



L5 – HTTP

by

T.S.R.K. Prasad

EA C451 Internetworking Technologies

21/01/2013

References / Acknowledgements



Dr. Markus Hoffman for Slides

(The slides have been adopted from presentations made by Dr. Markus Hofmann, Head of Bell Labs Research, and Adjunct Professor, Columbia University for the book *Content Networking: Architecture, Protocols, and Practice*, Markus Hoffman and Leland R. Beaumont, Morgan Kaufmann, 2005.)

Sec 2.2: HTTP, [Hoffman]

Sec 2.2: The Web and HTTP, [Kurose]

Sec 9.1.2: World Wide Web (HTTP), [Peterson]



Optional Readings

[rfc2616] - Hypertext Transfer Protocol – HTTP/1.1

[rfc2817] - Upgrading to TLS Within HTTP/1.1

[rfc3205] - On the use of HTTP as a Substrate

[rfc6265] - HTTP State Management Mechanism(Cookies)

[rfc6797] - HTTP Strict Transport Security (HSTS)

[Pantos-HTTP] - HTTP Live Streaming (RFC Draft)

<http://tomayko.com/writings/on-http-abuse>

<http://tomayko.com/writings/rest-to-my-wife>



Presentation Overview

Latest Twist

Examples

Cookies

HTTP/1.1

HTTP Overview



Presentation Overview

Latest Twist

Examples

Cookies

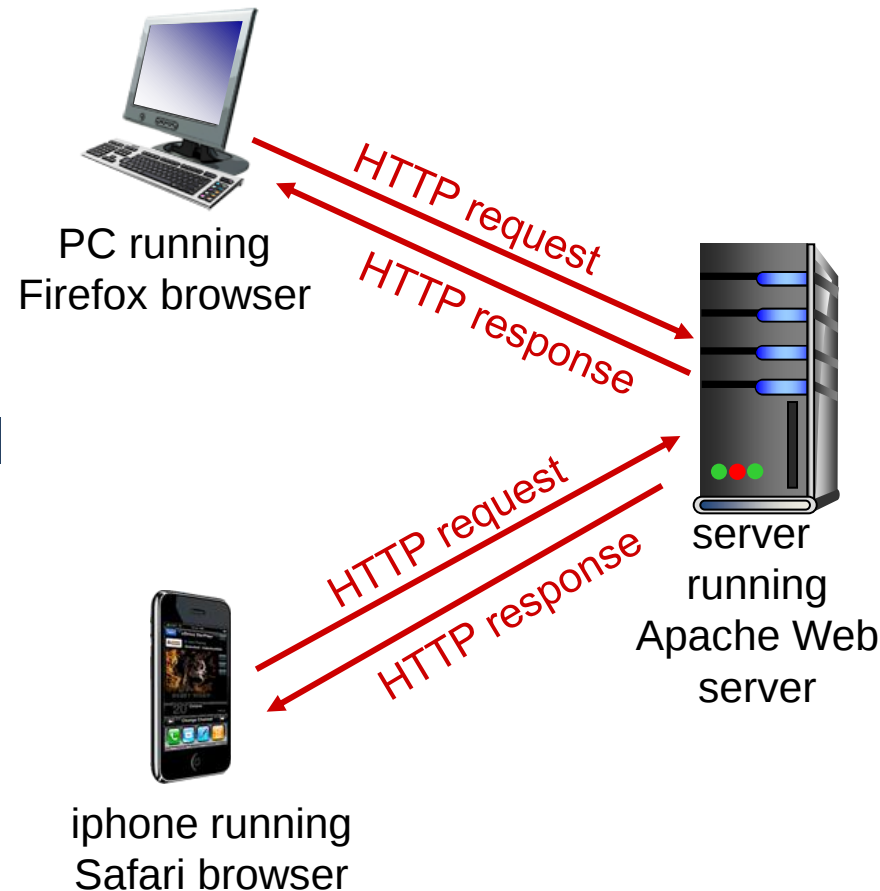
HTTP/1.1

HTTP Overview

HTTP overview



client/server model
of communication





HTTP Protocol Properties

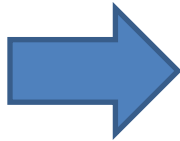
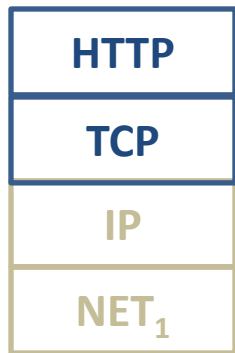
- **Statelessness**
 - Web server maintains no information about past client requests.
- **Global URI**
 - HTTP uses URIs in all its transactions to identify resources.
- **Resource metadata**
 - Web transfers can include information about resources.
- **Request/Response Exchange**
 - Requests are sent by clients, responses are generated by servers.
 - Transactions are initiated by clients only.

