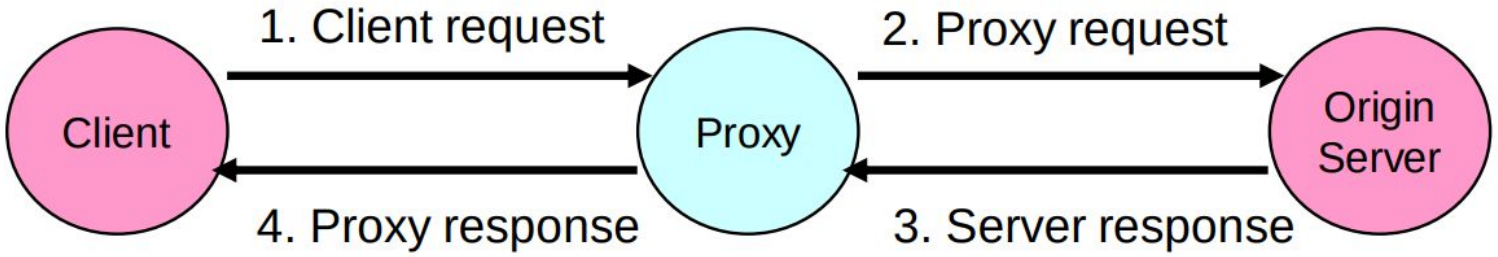
- Implementing an HTTP server
  - HTTP server concept
  - HTTP server set up
  - HTTP server implementation guide
  - Tips

- **Implementing an proxy server**
  - Proxy server concept
  - Proxy server setup
  - Proxy server implementation guide
  - Tips

- Caching and Logging
  - Proxy cache concept
  - Proxy cache setup
  - Proxy server implementation guide
  - Proxy log implementation guide
  - Tips

