【CN】Day15(2)

Created	@June 6, 2022 1:41 PM
Class	
Type	
Materials	Web Caching
Reviewed	☐

【Ch2】Application Layer

2.2.5 Web Caching

A Web cache-also called a proxy server-is a network entity that satisfies HTTP requests on the behalf of an origin Web server.

The Web cache has its own disk storage and keeps copies of recently requested objects in this storage. A user's browser can be configured so that all of the user's HTTP requests are first directed to the Web cache.

As an example, suppose a browser ire requesting the object `http://www.someschool.edu/campus.gif` `.` Here is what happens:

1. The browser establishes a TCP connection to the Web cache and sends an HTTP request for the object to the Web cache.
2. The Web cache checks to see if it has a copy of the object stored locally. If it does, the Web cache returns the object within an HTTP response message to the client browser.
3. If the Web cache does not have the object, the Web cache opens a TCP connection to the origin server. Then it sends an HTTP request for the object into the cache-to-server TCP connection. After receiving this request, the origin server sends the object within an HTTP response to the Web cache.
4. When the Web cache receives the object, it stores a copy in its local storage and sends a copy, within an HTTP response message, to the client browser.

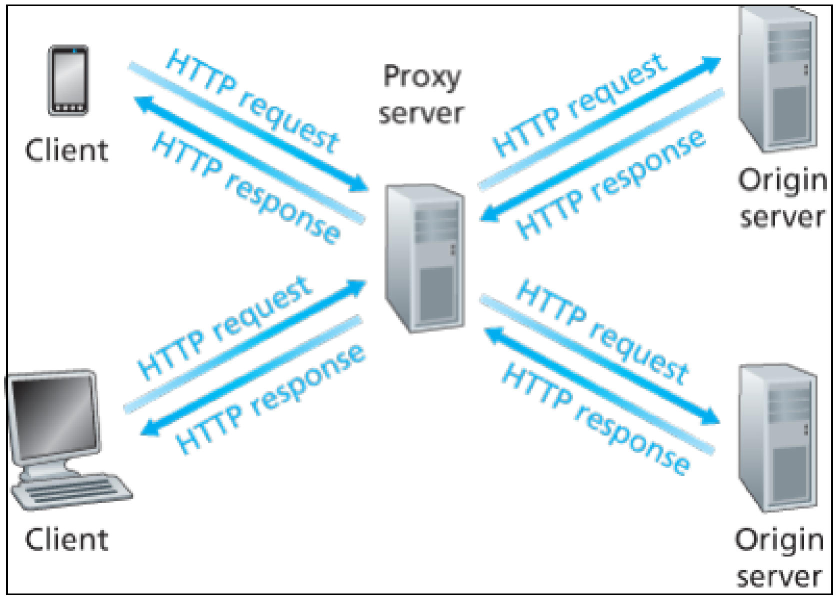

Figure 2.11 Clients requesting objects through a Web cache

Note that a cache is both a server and a client at the same time. When it receives requests from and sends responses to a browser, it is a server. When it sends requests to and receives responses from an origin server, it is a client.

Typically a Web cache is purchased and installed by an ISP. For example, a university might install a cache on its campus network and configure all of the campus browsers to point to the cache.

The Conditional GET

Although caching can reduce user-perceived response times, it introduces a new problem-the copy of an object residing in the cache may be stale. In other words, the object housed in the Web server may have been modified since the copy was cached at the client.

Fortunately, HTTP has a mechanism called the conditional GET that allows a cache to verify that its objects are up to date.

An HTTP request message is a so-called condition GET message if

1. the request message uses the `GET` method
2. and the request message includes an `If-Modified-Since:` header line.

Let’s walk through an example. First, on the behalf of a requesting browser, a proxy cache sends a request message to a Web server:

```
GET /fruit/kiwi.gif HTTP/1.1
Host: www.exotiquecuisine.com
```

Second, the Web browser sends a response message with the requested object:

```
HTTP/1.1 200 OK
Date: Sat, 3 Oct 2015 15:39:29
Server: Apache/1.3.0 (Unix)
Last-Modified: Wed, 9 Sep 2015 09:23:24
Content-Type: image/gif

(data data data data data ...)
```

The cache forwards the object to the requesting browser but also caches the object locally. The cache also stores the last-modified date along with the object.

One week later, another browser requests the same object via the cache, and the object is still in the cache. Since this object may have been modified, the cache performs an up-to-date check by issuing a conditional GET. Specifically the cache sends:

```
GET /fruit/kiwi.gif HTTP/1.1
Host: www.exotiquecuisine.com
If-modified-since: Wed, 9 Sep 2015 09:23:24
```

The conditional GET is telling the server to send the object only if the object has been modified since the specified date.

Then, the Web server sends a response message to the cache:

```
HTTP/1.1 304 Not Modified
Date: Sat, 10 Oct 2015 15:39:29
Server: Apache/1.3.0 (Unix)

(empty entity body)
```