aside

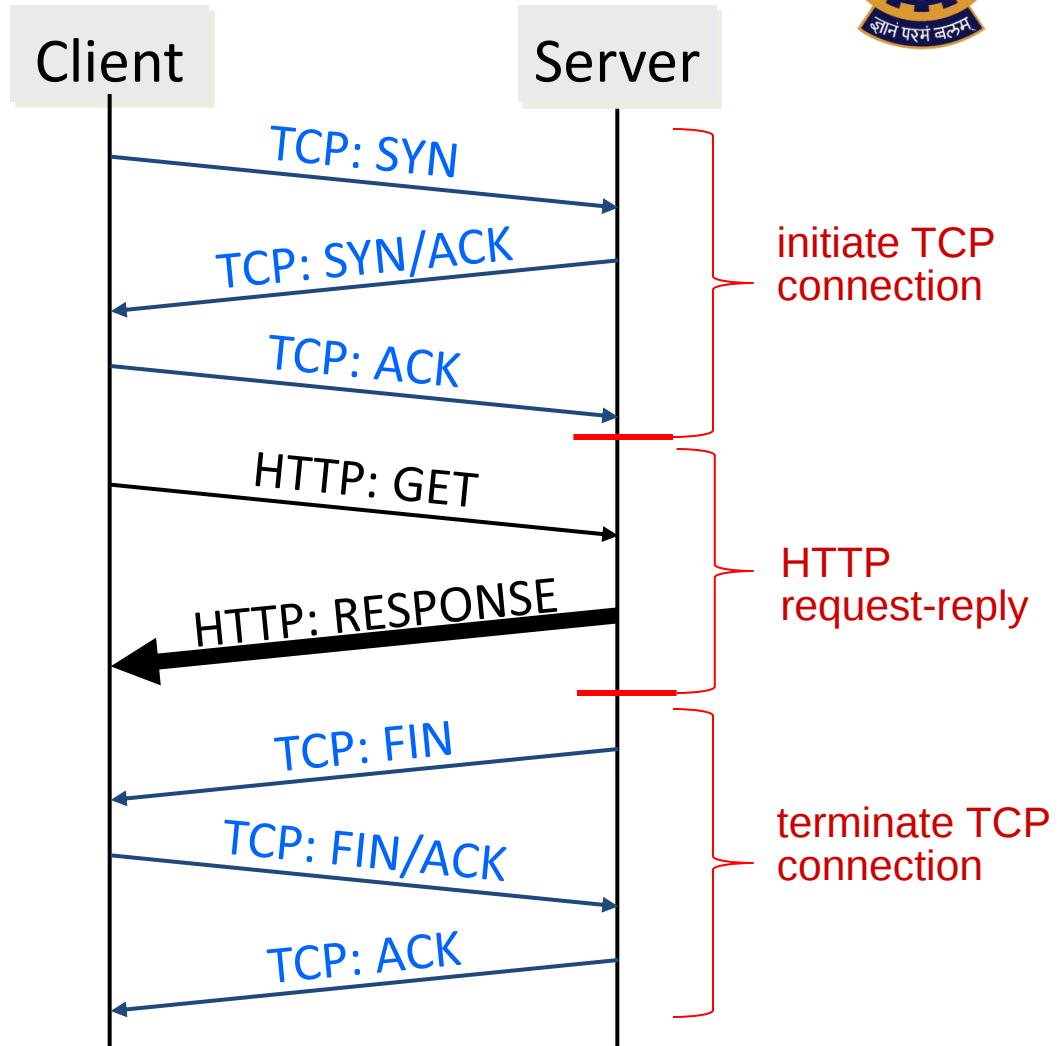
Protocols that maintain “state” are complex!

- Past history (state) must be maintained.
- If server/client crashes, their views of “state” may be inconsistent; must be reconciled.

