HTTP AND THE WEB

George Porter
Jan 23, 2019

ATTRIBUTION

• These slides are released under an Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0) Creative Commons license
• These slides incorporate material from:
  • Computer Networks: A Systems Approach, 5e, by Peterson and Davie
Outline

1. HTTP protocol
2. Demo: interacting with web servers

HTTP AS AN EMERGING TRANSPORT LAYER

• HTTP: HyperText Transfer Protocol
  • Tim Berners-Lee at CERN in 1989
• In addition to web browsing:
  • Video streaming via DASH on YouTube.com
  • REST (Representational state transfer)
  • Chat apps like Slack
  • Many others
WEB/HTTP OVERVIEW

- Documents link to other documents
  - Specified in HTML files
- HTTP is the protocol for retrieving HTML files from servers
  - and images, sounds, video, ...
- Implemented in servers
  - Apache, nginx, MSFT IIS
- and clients
  - Chrome
  - MSFT Edge
  - Apple Safari...

SAMPLE HTML FILE

http://cseweb.ucsd.edu/~gmporter/test.html

```
<html>
<head>
<title>It works!</title>
</head>
<body>
<h1>It works</h1>
nice!
</body>
</html>
```
HTTP OVERVIEW

• HTTP is a text oriented protocol.
• HTTP is a request/response protocol.
• Requests and responses both look like:
  
  START_LINE <CRLF>
  MESSAGE_HEADER <CRLF>
  <CRLF>
  MESSAGE_BODY <CRLF>

  The first line (START LINE) indicates whether this is a request message or a response message.

HTTP REQUESTS

• Request Messages define
  
  • The operation (called method) to be performed
  • The web page the operation should be performed on
  • The version of HTTP being used.

• Examples:
  
  • GET /index.html HTTP/1.0
  • GET /images/catimg23.jpg HTTP/1.1
  • GET /contracts/contract3.txt HTTP/1.1
OPTIONAL HTTP REQUEST HEADERS

- After the start line are request headers:
  - Text-based, key and value separated by a colon
  - Example 1:
    
    GET /index.html HTTP/1.0  
    User-Agent: Firefox 23.3.1

- Example 2:
  
  GET /images/cat2.jpg HTTP/1.1  
  Host: www.cs.ucsd.edu 
  User-Agent: Chrome 12.1

HTTP RESPONSES

- Also begins with a single START LINE.
  - The version of HTTP being used
  - A three-digit status code
  - Text string giving the reason for the response.

- Example:
  
  HTTP/1.1 200 OK  
  Content-Type: text/html  
  Content-Length: 291
borabora:~ gmporter$ telnet oec-vmweb09.ucsd.edu 80
Trying 132.239.8.67...
Connected to oec-vmweb09.ucsd.edu.
Escape character is '^]'.

GET /index.html HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 12 Jan 2015 19:36:37 GMT
Server: Apache/2.2.22 (Ubuntu)
Last-Modified: Thu, 28 Feb 2013 17:35:36 GMT
ETag: "fc7b21-a-4d6cc51858aec"
Accept-Ranges: bytes
Content-Length: 10
Vary: Accept-Encoding
Connection: close
Content-Type: text/html

It works!
Connection closed by foreign host.
borabora:~ gmporter$
HTTP PIPELINING (VERSION HTTP/1.1)

• HTTP/1.0 opened a new connection for every data item it retrieved
• Overhead in establishing a new connection to the same server over and over again
• HTTP/1.1 Persistent Connections
  • Reuse connection over many requests/responses
  • But more complex in terms of framing/parsing
    • How to know when one request ends and the next begins?
    • This is part of the 1.1 spec

REQUIRED REQUEST HEADERS (AT LEAST FOR US)

• Host:
  • Indicates the name of the server you are accessing
  • Used to implement virtual hosts
REQUIRED RESPONSE HEADERS (AT LEAST FOR US)

- **Server:**
  - Identifies the server
    - Server: Apache/2

- **Content-Length:**
  - How many octets (byte) in the response

- **Content-Type:**
  - text/html
  - image/jpeg
  - image/png

PROJECT 1 WALK-THROUGH
Outline

1. HTTP protocol
2. Demo: interacting with web servers

DEMO: INTERACTING WITH WEB SERVERS

• Usage:
  • curl -v -o /dev/null http://<URL>