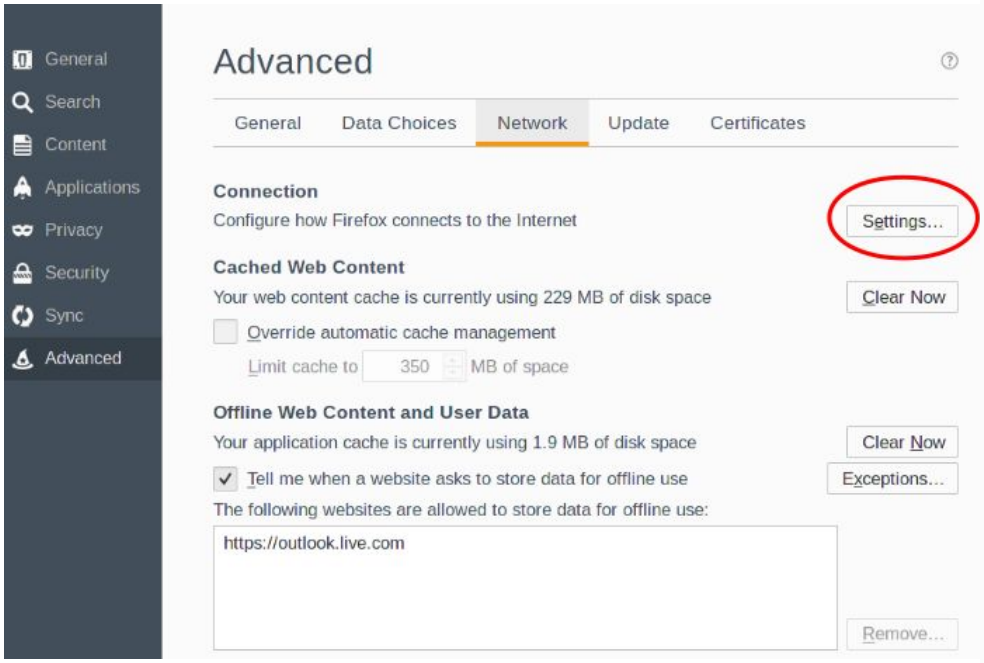
Proxy servers concept

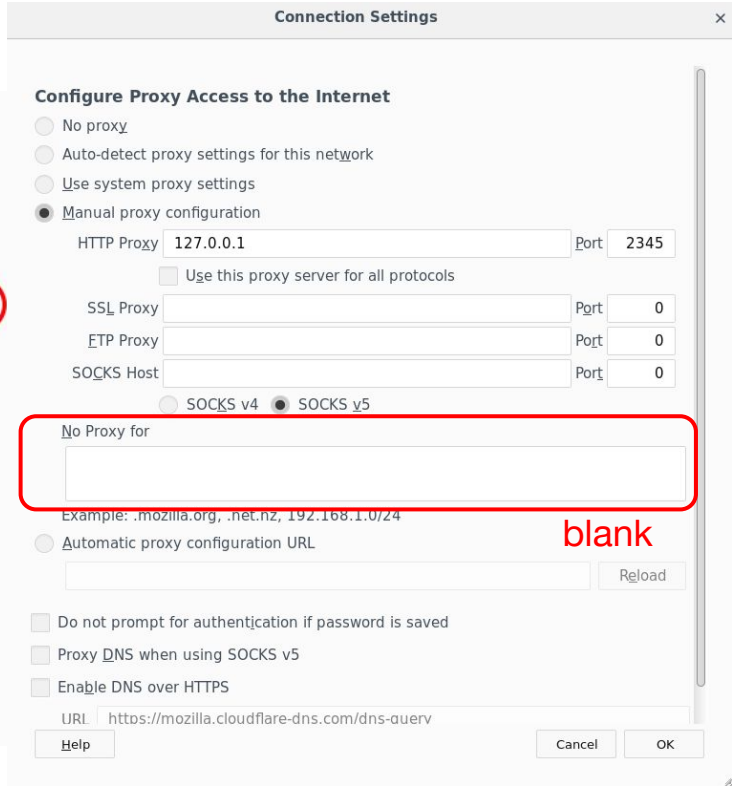
- A proxy is an intermediary between a client and an origin server
- To the client, the proxy acts like a server
- To the server, the proxy acts like a client
- Usages: hide client information, caching, etc



Proxy servers setup

- Go to the preferences -> advanced -> network -> settings and configure as shown in the second picture
- This tells your browser to send all requests to your proxy server
- Don’t use proxy server for SSL (HTTPS)





Proxy implementation guide 1

- **step 1: Implement accepting connections from the browser (the same as the HTTP server)**
- **step 2: Parse URI to extract the host and port information**
- step 3: Send the request on to the end server
- step 4: Receive response from end server
- step 5: Send this back to the client

Proxy implementation guide 2

- **URI Parsing**
  - Get host name, port
  - Set default port number as ‘80’ **(We are not consider HTTPS)**
  - HTTP URI parsing **(Don’t too focus on corner case)**

Example1: http:// www.snu.ac.kr /index.html

Host: **www.snu.ac.kr** Port: **80**

Example2: http:// 127.0.0.1 : 1234 /index.html

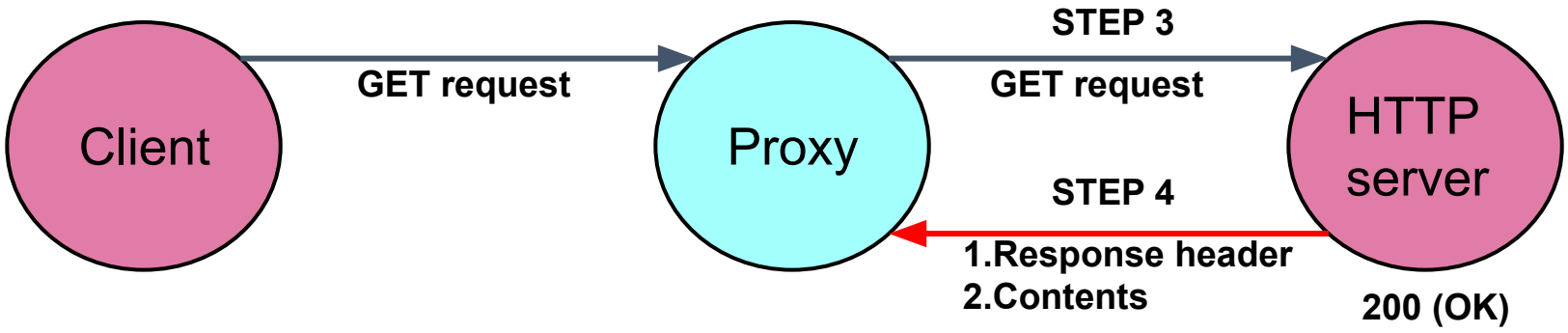
Host: **127.0.0.1** Port: **1234**

Proxy implementation guide 3

- step 1: Implement accepting connections from the browser (the same as the HTTP server)
- step 2: Parse the host and port information from the request
- **step 3: Send the request on to the end server**
- **step 4: Receive response from end server**
- **step 5: Send this back to the client**

