Computer Networks and Distributed Systems

Computer Networks – Application Layer

Course 527 – Spring Term 2014-2015

Anandha Gopalan

a.gopalan@imperial.ac.uk
http://www.doc.ic.ac.uk/~axgopala

Contents

- Application Layer
  - Application level protocols
    - Domain name resolution (DNS)
    - Email (SMTP, POP, IMAP)
    - World Wide Web (HTTP)

Client

- Initiates connection
- Often user-invoked app running on local machine
  - e.g. web browser, email clients
- Uses service/resource from server
- Resource use usually temporary

Server

- Waits for connections
  - Handles multiple clients
- Special purpose app providing service/resource
  - e.g. web server, chat server
- Provides controlled access to resources/services
- Often started at boot time
  - e.g. “daemon” in UNIX; “service” under WIN

Domain Name System (DNS)

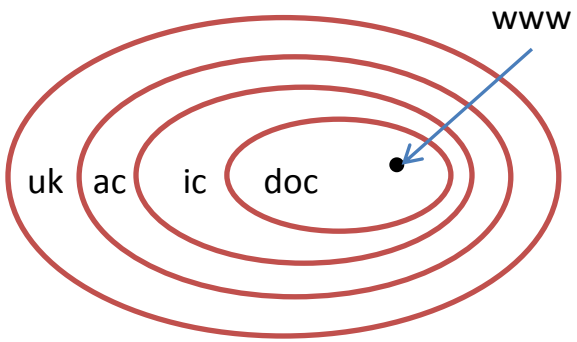
- An end system is identified and addressed by its IP address
  - Advantage → Computers (e.g., routers) are good at processing bits – especially in small packs of a size that is a power of two
  - Disadvantage → Not practical for use by people i.e., not mnemonic, e.g. “look it up on 64.233.183.104”

- Goal: help the human users of the Internet
  - Human-readable, mnemonic addresses, aliases
- Solution (up to 1982): all mapping were contained in the file `hosts`
- Solution (from 1982 onward): domain name system (DNS)
  - Distributed lookup facility
- Primary function of the domain name system
  - Map a name to an IP Address, e.g. www.doc.ic.ac.uk → 146.169.13.59

Domain Name System (DNS)

- Internet is inter-network of autonomous networks
  - Must avoid conflicts between names
  - Must support independent administration of names

- DNS names form hierarchies
  - e.g. `www.doc.ic.ac.uk`



- Requires uniqueness of complete name only
  - Like postal address
  - e.g. `fred.foo.com` different to `fred.bar.com`

Names and Domains: Externally Managed

- Top-level structure conveys meaning
  - `uk` = United Kingdom
    - Country ID from global naming standard
  - `ac` = Academic network
    - Standard sub-domain within UK
  - `ic` = Imperial College
    - Name assigned by UK academic net administration
    - Owned by Imperial College

- Within each domain, naming managed independently
  - e.g. `co.uk` is independent of `co.fr`

Top Level Domains (TLDs)

Domain	Specification	Example
`com`	Commercial	`google.com`
`net`	Network providers	`internic.net`
`edu`	Educational institution	`mit.edu`
`gov`	Government organisation	`whitehouse.gov`
`mil`	US Military organisation	`navy.mil`
`org`	Other organisation	`linux.org`
<country code>	Domain administered by country	`fr, uk, in, ch, de`
2nd level country domains	Sub-domains to TLD administered by organisation for that country	`ac.uk, co.uk`