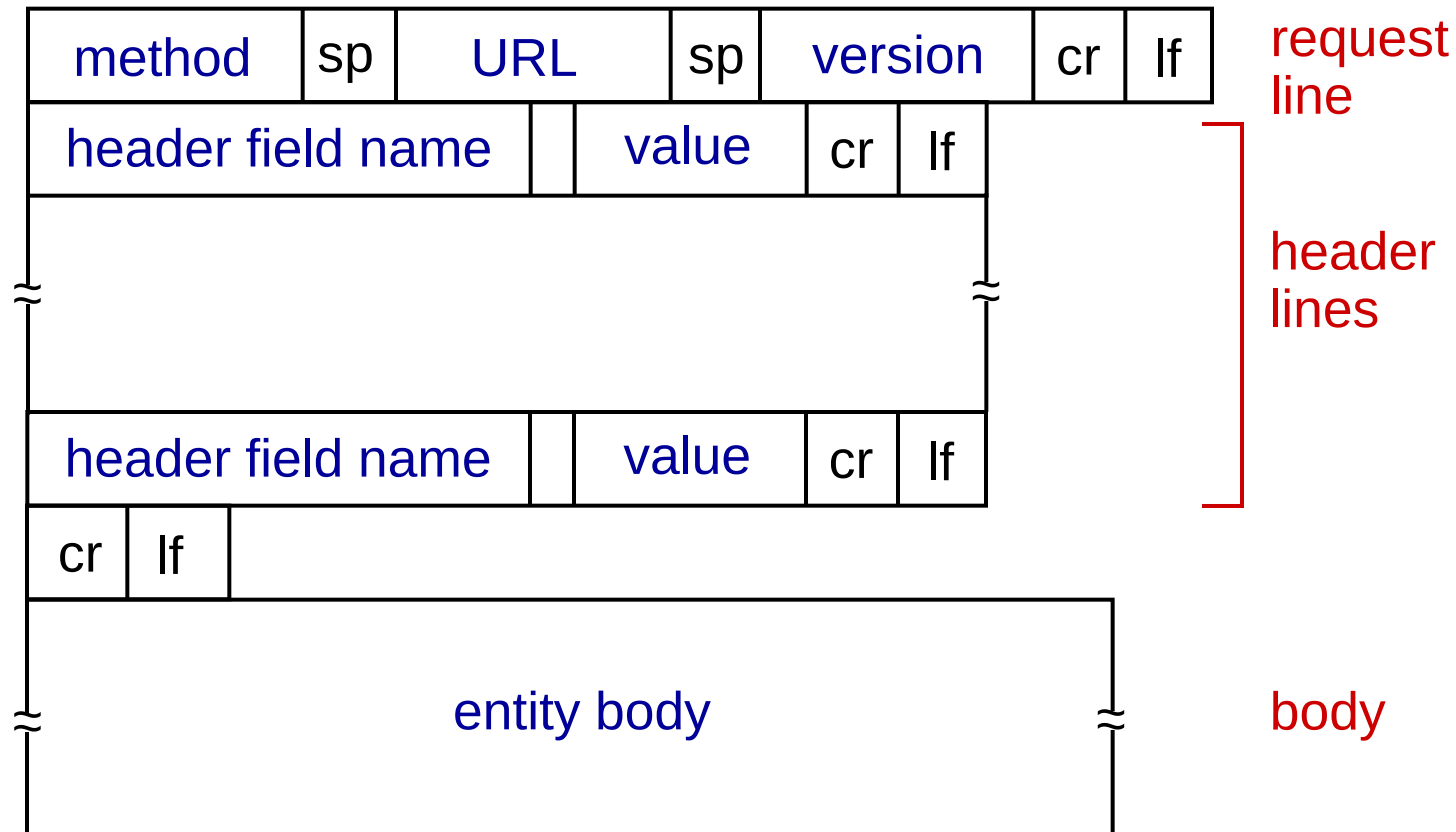
HTTP/1.0 Transactions



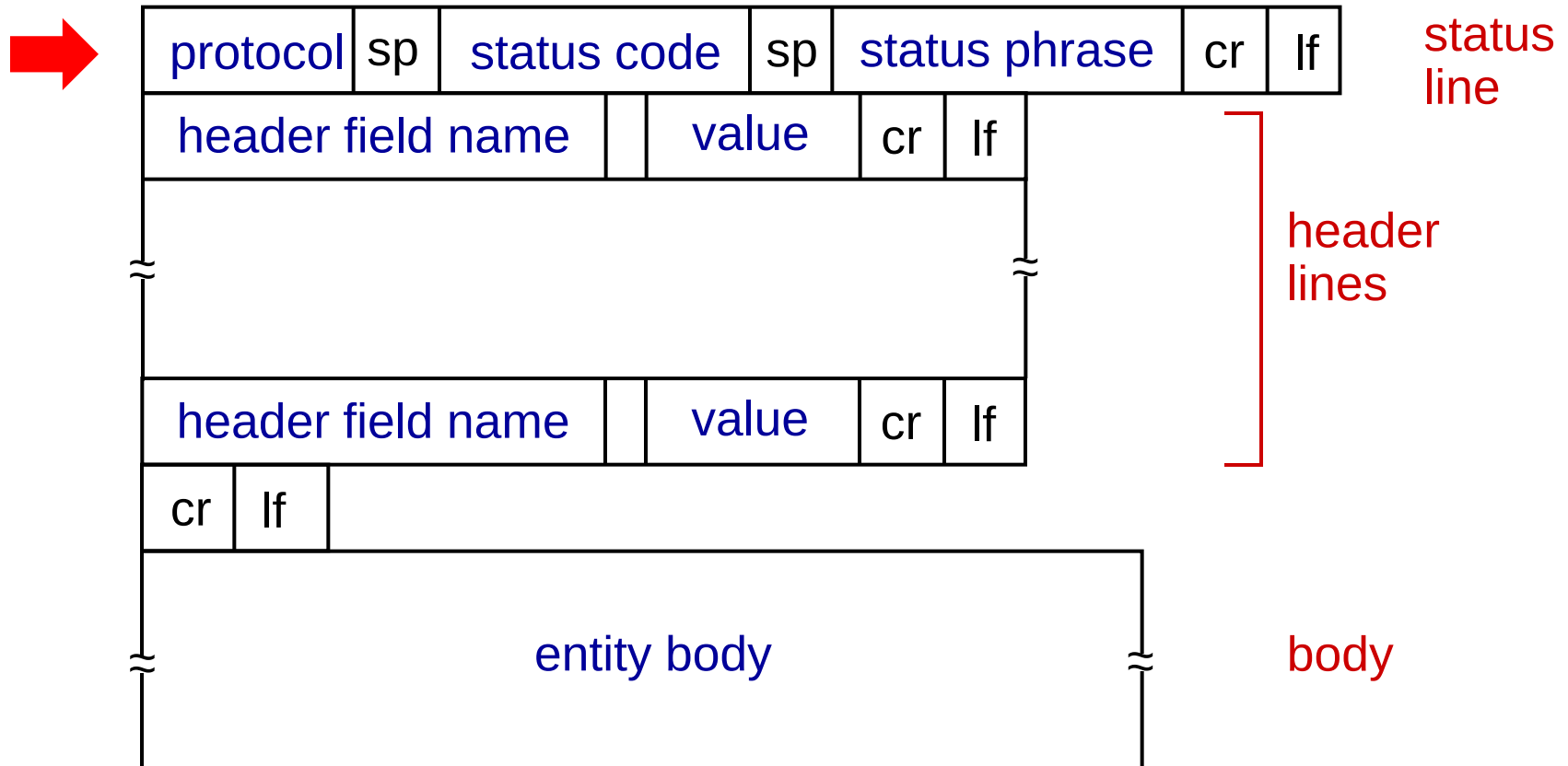
All the steps are repeated for each object



