



15-441 Computer Networking

Lecture 25 – The Web

Outline

- HTTP review and details (more in notes)
- Persistent HTTP review
- HTTP caching
- Content distribution networks

Lecture 10: 2008-11-02

3

HTTP Basics (Review)

- HTTP layered over bidirectional byte stream
 - Almost always TCP
- Interaction
 - Client sends request to server, followed by response from server to client
 - Requests/responses are encoded in text
- Stateless
 - Server maintains no information about past client requests

Lecture 10: 2008-11-02

4

How to Mark End of Message? (Review)

- Size of message → Content-Length
 - Must know size of transfer in advance
- Delimiter → MIME-style Content-Type
 - Server must "escape" delimiter in content
- Close connection
 - Only server can do this

Lecture 10: 2008-11-02

5

HTTP Request (review)

- Request line
 - Method
 - GET – return URI
 - HEAD – return headers only of GET response
 - POST – send data to the server (forms, etc.)
 - URL (relative)
 - E.g., /index.html
 - HTTP version

Lecture 10: 2008-11-02

6

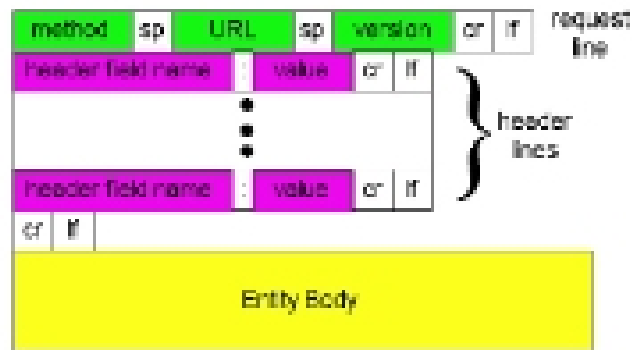
HTTP Request (cont.) (review)

- Request headers
 - Authorization – authentication info
 - Acceptable document types/encodings
 - From – user email
 - If-Modified-Since
 - Referrer – what caused this page to be requested
 - User-Agent – client software
- Blank-line
- Body

Lecture 10: 2008-11-02

7

HTTP Request (review)



Lecture 10: 2000-11-02

7

HTTP Request Example (review)

```
GET / HTTP/1.1
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT
5.0)
Host: www.intel-irls.net
Connection: Keep-Alive
```

Lecture 10: 2000-11-02

8

HTTP Response (review)

- Status-line
 - HTTP version
 - 3 digit response code
 - 1XX – informational
 - 2XX – success
 - 200 OK
 - 3XX – redirection
 - 301 Moved Permanently
 - 302 Moved Temporarily
 - 304 Not Modified
 - 4XX – client error
 - 404 Not Found
 - 5XX – server error
 - 502 HTTP Version Not Supported
 - Reason phrase

Lecture 10: 2000-11-02

9

HTTP Response (cont.) (review)

- Headers
 - Location – for redirection
 - Server – server software
 - WWW-Authenticate – request for authentication
 - Allow – list of methods supported (get, head, etc)
 - Content-Encoding – E.g. x-gzip
 - Content-Length
 - Content-Type
 - Expires
 - Last-Modified
- Blank-line
- Body

Lecture 10: 2000-11-02

10

HTTP Response Example (review)

```
HTTP/1.1 200 OK
Date: Tue, 27 Mar 2001 03:49:38 GMT
Server: Apache/1.3.14 (Unix) (Red-Hat/Linux) mod_ssl/2.7.1
OpenSSL/0.9.5a DAV/1.0.2 PHP/4.0.1 pl2 mod_perl/1.24
Last-Modified: Mon, 29 Jan 2001 17:54:18 GMT
ETag: "7a11f10ed-3a75ee4e"
Accept-Ranges: bytes
Content-Length: 4333
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html
-----
```

Lecture 10: 2000-11-02

11

Outline

- HTTP intro and details
- **Persistent HTTP**
- HTTP caching
- Content distribution networks

Lecture 10: 2000-11-02

12

Typical Workload (Web Pages)