13 root DNS servers (labeled A – M) know location of top-level servers

http://www.root-servers.org

Names and Domains: Internally Managed

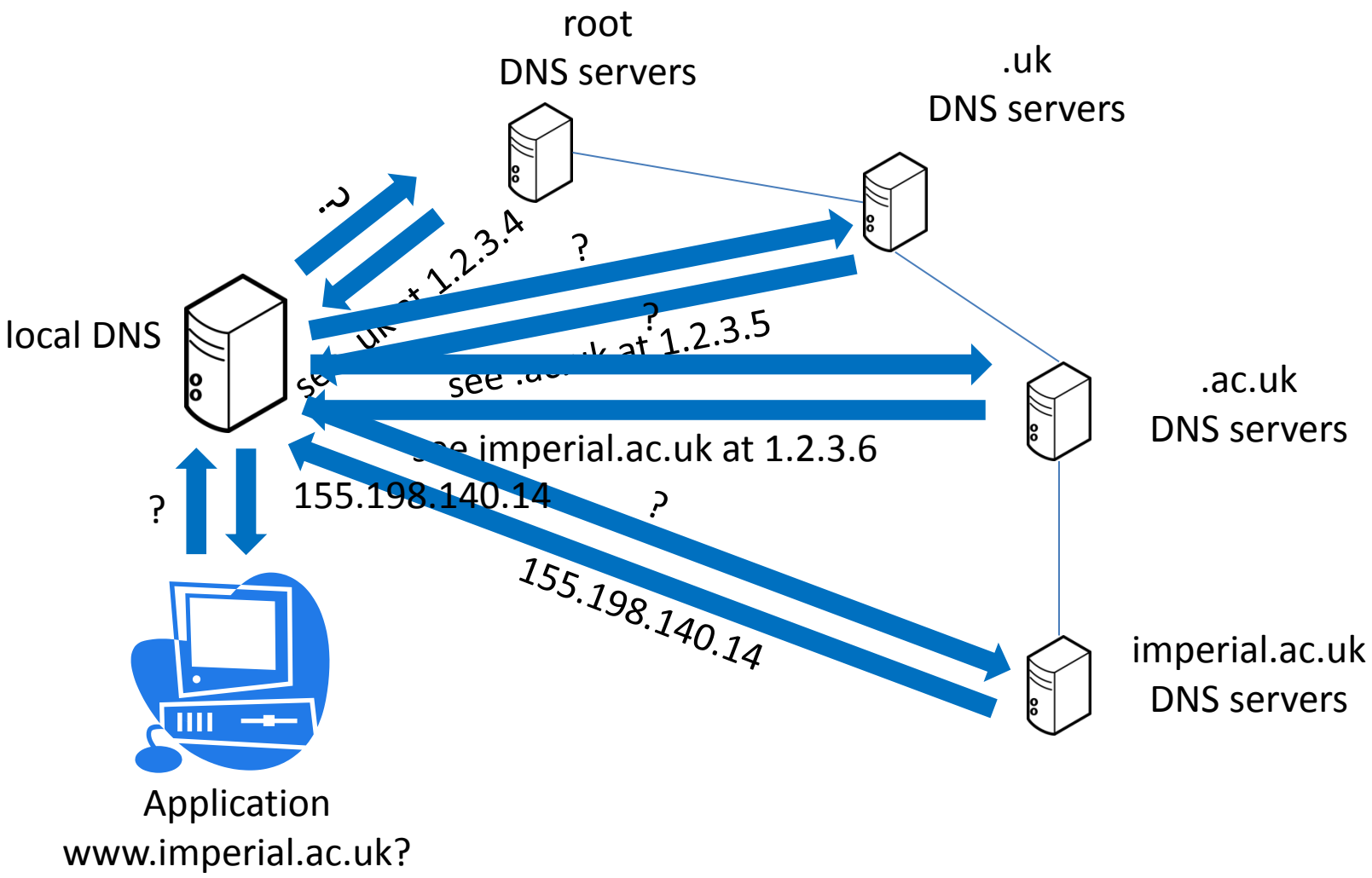
- Local domains follow organisational structure
  - `doc` = Dept. of Computing
    - Assigned by College admin
  - `www` = Machine in DoC
    - Assigned by Dept admin

- Type of name may reflect type of machine
  - e.g. in DoC birds are servers, colours are printers
- Names may reflect services machine provides
  - e.g. `www, mail, cate, …`

DNS Name Resolution

- Host knows local DNS server to ask for names
  - Local database of names maintained manually
- Authoritative results
  - Returned from name server managing that name
- Local DNS servers do not move often
  - May be hard-coded or given by DHCP
  - DNS server known by its IP address
  - Domain will typically have 2-3 DNS servers

Example DNS lookup



DNS Caching

- A lot of messages just to figure out where to connect to
  - DNS can indeed be a major bottleneck for some applications (typically, the Web)
  - It is also to a large extent a critical point of failure
- Servers cache responses as names often needed again
  - Exploits locality of reference
- Cached answers non-authoritative → may be wrong, out-of-date
- DNS entries give TTL based on volatility
  - Caches expire records after their TTL has expired
  - Stable names can safely be cached for much longer

Electronic Mail

- Features and Goals
  - Asynchronous communication
    - Alice sends a message when it is convenient to her
    - Bob reads Alice's message whenever he has time to do that
  - One-to-many communication
    - Alice can send a message to Bob and Charlie
    - A mailing list sends messages to several receivers
  - Multi-media content
    - Images and all sorts of attachments as well as normal text