Proxy implementation guide 4

- **Response header**



- **Example**

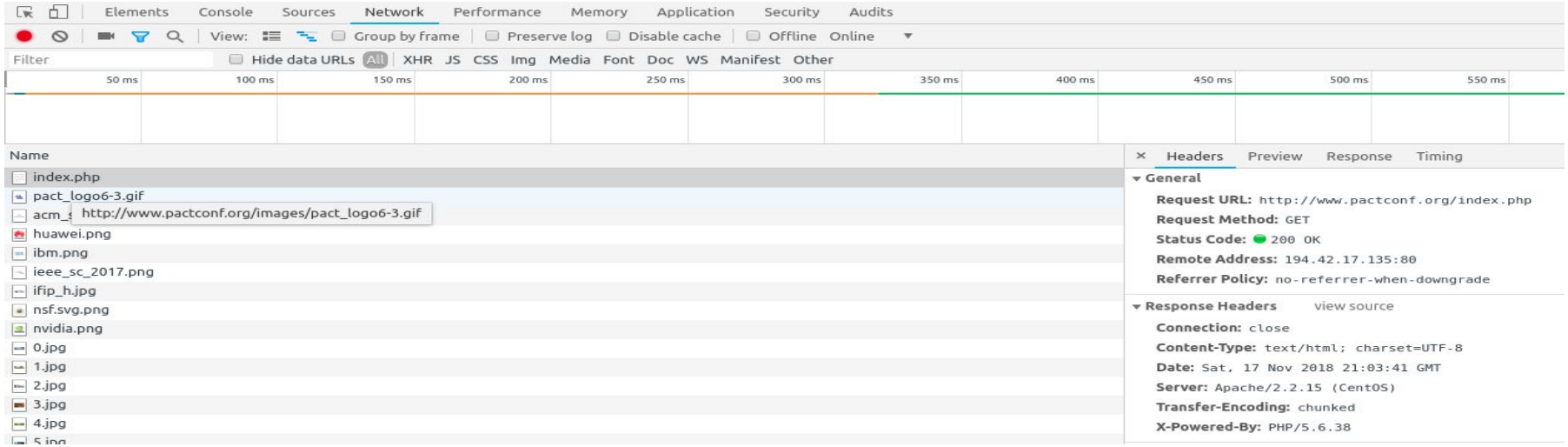
HTTP/1.1 200 OK
Date: Sat, 17 Nov 2018 20:08:54 GMT
Server: Apache/2.2.14 (Ubuntu)
Content-Type: text/html
Content-Length: 2940
Connection: keep-alive
....

Tips

- **You can use the Content-Length in the header to know how large the file you are returning is.**

- **Test site**
  - **Content-Length:** *http://www.columbia.edu/~fdc/sample.html*

- **Check header**
  - **chrome: <F12> / Firefox: Ctrl+Shift+Q /Or use wireshark**



Content

- **Implementing an HTTP server**
  - HTTP server concept
  - HTTP server set up
  - HTTP server implementation guide
  - Tips