Format of HTTP Request Message



Format of HTTP Response Message





A Sample HTTP Request

Request Line
(GET, POST,
HEAD etc methods)

`GET /index.html HTTP/1.0`

Header
Lines

`Host: www.mhof.com`

`Date: Wed, 24 Jan 2007 06:24 GMT`

`User-Agent: Mozilla/5.0 [en] (WinNT; U)`

`Accept: text/html image/gif, image/jpeg,`

`Accept-Language: en-us`

Carriage Return,
Line Feed
indicates end
of message

[no message body here]



A Sample HTTP Response

Status Line
(protocol,
status code,
status phrase)

HTTP/1.0 200 OK

Header
Lines

Date: Wed, 24 Jan 2007 06:25 GMT

Server: Apache/1.3.33 (Unix)

Last-Modified: Sun, 21 Jan 2007 ...

Content-Type: text/html

Carriage Return,
Line Feed

data data data data...

Data (e.g.
requested
HTML file)



HTTP/1.0 Request Methods

- **GET**
 - For object retrieval
- **POST**
 - For returning data (e.g. from an input form) to the server; data is included in the message body
- **HEAD**
 - Returns only the header of the response message (often used for debugging).



HTTP Response Status Codes

- Contained in the first line of the HTTP response message
- First digit of the status code specifies the class of response:
 - **1xx: Informal** – Request received, continuing process.
 - **2xx: Success** – The action was successfully received, understood, and accepted.
 - **3xx: Redirection** – Further action must be taken in order to complete the request.
 - **4xx: Client Error** – The request contains bad syntax or cannot be fulfilled.
 - **5xx: Server Error** – The server failed to fulfill an apparently valid request.
- Status code is augmented with a reason phrase.
 - Meant for human interpretation.

The “Real” Meaning of HTTP Status Codes



- Humorous interpretation of a Web server's response:

- **1xx: Informal** – Not done yet.
- **2xx: Success** – You win.
- **3xx: Redirection** – You loose, but try again.
- **4xx: Client Error** – You loose, your fault.
- **5xx: Server Error** – You loose, my bad.



Example HTTP Response Status Codes



- **200 OK**
 - Request succeeded, requested object enclosed in this message.
- **301 Moved Permanently**
 - Requested object moved permanently, new location specified in the “Location” field.
- **302 Found**
 - The requested resource resides temporarily under a different URI.
 - Temporary location specified in “Location” field

Example HTTP Response Status Codes



- **400 Bad Request**
 - Request message not understood by server.
- **404 Not Found**
 - Requested document not found on this server.
- **500 Internal Server Error**
 - Server encountered internal error and is unable to determine the precise error condition.
- **505 HTTP Version Not Supported**

Trying out HTTP (client side) for yourself



1. Telnet to your favorite Web server:

```
telnet www.mhof.com 80
```

- Opens TCP connection to port 80 at www.sun.com.
- Anything typed in sent to port 80 at www.sun.com.

2. Type in a GET http request:

```
GET /index.html HTTP/1.0
```

By typing this in (hit carriage return *twice*), you send this minimal (but complete) GET request to http server

Conditional GET

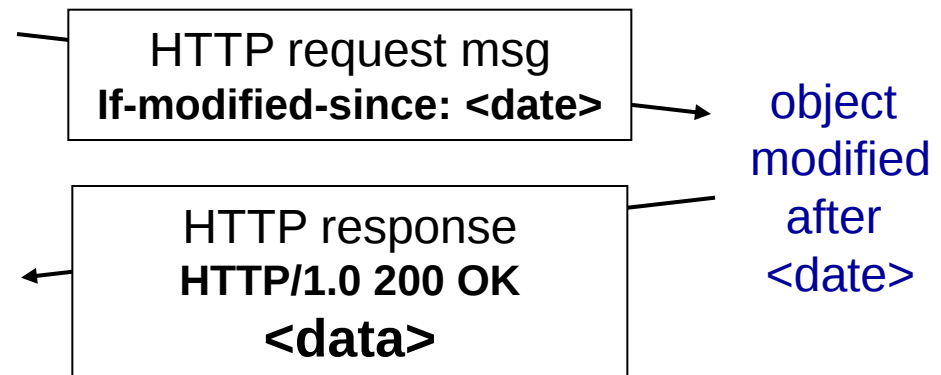
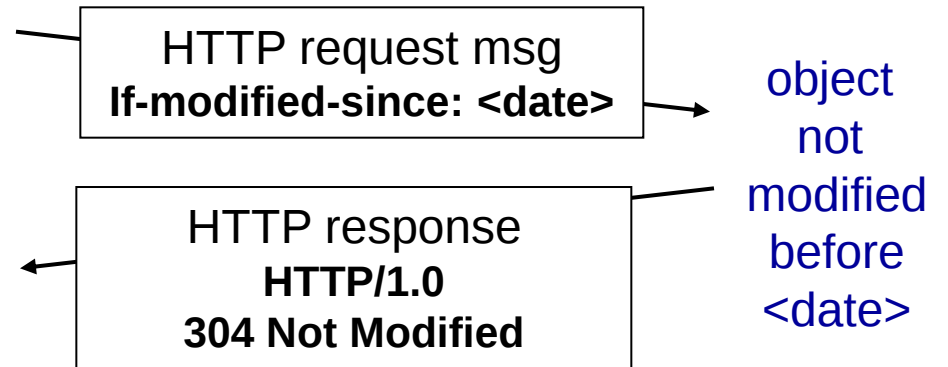


