ANNA UNIVERSITY, GUINDY, CHENNAI:: 600 025

DEPARTMENT OF COMPUTER SCIENCE AND

ENGINEERING

Course Code: CS6111

Course Name: Computer Networks

Semester : V Batch : N&Q

LAB 7: Implement Web Cache

Consider the following scenario, in which Web browsers connect to a web server . There is also a local web cache in the browsers' access network. In this question, we will ignore browser caching .

Assume that the total Round Trip Time propagation, queueing and object transmission delay between the browser and web server (and including TCP setup time) is 250 msec; if the object is retrieved from the local web cache, this delay is only 10 msec. origin servers 100 Mbps access link client local web cache Suppose a browser 1 makes a request for an HTTP object, which is not found in the local cache. A very short time later another browser 2(on the right) makes a request for the same HTTP object. What is the time from when the browser1 issues the HTTP GET request until receives an HTTP reply containing the object?

Your answer should be among the following list(250 msec, 200 msec, 20 msec, 10 msec, 100 msec, 550 msec, 300 sec, 350 msec).

LAB 7 SPOT PROGRAM :

Implement Web Cache in the scenario where the content was modified on the original server, rendering the copy on the proxy server to be an outdated one.

Reference:

HTTP request directives

The following directives can be submitted as part of a client's HTTP request.

- `max-age`
- Indicates the number of seconds before the client considers the HTTP response stale. If the *max-stale* directive is not included then HTTP responses older than this will not be returned.
- `max-stale`
- If no argument is supplied then the client will accept a stale HTTP response of any age. However, if *max-stale* is assigned a value then it will be the maximum number of seconds that a HTTP response can be stale for, and have the client still accept it.

- `min-fresh`
- Indicates that the client will only accept a HTTP response that will remain fresh for at least the specified number of seconds.
- `no-cache`
- Requires that the HTTP response cannot be retrieved from a cache unless it is first validated by the origin server.
- `no-store`
- Directs that no part of the HTTP request or HTTP response be stored in a cache. This applies to caches of any type, including private caches. In cases where the information is stored unintentionally in volatile storage, well-behaved systems will attempt to remove it as soon as possible.

Note: if a HTTP request containing the *no-store* directive was retrieved from a cache, then the directive does not apply to the existing stored HTTP response.

- `no-transform`
- Directs that intermediaries cannot transform the message body, regardless of whether they support caching.
- `only-if-cached`
- This indicates that the client only wants to receive a stored HTTP response. A system that receives this directive will return a stored version of the HTTP response that meets all of the requirements specified by the HTTP request. If no such copy exists then it will respond with a 504 Gateway Timeout status code.

HTTP response directives

The following directives are sent as part of the HTTP response from the server and direct the client on the requirements for caching the accompanying resources.

- `must-revalidate`
- This instructs the client that once a HTTP response is stale, the client cannot rely on the cached version before it is revalidated. With this directive present, if a client is unable to reach the origin server to revalidate the cache then a 504 Gateway Timeout status code will accompany the HTTP response.
- `no-cache`
- Directs client that new HTTP requests cannot rely on a cached version of the resource unless it is first validated, even if stale HTTP responses are allowed.
- `no-store`
- Directs client and intermediaries not to intentionally store any part of the HTTP response in a cache. This differs from the *no-cache* directive because HTTP response caching is allowed, but cannot be used unless first validated. The *no-store* directive implies that a

definite effort is made to eliminate all traces of this HTTP response from non-volatile storage as soon as possible.

- `no-transform`
- This instructs intermediaries not to transform the message body, regardless of whether a cache is involved.
- `public`
- Informs clients and intermediaries that the HTTP response may be cached, even in cases where it is normally non-cacheable or is restricted to a private cache.
- `private`
- This directive implies that this HTTP response cannot be stored in a shared cache. Furthermore, storing the HTTP response in a private cache is allowed, even in cases where the HTTP response is not normally cacheable.
- `proxy-revalidate`
- The directive is the same as *must-revalidate* but it does not apply to HTTP responses stored in a private cache.
- `max-age`
- Indicates the number of seconds before the client considers the HTTP response stale.
- `s-maxage`
- This directive is the same as *max-age* although it only refers to HTTP responses that are stored in a shared cache.