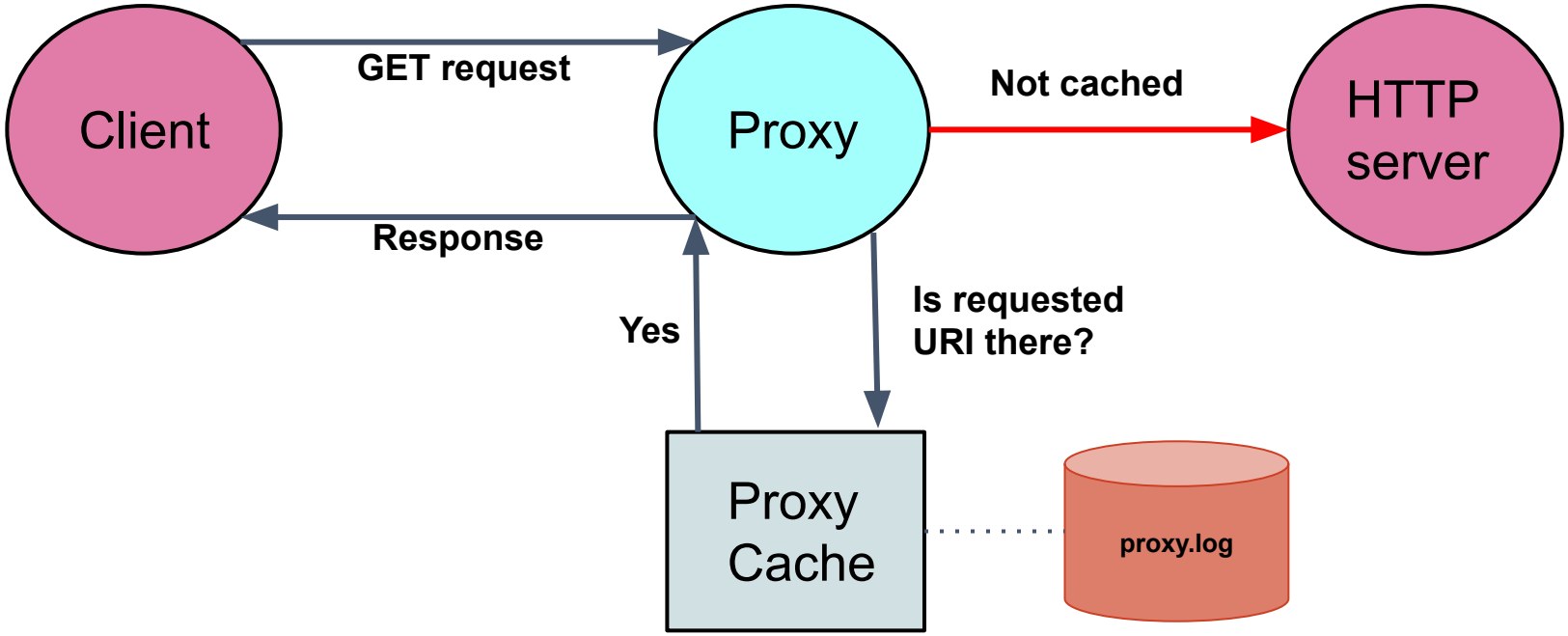
- **Implementing an proxy server**
  - Proxy server concept
  - Proxy server setup
  - Proxy server implementation guide
  - Tips

- **Caching and Logging**
  - Proxy cache concept
  - Proxy cache setup
  - Proxy server implementation guide
  - Proxy log implementation guide
  - Tips

Proxy cache concept

- **Proxy caching allows a server to act as an intermediary between a user and a provider of web content.**

