CS118 Discussion 1D, Week 2

Zengwen Yuan
Boelter Hall 2760, Friday 4:00—5:50 p.m.
Outline

• Lecture Review: Application Layer

• Project 1 spec
Application Layer: models

- Application Architectures
  - Client-server model: Web (TCP), FTP (TCP), E-mail (TCP), DNS (UDP/TCP), RTP
  - Peer-to-Peer (P2P): BitTorrent (TCP), Tor (aka Onion Routing, TCP)
  - Hybrid model: Skype (TCP&UDP), GTalk (TCP&UDP)
Application Layer: protocols

- HTTP: a stateless protocol on top of TCP
  - HTTP is based on pull model
  - Persistent HTTP V.S. Non-persistent HTTP
- Method Types: GET, HEAD, POST, PUT, DELETE, Conditional GET
  - What if we want stateful service (e.g. shopping cart)?
  - Web Caches (proxy server)
Non-persistent v.s. Persistent v.s. Pipelining
Quiz

• How many TCP connections do we need to get one HTML file with 5 embedded images? How many RTTs shall we need?
HTTP Header: request

- Request message elements:
  - Method
  - URL
  - HTTP Version
  - Header lines
  - CRLF
HTTP Header: response

• Response message elements:
  • HTTP Version
  • Status line
  • Header lines
  • CRLF
  • Data requested
Try HTTP GET yourself

• telnet google.com 80
  • Get / HTTP/1.0
  • Host: google.com
  • <Enter>
  • <Enter>
Cookie

• Bring statefulness into HTTP

• Components
  • Cookie header line of HTTP response message
  • Cookie header line of HTTP request message
  • Cookie file on the browser
  • Back-end database at web-site

• Types of Cookies
  • Session Cookies (eg. shopping cart)
  • Persistent Cookies (eg. authentication id and password)
  • Third Party Cookies (eg. ad tracking agencies)
Cookie: make HTTP stateful

HTTP Client

Login POST
username=david
password=davidh

Set-Cookie: SESSIONID=66C530ACAF44D1605588619ECB0C737C

HTTP is Stateless

HTTP Server

Login successful?
1. create session id
2. return session id in cookie
3. store session id in database

SESSION_ID
SessionId
Username
createDate
expireDate
lastAccessDate

Database

Lookup Session ID
1. session match a username?
2. session still valid?

Content for ‘david’
Cookie: operations

**Client:** has a cookie file

- eBay 8734
- Amazon 1678

**Web server**

- Amazon server creates ID 1678 for user

**one week later:**

- eBay 8734
- Amazon 1678

- Usual HTTP request msg
- **Cookie:** 1678
- Usual HTTP response msg
Web caching: Proxy v.s. CDN

- Proxy acts both as client and server
  - What if cache is stale?
    - HTTP conditional GET
- CDN: Content Distribution Network
  - Globally distributed network of web servers
  - Stores and replicates images, videos and other files
Application Layer: protocols

- FTP: separate control from data transmission ("out-of-band")
- SMTP: protocol for email exchange between email servers
  - SMTP is based on push model
- Mail access protocol: POP, IMAP, HTTP-based
- P2P: no always-on server, peers are intermittently connected
  - BitTorrent: tracker and torrent. Files are divided into multiple chunks.
Application Layer: protocols

- DNS: convert hostname to IP address (and more)
- A distributed and hierarchical database
  - Root DNS servers (a—m)
  - Top-level domain (TLD) servers
  - Authoritative DNS servers
  - Local DNS server
Application Layer: protocols

- DNS:
  - What is the transport layer protocol?
  - How the scalability is achieved?
  - Who will use iterative/recursive query?
  - Why is DNS resolver needed?
Project 1 Demo

• How to use Makefile

• See HTTP request received by our server program

• Serve content to a standard web browser
Clarifications on Byte order conversion


• Make a byte-order conversion if you have the data type longer than a byte!