client



server

Useful to caches





Presentation Overview

Latest Twist

Examples

Cookies

HTTP/1.1

HTTP Overview



Main Improvements in HTTP/1.1

- **Persistent connections**
 - Multiple HTTP object transfers with pipelining for one TCP connection
- **Compatibility with caches**
 - Special headers for caches
Ex: cache-control

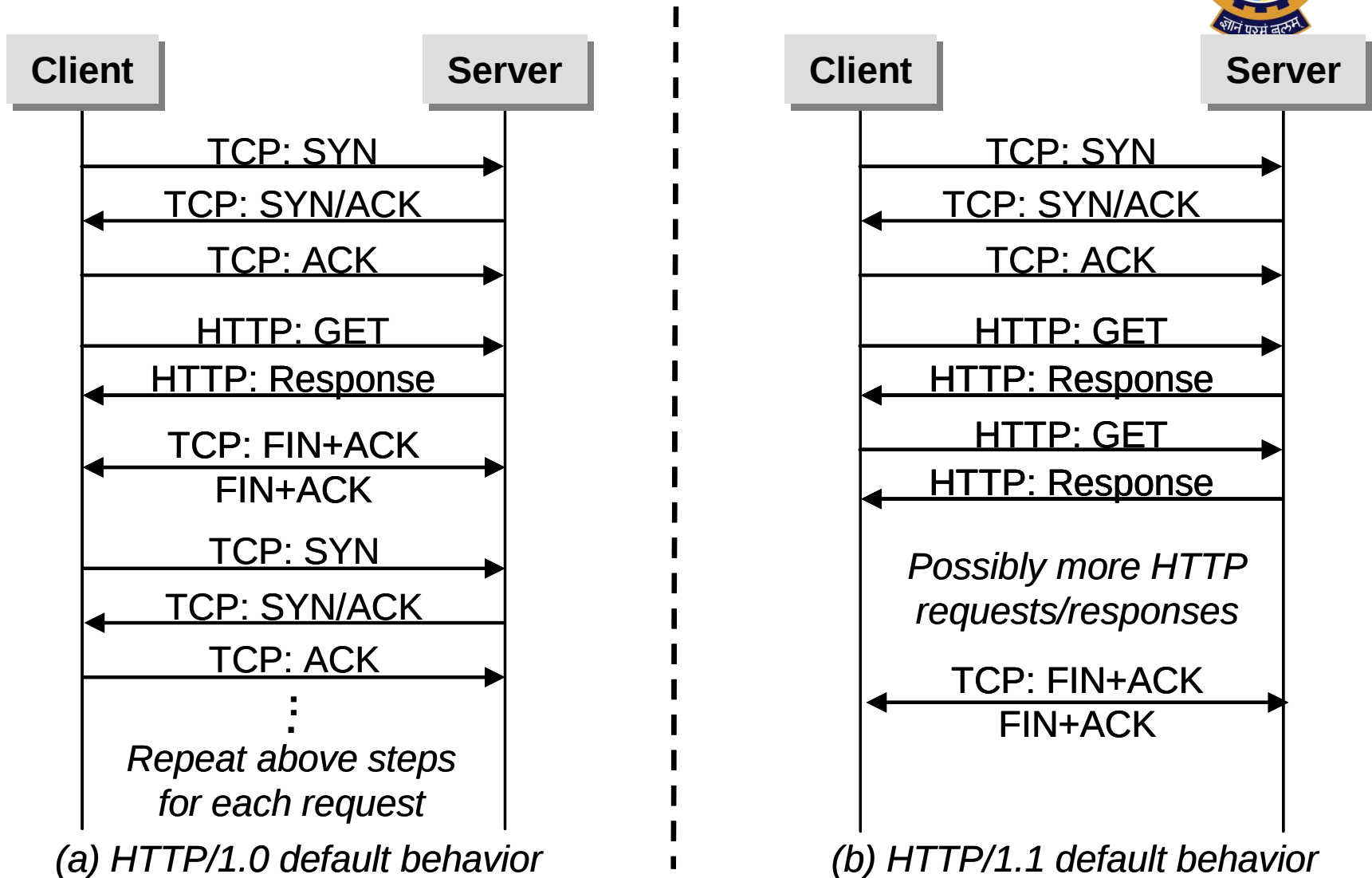


More on Persistent Connections

- After sending the first response, the server leaves the TCP connection open.
- Subsequent requests and responses to the same server are sent over the persistent connection.
 - Pipelining: Client can issues a new requests at any time.
 - No pipelining: Client issues a new request only after receiving a previous response.
- Either the client closes the connection explicitly, or the server terminates it after a certain idle time.



(Non-) Persistent Connections



(Non-) Persistent Connections (contd.)