Electronic Mail

- Limitations
  - No authentication
    - Messages can be modified
    - Messages can be forged
  - No confidentiality
    - The message can be read by others
  - Little or no delivery guarantees
    - Messages can be accidentally lost or intentionally blocked
    - No reliable acknowledgment system

Electronic Mail

- Movement of structured text msg between systems
- Email message fields

  To:, Cc:, Sender:, Reply-to: Email addresses
  From: Who sent the message
  Received: List of mail servers that processed mail
  Subject: Short summary of message
  Message-Id: Unique id for message

  - Detect duplicate messages, responses

- Email address: <user name>@<mail domain name>
  - e.g. axgopala@imperial.ac.uk
  - Mail server for domain found through DNS MX records

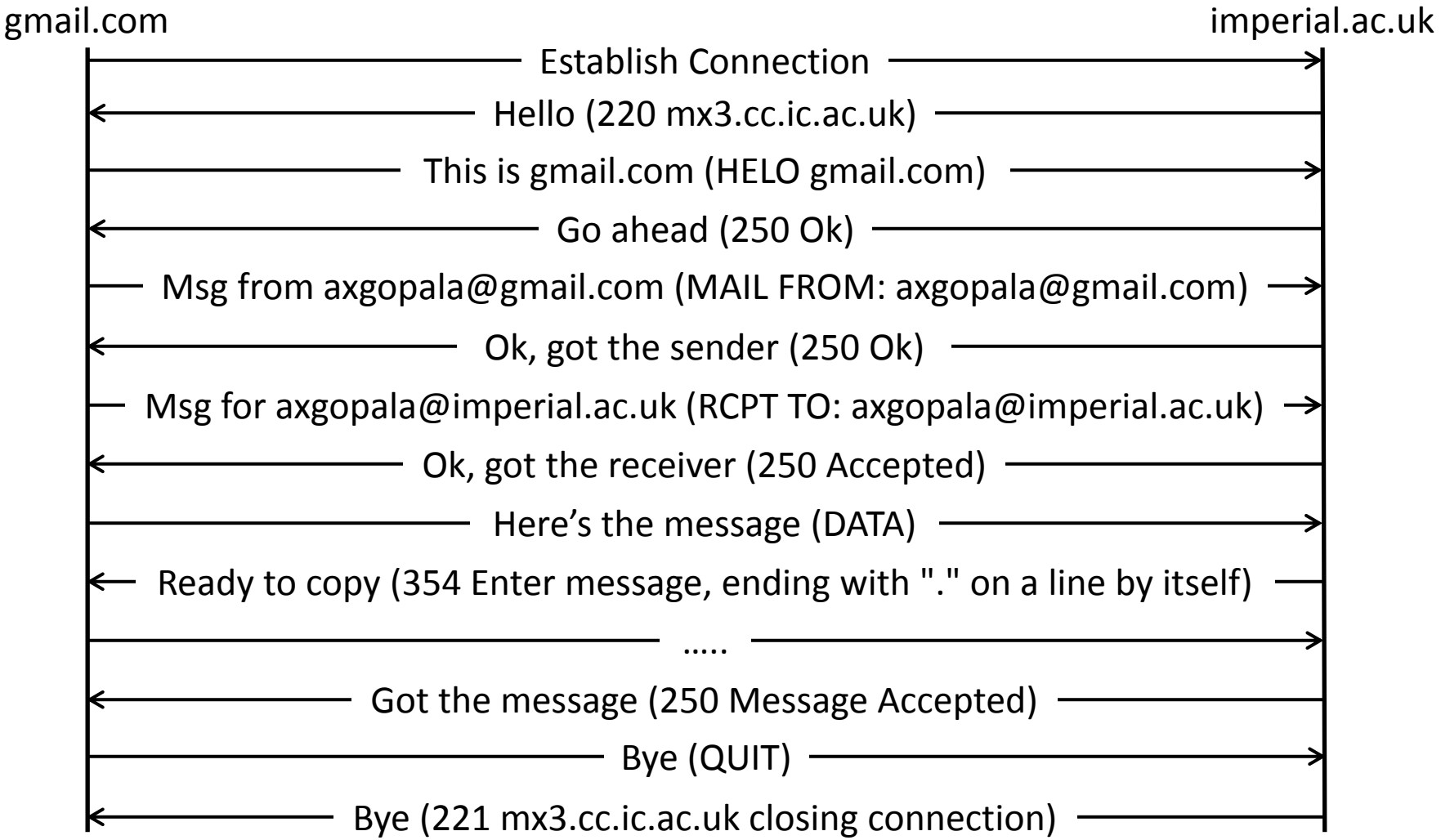
Email Subsystems

- Mail User Agents (MUA)
  - e.g. Thunderbird, Outlook, mutt, pine, web-based, …
  - Email client for composition, display, filing, …
- Message Transfer Agents (MTA)
  - e.g. sendmail, exim, Exchange, …
  - Mail server that handles message transfer
- Email often sent from MUA to relay (or gateway) MTA
  - Usually in same organisation as sender
  - Delivers to MTA in recipient's domain
    - Possibly via other relays
  - Holds (spools) mail if destination unreachable
    - Supposed never to lose mail

Simple Mail Transfer Protocol (SMTP)

- Protocol used by MTAs (and MUAs)
  - Delivers mail from one system to another
- Two parties communicate using TCP and port 25
  - Sender (client) & Receiver (server)
- Three basic steps
  1. Start session
  2. Exchange data
  3. Complete session

SMTP: Protocol Steps



Multipurpose Internet Mail Extensions (MIME)

- The standard message format has some serious limitations
  - 7-bit (text) content
  - Only text
  - Essentially good exclusively for the English language

- Any content that does not fit the 7-bit (ASCII) character set must be encoded