- Multiple (typically small) objects per page
- File sizes
 - Heavy-tailed
 - Pareto distribution for tail
 - Lognormal for body of distribution
- Embedded references
 - Number of embedded objects =
pareto - $p(x) = ak^x \cdot x^{-(k+1)}$

Lecture 10: 2008-11-02

12

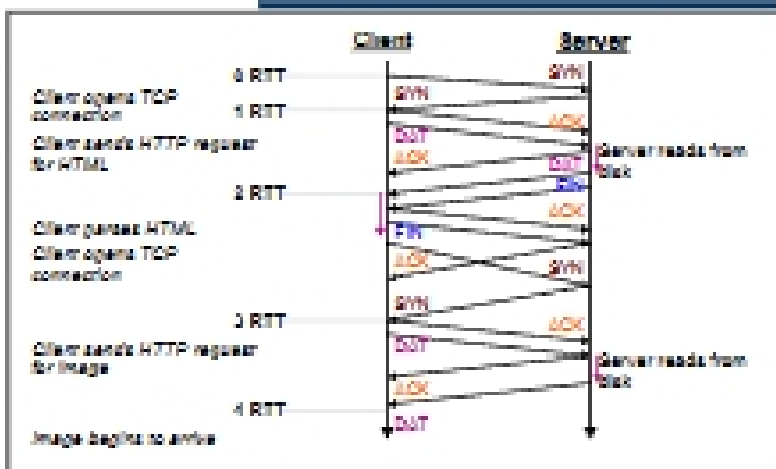
HTTP 0.9/1.0 (mostly review)

- One request/response per TCP connection
 - Simple to implement
- Disadvantages
 - Multiple connection setups → three-way handshake each time
 - Several extra round trips added to transfer
 - Multiple slow starts

Lecture 10: 2008-11-02

13

Single Transfer Example



Lecture 10: 2008-11-02

14

More Problems

- Short transfers are hard on TCP
 - Stuck in slow start
 - Loss recovery is poor when windows are small
- Lots of extra connections
 - Increases server state/processing
- Server also forced to keep TIME_WAIT connection state
 - Things to think about:
 - Why must server keep these?
 - Tends to be an order of magnitude greater than # of active connections, why?

Lecture 10: 2008-11-02

15

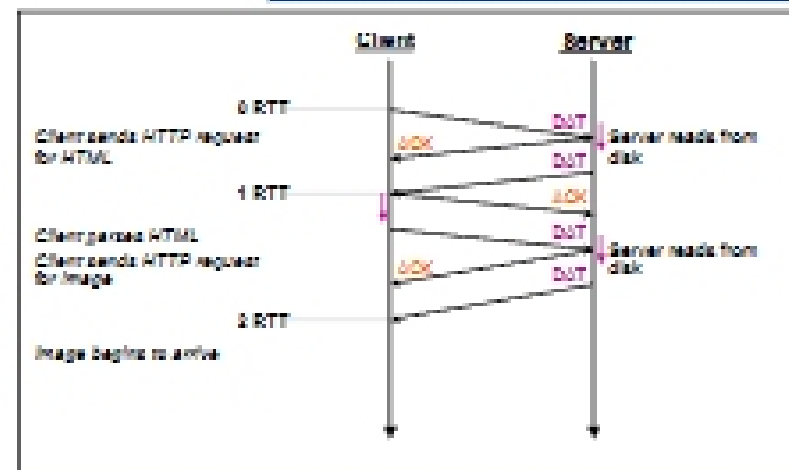
Persistent Connection Solution (review)

- Multiplex multiple transfers onto one TCP connection
- How to identify requests/responses
 - Delimiter → Server must examine response for delimiter string
 - Content-length and delimiter → Must know size of transfer in advance
 - Block-based transmission → send in multiple length delimited blocks
 - Store-and-forward → wait for entire response and then use content-length
 - Solution → use existing methods and close connection otherwise

Lecture 10: 2008-11-02

16

Persistent Connection Example (review)



Lecture 10: 2008-11-02

17