Non-persistent

- HTTP/1.0
- Server parses request, responds, and closes TCP connection
- 2 RTTs to fetch each object
- Each object transfer suffers from slow start

Persistent

- Default for HTTP/1.1
- On same TCP connection: server parses request, responds, parses new request, ...
- Client sends requests for all referenced objects as soon as it receives base HTML.
- Fewer RTTs and less slow start.

But most 1.0 browsers use
parallel TCP connections
=> Increased server load!



Chunked Encoding

- Problem: How does a receiver recognize that a message has been received in full?
- **Solutions in HTTP/1.0:**
 - Include “Content-Length” field in HTTP response header – problematic with dynamically created content.
 - Server indicates the end of a message by closing the TCP connection – does not work with persistent connections.
- **Solution introduced in HTTP/1.1: Chunked encoding**
 - Allows the sender to break a message body into arbitrary sized chunks and send them separately.
 - Works for requests as well as for responses.
 - End of message transportation is indicated by a zero-length chunk.

Chunked Encoding - Example



```
HTTP/1.1 200 OK
Server: Apache/1.3.33 (Unix)
Date: Wed, 24 Jan 2007 06:25 GMT
Transfer-Encoding: chunked
Content-Type: text/html
```

1 st Chunk	21E	<...542 bytes of chunked data...>
2 nd Chunk	61	<...97 bytes of chunked data...>
3 rd Chunk	0	



HTTP/1.1 Request Methods

- **PUT**
 - For creating or modifying a Web object associated with the Request-URI
- **DELETE**
 - For removing the Web object identified by the Request-URI from the Web server
- **TRACE**
 - For invoking a remote application-layer loopback
- **OPTIONS**
 - For requesting communication options available on the Web server



Presentation Overview

Latest Twist

Examples

Cookies

HTTP/1.1

HTTP Overview

Cookies



- Mechanism to introduce state into HTTP interactions
- Server sends SET-COOKIE as part of response header
- Client saves the cookie in a cookie file
- In subsequent requests to the same server, the client software checks the cookie file and includes available cookie information in the COOKIE request header.
- Server can add/modify cookies using the SET-COOKIE header again.
- Application Example:
 - Server assigns ID to client on the first access and returns the number in the SET-COOKIE response header.
 - Whenever user accesses the site again, it sends ID number in the COOKIE request header.

Maintaining “state”



client



server



cookie file



ebay 8734
amazon 1678

usual http request msg

Amazon server
creates ID
1678 for user

usual http response
set-cookie: 1678

create
entry

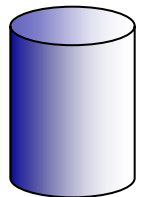
backend
database

usual http request msg
cookie: 1678

cookie-
specific
action

access

usual http response msg



access

cookie-
specific
action

one week later:



ebay 8734
amazon 1678

usual http request msg
cookie: 1678

usual http response msg

Cookies Summary



what cookies can be used for:

- authorization
- shopping carts
- recommendations
- user session state (Web e-mail)

aside
cookies and privacy.

- ❖ cookies permit sites to learn a lot about you
- ❖ you may supply name and e-mail to sites

how to keep “state”:

- ❖ protocol endpoints: maintain state at sender/receiver over multiple transactions
- ❖ cookies: http messages carry state



Presentation Overview

Latest Twist

Examples

Cookies

HTTP/1.1

HTTP Overview

MOODLE Login Page Request



GET /moodle/login/index.php HTTP/1.1

Host: 10.1.1.242

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:14.0) Gecko/20100101

Firefox/14.0.1

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: en-us,en;q=0.5

Accept-Encoding: gzip, deflate

Connection: keep-alive

Response for MOODLE Login

Page Request



HTTP/1.1 200 OK

Date: Fri, 17 Aug 2012 23:42:11 GMT

Server: Apache

X-Powered-By: PHP/5.3.9-ZS5.6.0 ZendServer/5.0

Set-Cookie: ZDEDebuggerPresent=php,html,php3; path=/
Set-Cookie: MoodleSession=q3j7dgfdg67f6dekemv8fr4fk0; path=/
Expires:

Cache-Control: private, pre-check=0, post-check=0, max-age=0

Pragma: no-cache

Content-Language: en

Content-Script-Type: text/javascript

Content-Style-Type: text/css

Accept-Ranges: none

X-Frame-Options: sameorigin

Vary: Accept-Encoding,User-Agent

Content-Encoding: gzip

Content-Length: 5182

Connection: close

Content-Type: text/html; charset=utf-8

HTTP Requests for displaying http://10.1.1.242/moodle



▼ HTTP Requests by HTTP Host

▼ 10.1.1.242

/moodle/login/index.php
/moodle/theme/yui_combo.php?3.5.1/build/cssreset/reset-min.css&3.5.1/build/cssfonts/fonts-min.css&3.5.1/build/cssgrids/grids-min.css&3.5.1/build/cssbase/base-min.css
/moodle/theme/yui_combo.php?3.5.1/build/yui/yui-min.js
/moodle/theme/yui_combo.php?2.9.0/build/yahoo/yahoo-min.js&2.9.0/build/dom/dom-min.js
/moodle/theme/yui_combo.php?2.9.0/build/assets/skins/sam/skin.css
/moodle/theme/styles.php/bitstheme/1343020318/all
/moodle/lib/javascript.php/1342174251/lib/javascript-static.js
/moodle/theme/javascript.php/bitstheme/1343020318/head
/moodle/theme/javascript.php/bitstheme/1343020318/footer
/moodle/pix/BITS-bar.png
/moodle/theme/image.php/bitstheme/core/1343020318/help
/moodle/theme/image.php/bitstheme/theme/1343020318/footer/moodle-logo
/moodle/pix/lead.png
/moodle/theme/image.php/bitstheme/theme/1343020318/favicon
/moodle/theme/image.php/bitstheme/theme/1343020318/core/bgground
/moodle/pix/photon-t.png
/moodle/theme/image.php/bitstheme/theme/1343020318/images/logo
/moodle/theme/image.php/bitstheme/theme/1343020318/menu/nav-arrow-right
/moodle/theme/image.php/bitstheme/theme/1343020318/images/light3
/moodle/theme/image.php/bitstheme/theme/1343020318/core/h2grad
/moodle/theme/yui_combo.php?3.5.1/build/ooop/ooop-min.js&3.5.1/build/event-custom-base/event-custom-base-min.js&3.5.1/build/dom-core/dom-core-min.js&3.5.1/build/dom-base/

21 HTTP GET requests to display a page

TCP Connections for displaying http://10.1.1.242/moodle



Conversations: eth0 (port 80)

Ethernet: 2 Fibre Channel FDDI IPv4: 1 IPv6 IPX JXTA NCP RSVP SCTP **TCP: 21** Token Ring UDP USB WLAN

TCP Conversations - Filter: ip.addr == 10.1.1.242

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A→B	Bytes A→B	Packets A←B	B
10.5.2.44	60073	10.1.1.242	http	16	7 173	8	848	8	
10.5.2.44	60074	10.1.1.242	http	10	2 949	5	883	5	
10.5.2.44	60075	10.1.1.242	http	41	27 999	21	1 813	20	
10.5.2.44	60076	10.1.1.242	http	22	10 945	11	1 184	11	
10.5.2.44	60077	10.1.1.242	http	30	19 719	15	1 443	15	
10.5.2.44	60078	10.1.1.242	http	64	48 922	30	2 417	34	
10.5.2.44	60079	10.1.1.242	http	26	12 925	13	1 293	13	
10.5.2.44	60080	10.1.1.242	http	10	1 707	5	757	5	
10.5.2.44	60081	10.1.1.242	http	11	1 775	6	825	5	
10.5.2.44	60082	10.1.1.242	http	10	1 610	5	757	5	
10.5.2.44	60083	10.1.1.242	http	10	2 467	5	787	5	

**21 TCP Connections (one HTTP GET
requests/TCP connection) to display a page
→ Non-persistent connections**

HTTP Request for displaying <http://google.co.in>



GET / HTTP/1.1

Host: www.google.co.in

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:14.0) Gecko/20100101 Firefox/14.0.1

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: en-us,en;q=0.5

Accept-Encoding: gzip, deflate

Connection: keep-alive

HTTP/1.1 200 OK
Date: Sat, 18 Aug 2012 00:26:02 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=UTF-8
Set-Cookie:

HTTP Response



PREF=ID=a8abc9c3f02ef81d:FF=0:TM=1345249562:LM=1345249562:S=DXaQJpF6arjODGNZ;
expires=Mon, 18-Aug-2014 00:26:02 GMT; path=/; domain=.google.co.in
Set-Cookie: NID=63=IPldVN29RfiV-
JjVmAEM7MrXqB7qsFbkKgT5F6ZrfkwjtC7FWzTzY46JLGikKNMICN1Em4QIQELbf3Uj6JQtqFWdHva
nS4mgtop1QqBEVvIXabltWVtylGgY0YhgRtftp; expires=Sun, 17-Feb-2013 00:26:02 GMT; path=/;
domain=.google.co.in; HttpOnly
P3P: CP="This is not a P3P policy! See
<http://www.google.com/support/accounts/bin/answer.py?hl=en&answer=151657> for more
info."
Content-Encoding: gzip
Server: gws
Content-Length: 27185
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN

[[DATA]]

Second Request



GET /images/icons/product/chrome-48.png HTTP/1.1
Host: www.google.co.in
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:14.0) Gecko/20100101 Firefox/14.0.1
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://www.google.co.in/
Cookie:
PREF=ID=a8abc9c3f02ef81d:FF=0:TM=1345249562:LM=1345249562:S=DXaQJpF6arjODGNZ;
NID=63=IPldVN29RfiV-
JjVmAE7MrXqB7qsFbkKgT5F6ZrfkwjtC7FWzTzY46JLGikKNMlCN1Em4QIQELbf3Uj6JQtqFWdHva
nS4mgtop1QqBEVvIXabltWVtylGgY0YhgRtfp

Second Response



HTTP/1.1 200 OK
Content-Type: image/png
Last-Modified: Mon, 02 Apr 2012 02:13:37 GMT
Date: Sat, 18 Aug 2012 00:26:03 GMT
Expires: Sat, 18 Aug 2012 00:26:03 GMT
Cache-Control: private, max-age=31536000
X-Content-Type-Options: nosniff
Server: sffe
Content-Length: 1834
X-XSS-Protection: 1; mode=block

[[DATA]]

The scenario for <http://www.avaxhome.ws>



Endpoints: eth0 (port 80)

Ethernet: 3 Fibre Channel FDDI **IPv4: 24** IPv6 IPX JXTA NCP RSVP SCTP TCP: 68 Token Ring UDP USB WLAN