- The Multipurpose Internet Mail Extensions (MIME) specification
  - (RFC 2045 and RFC 2046) defines useful extensions
- Valid types include
  - text/plain — this is a normal ASCII message
  - text/html — this is an HTML-formatted message
  - image/jpeg — this message contains (only) an image file
  - multipart/mixed — this message consists of multiple parts

Post Office Protocol (POP)

- SMTP designed for permanently available hosts
  - e.g. Internet mail servers
  - SMTP delivers mail to mailbox at ISP
  - Not usually used to deliver mail to user's desktop

- Post Office Protocol allows user access to mailbox
  - POP client (MUA) connects to mailbox (POP) server
  - Connection to port 110 when "get mail" is requested
  - POP authenticates user (with password)
  - Mailbox downloaded for processing

Internet Message Access Protocol (IMAP)

- POP only supports downloads of Inbox
  - Implicitly assumes that retrieved mail is deleted at the server
- IMAP handles multiple folders
  - Possible to leave emails on mail server
  - Useful when using multiple MUAs or machines
- Other features
  - Partial downloads of emails/attachments
  - Offline mode when unconnected
  - Server-side searches
- But IMAP complex and adds server load

POP vs. IMAP

Feature	POP	IMAP
Where is e-mail stored	User's PC	Server
Where is e-mail read	Off-line	On-line
Connect time required	Little	Much
User of server resources	Minimal	Extensive
Multiple mailboxes	No	Yes
Who backs up mailboxes	User	ISP
Partial message downloads	No	Yes
Are disk quotas an issue?	No	Could be in time
Simple to implement	Yes	No

World Wide Web (WWW)

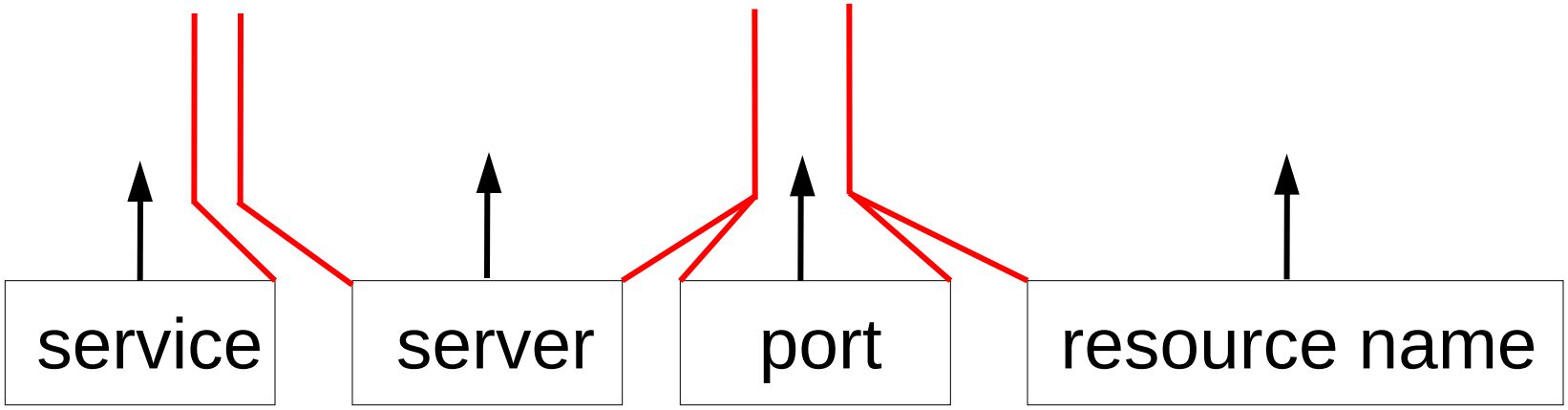
- Developed by Tim Berners-Lee (CERN in 1989)
- Killer application for the Internet
  - Supports transfer and display of documents
  - Documents include multimedia content and hyperlinks
- Client/server based
  - Client (browser): Firefox, Chrome, Chromium, Explorer, Opera, lynx, …
  - Web servers: Apache, IIS, …

