

# Class Notes

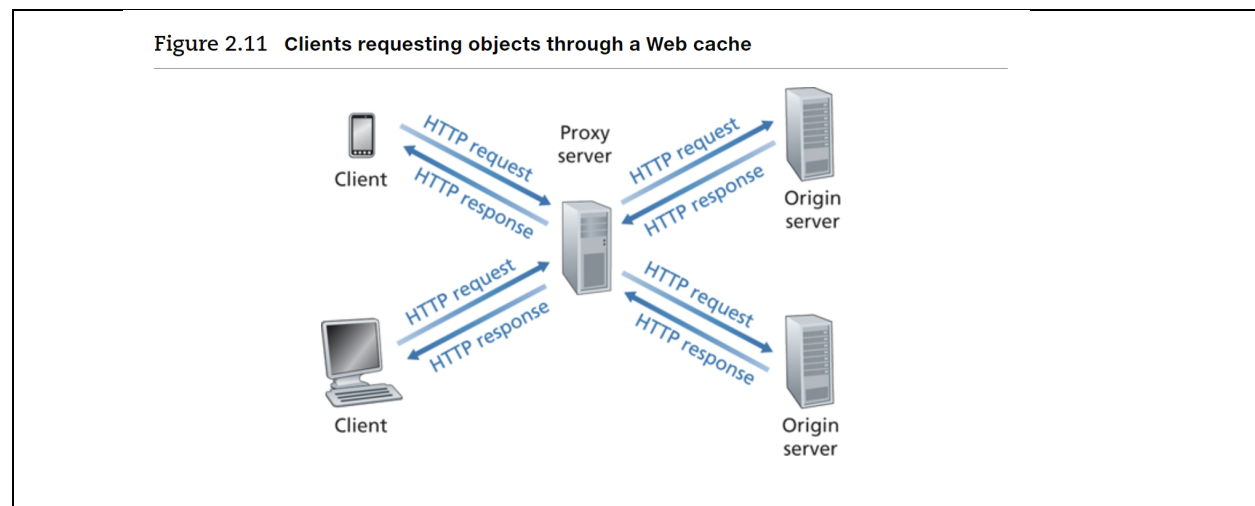
## Current Status

- (1) We will complete our discussion on http caches today.
- (2) Today we will discuss smtp to complete chapter two.
- (3) We will start chapter 3 today.
- (4) Check 1 has been released. (Due on 28 September) Start early. Read whole doc, think first, discuss with your team-member before starting. Do it in many small sessions.
- (5) Sessional-1 exam coming. (See official datasheet for day and time.)
- (6) Start picking topics for presentation 1 in the doc I mentioned earlier.

Problem: What if multiple parties are asking for the same object again and again?

In an organization like FAST, what if many students are fetching [www.nu.eud.pk](http://www.nu.eud.pk) multiple times in a day. FAST's logo will still be fetched once and cached locally in the browser cache. But can we do better? Can we have a cache for the organization?

Solution 10: Web Caching vi a proxy server (at times called forward proxy)



## An aside:

When HTTPS was not widely used, web caches were common.

Now, when HTTPS is a norm, parties who have session key can cache only.

Example: Client's browser cache

Front-end server close to the customer where a TCP connection terminate

There is a topic with the name "**split TCP**" in the presentation 1 list. Usually, split TCP is used where client's original connection terminates at a front-end proxy, and then data is sent on a different TCP connection from front-end to the origin server(s).

Let's switch to second slide deck (already posted on GCR), at slide no 42 for caches discussion.

## Problem: How to do stateful work on stateless HTTP?

Remember HTTP is a stateless request-response protocol. But many applications need state across multiple requests.

Examples: Buying books online, logged-in customer of bank etc.

## Solution 11: Use cookies to create application-level state

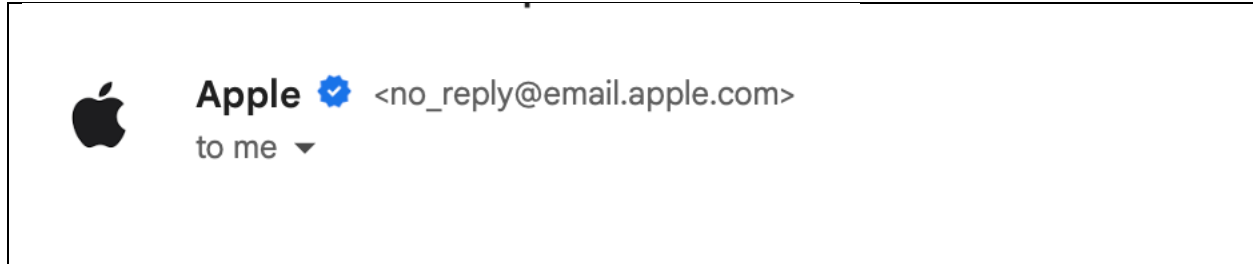
Let's switch to second slide deck for edition 9 of the textbook (already posted on GCR), at slide no 32 for state discussion.

That completes our overview of the HTTP protocol. See the wiki page of HTTP or read the relevant HTTP RFCs for more details.

## Simple Mail Transfer Protocol (SMTP)

Let's go to slide no 59.

## Fighting SPAM – SPF, DKIM, DMARC



|        |                                                               |
|--------|---------------------------------------------------------------|
| SPF:   | PASS with IP 17.111.110.68 <a href="#">Learn more</a>         |
| DKIM:  | 'PASS' with domain email.apple.com <a href="#">Learn more</a> |
| DMARC: | 'PASS' <a href="#">Learn more</a>                             |