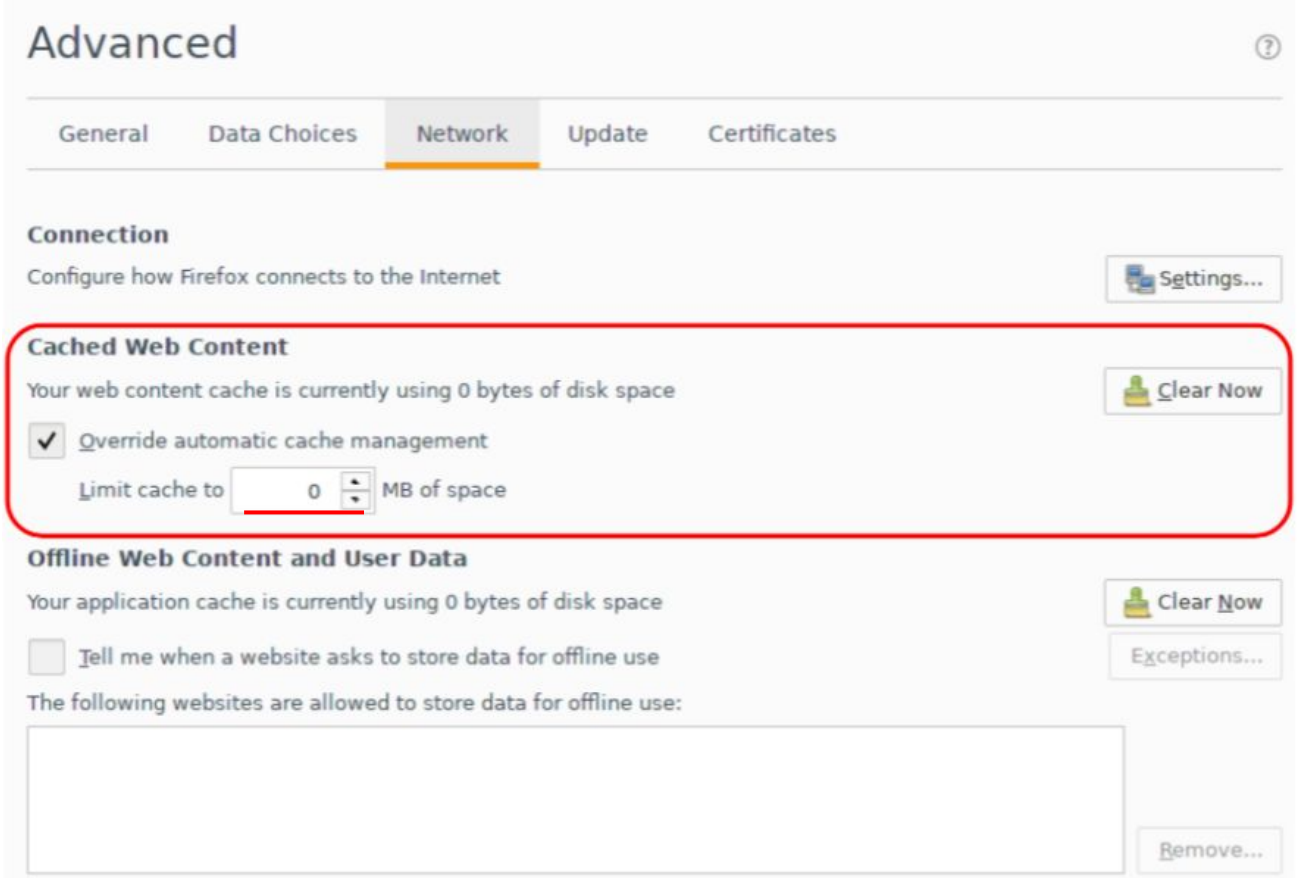
- **When a user accesses a website, proxy server interpret and respond to request on behalf of the original server.**



Proxy cache setup

- Go to the preferences -> advanced -> network -> settings

- Check the override automatic cache management and set the limit cache to 0MB



Proxy cache implementation guide

- **Step**
  - Search URI in the proxy cache.
  - Hit: Get response header and content, and send to the client
  - Miss: Send request to the end server. Get response. Add response header and contents in proxy cache. Check the current size of cache and object get from end host

- **Data structure**
  - It is okay to use **any C libray** to support data structure.
  - You can use any cache replacement policies.

- **Size of cache**
  - Total cache size: **1MB**
  - Cache block for one content: **200kB**

Proxy log implementation guide

- **Proxy server record when it is newly cached or cached contents are sent to the client.**

- **The information of the proxy log should include:**

```
cacheStatus date requestedURL contentLength
```

- **This is one of the example for the cached/uncached URI**

```
[uncached] Thu 14 Nov 2019 14:35:43 KST: http://localhost:1234/index.html 35
[cached] Thu 14 Nov 2019 14:35:45 KST: http://localhost:1234/index.html 35
```

You should match the format! otherwise the score can be penalized!

Tips

- You can test your own http server you made in part1.
- I recommend to use URI as a key value.
- Structures for cache blocks, variables should include at least ‘URI’, ‘contents’, ‘response header’.
- Watch out memory allocation, especially ‘free()’

Evaluation

- (15pts) HTTP server implementation
- (20pts) Proxy server implementation
- (30pts) Proxy cache implementation
- (10pts) Proxy log implementation
- (5pts) Code style
- (20pts) Reports

Deadline: Thur Dec 3th

Thank you

Questions?