Web Standards

- World Wide Web Consortium (W3C) manages standards
- HyperText Transfer Protocol (HTTP)
  - Used by browsers to get resources from servers
- HyperText Mark-up Language (HTML)
  - Encoding of data for web pages
  - Supports text with images, formatting and hyperlinks
- Uniform Resource Locator (URL)
  - Used to identify links to resources on servers
- Uniform Resource Identifier (URI)
  - More general, newer version of URLs
  - URIs can also be location independent (pure names)

Uniform Resource Locators (URLs)

`http://www.doc.ic.ac.uk:80/~axgopala/research.html`



- URLs can indicate
  - other protocols, e.g. ftp, gopher, telnet, mail, news, file, …
  - other resource types, e.g. image, program, service …
- URLs encode location of resource

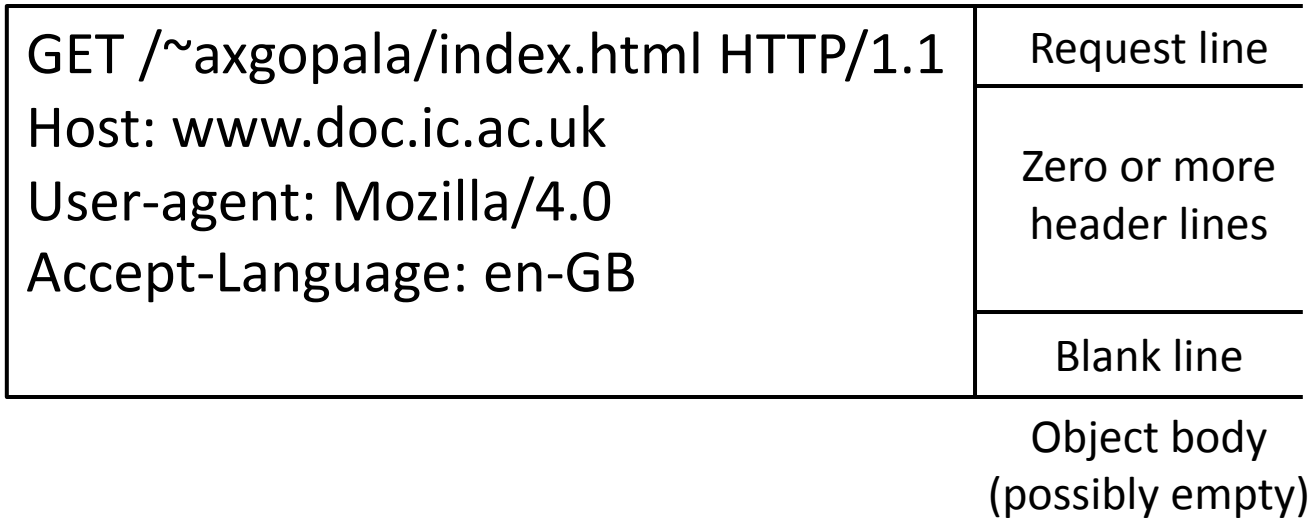
Hypertext Transfer Protocol (HTTP)

- HTTP server usually listens at TCP port 80
- Stateless, transaction-oriented protocol
  1. Client opens connection to server
  2. Request sent from client to server
  3. Server responds
  4. Connection closed

HTTP Request Message

- Request format
  - Request line (method, identifier, version)
  - Header (additional info)
  - Body (data)