IPv4 Endpoints

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Latitude	Longitude
10.5.2.44	1 629	1 051 238	771	73 697	858	977 541	-	-
95.215.0.151	166	96 828	82	88 012	84	8 816	-	-
174.120.252.45	42	10 985	21	7 119	21	3 866	-	-
173.194.36.46	9	2 426	4	1 780	5	646	-	-
46.4.68.55	745	576 782	416	553 639	329	23 143	-	-
23.57.207.139	81	63 696	45	61 446	36	2 250	-	-
124.124.252.41	72	45 873	35	42 866	37	3 007	-	-
66.220.149.94	13	7 010	6	5 987	7	1 023	-	-
173.194.36.34	33	18 257	16	16 328	17	1 929	-	-
202.43.205.15	26	5 028	13	3 500	13	1 528	-	-
202.43.205.16	76	21 022	38	13 680	38	7 342	-	-
209.85.175.95	43	27 760	21	26 149	22	1 611	-	-
68.67.179.202	14	4 100	7	2 339	7	1 761	-	-
49.248.249.17	121	82 067	63	76 972	58	5 095	-	-
202.43.206.39	20	3 825	10	2 186	10	1 639	-	-
68.67.159.201	10	1 842	5	996	5	846	-	-
68.67.130.95	11	2 757	5	1 355	6	1 402	-	-
165.254.127.234	94	63 419	46	59 944	48	3 475	-	-
81.218.31.168	21	12 188	10	10 592	11	1 596	-	-
184.173.146.42	13	3 235	7	1 943	6	1 292	-	-
204.93.223.146	7	1 460	3	408	4	1 052	-	-
202.137.238.20	3	174	2	120	1	54	-	-
195.59.58.219	3	168	1	60	2	108	-	-
165.254.40.41	6	336	2	120	4	216	-	-

☒ Name resolution

☐ Limit to display filter

Help

Copy

Map

Close



Presentation Overview

Latest Twist

Examples

Cookies

HTTP/1.1

HTTP Overview

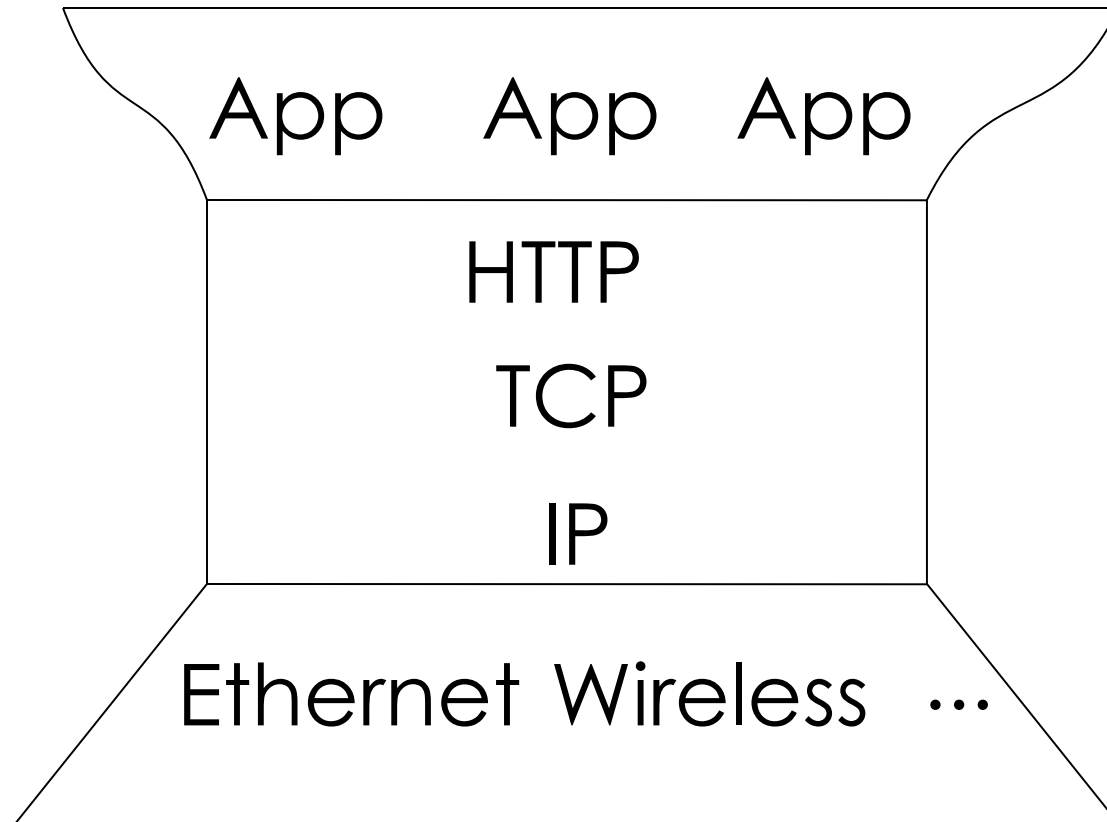


Latest Twist

- HTTP Use Cases
 - WWW
 - Web Services (SOAP, REST)
 - Live Streaming (Ex: Youtube)
 - Transport protocol for InterPlanetary Network (IPN)
- Google SPDY (SPeeDY)
 - Reference point for HTTP/2.0



Today's Hourglass?



Courtesy: Steve M. Bellovin (<http://www.research.att.com/~smb>)
Protocol Problems and Architectural Issues, 10/03/2003