- Anatomy of a request

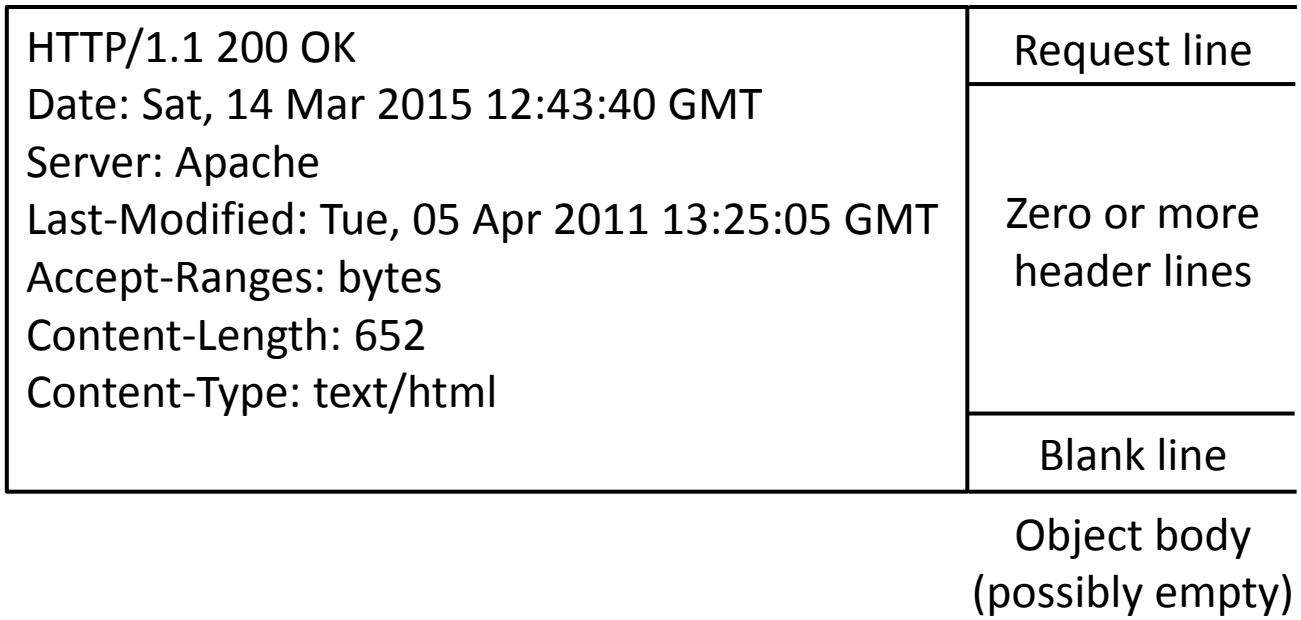


HTTP Methods

- GET: Client requests resource from server
  - No permanent action on server is implied
  - Most common method
- HEAD: Requests only header of web page
  - Useful when deciding if changed and to test validity of links
- POST: Append/send data to named resource
  - Used to submit client data from web forms
- Others: PUT, DELETE, LINK, UNLINK

HTTP Reply Messages

- Reply format
  Status line (version, code, optional message)
  Header (additional info)
  Body (data, MIME compatible)
- Anatomy of a response



HTTP Reply Codes

1xx	Informational
2xx	Successful operation OK (200), Accepted (202), No response (204)
3xx	Redirection Moved permanently (301), Not modified (304)
4xx	Client error Bad request (e.g. syntax error) (400) Unauthorised (401), Payment required (402) Not found (404)
5xx	Server error Service unavailable (503), HTTP version unsupported (505)

Stateless and Non-Persistent

- HTTP protocol is stateless
  - Server doesn't keep track of client requests
    - Simplifies server design
  - But websites would like to keep track of customers

- HTTP/1.0 uses non-persistent connections
  - New TCP connection opened for every request
    - Page with 10 images will open 11 TCP connections
  - Adds server load and setup costs and delays
  - TCP congestion control inefficient for short transfers

Maintaining State: Cookies

- Websites want to identify users
  - IP addrs not practical ids for users due to NAT, sharing
- Persistent information through cookies
  - Allows server to maintain state between HTTP requests
  - Sent by web server and stored by browser
    - Name/value pair with expiry time
    - Set-Cookie header in HTTP response

Persistent Connections: HTTP/1.1

- Deals with some of HTTP/1.0's performance issues
  - Persistent connections
    - Client opens TCP connection
    - HTTP requests pipelined through this connection
    - Multiple requests with one TCP open/close overhead
  - Other features
    - Better proxy handling; more methods; improved encoding and authentication
- Now in widespread use
  - Easy transition because backwards compatible