

Week 11 - Tuesday

---

## HTTP

---

School of Information Technology and Electrical Engineering  
The University of Queensland

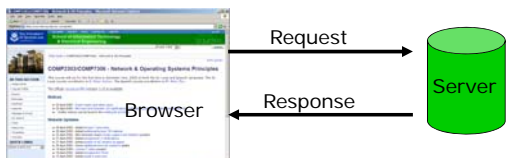
## HTTP

- **HTTP = HyperText Transfer Protocol**
  - Not the same thing as HTML
  - Can transport other things
  - ASCII based protocol (not the body)
    - i.e. lines of text – each terminated by carriage-return newline (\r\n)
      - Most ASCII based network protocols use \r\n line endings
  - Usually on top of TCP, but standard does not require this

2

## How does HTTP work?

- HTTP has two phases
  - Request - browser sends request
  - Response - server sends response



The diagram shows a browser window on the left and a server cylinder on the right. An arrow labeled 'Request' points from the browser to the server. A return arrow labeled 'Response' points from the server back to the browser.

- ASCII based protocol
  - Can simulate this with telnet connection

3

## HTTP Requests and Responses

- Requests and Responses both have three parts
  1. **request or response line**
  2. **header section**
  3. **body**
- HTTP Request:
  - Client (browser) first connects to the server
    - Usually on TCP port 80

4

## HTTP Request

{ **HTTP Request Line**  
Header Section  
Body

- After connection, client sends a request line, consisting of
  - method
  - document address
  - HTTP version no.
- Example:
  - `GET /~comp2303/index.html HTTP/1.0`

⏟
⏟
⏟  
 Method      Document Address      HTTP Version

5

## Example

- telnet www.uq.edu.au 80  
GET / HTTP/1.0  
(followed by two newlines)
- echo -e "GET /~comp2303/ HTTP/1.0\r\n\r\n" | nc www.itee.uq.edu.au 80  
(The -e interprets escaped chars)

6

COMP2303  
COMP7306

## HTTP Request Methods:

- **GET** - tells server to retrieve entire document
- **HEAD** - requests just the header information for the response (not HTML document <HEAD>)
- **POST** - for posting data (e.g. from forms)
  - Contents of form encoded and sent to server
- **PUT** - for replacing document with data sent from the client
- **DELETE** - for removing a document from server
  - **PUT** and **DELETE** used for direct-to-web publication
- **LINK, UNLINK, OPTIONS, TRACE, ...**

7

COMP2303  
COMP7306

## HTTP Request

HTTP Request Line  
Header Section  
Body

- After request line, client can send any number of header lines
  - mostly informational
  - usually entirely optional
- Example:
 

```
Connection: Keep-Alive
Accept: image/gif, image/jpeg, */*
Accept-Encoding: gzip
Accept-Language: en
User-Agent: Mozilla/4.5 [en] (Win98; I)
```

8

COMP2303  
COMP7306

## HTTP Request Headers

- **User-Agent** - name and version of client
- **Referer** - URL of last document displayed
- **Authorization** - client's authorization to access the data (e.g. encoded name & password)
- **If-Modified-Since** - return document, only if modified since given date
- **Content-Length** - Number of data bytes
  - Mandatory for **PUT** and **POST** requests
- **Connection** - Connection options (e.g. **Keep-Alive**)
- **Host** - Virtual host to retrieve data from
- **Cookie** - Cookie(s) for that URL
- plus many others ...

9

COMP2303  
COMP7306

## HTTP Request

HTTP Request Line  
Header Section  
Body

- Body (or Request Data)
  - Follows the header section (and a blank line)
  - Only needed for **PUT** and **POST** requests
  - Not needed for **GET, HEAD, DELETE** requests

10

COMP2303  
COMP7306

## HTTP Response

HTTP Response Line  
Header Section  
Body

- First line of response consists of
  - Protocol version number
  - Status code
  - Explanation of status
- Example:
 

```
HTTP/1.1 200 OK
HTTP/1.1 301 Moved Permanently
```

Protocol Version      Status      Explanation

11

COMP2303  
COMP7306

## Common Status Codes

- **100's** - Informational
- **200's** - Client request successful
  - 200 - OK - URL found, contents follow
  - 204 - No Response - Request OK but no data
- **300's** - Redirecton, further action needed
  - 301 - Moved - URL permanently moved
  - 304 - Not Modified - Possible response to "If-Modified-Since"
- **400's** - Client request incomplete
  - 401 - Unauthorised - user must produce authorisation
  - 403 - Forbidden - authorisation failed
  - 404 - Not Found - document does not exist
- **500's** - Server errors

12

COMP2303 / COMP7306

## HTTP Response

HTTP Response Line  
Header Section  
Body

- Header lines consist of
  - Information about server
  - Information about document that follows
  - Mostly optional - except for **Content-Type**
- Example:
 

```
Date: Sat, 09 Oct 1999 11:32:15 GMT
Server: Apache/1.3.6 (Unix)
Last-Modified: Sat, 09 Oct 1999 11:30:06 GMT
Content-Length: 111
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html
```

13

COMP2303 / COMP7306

## Common Response Headers

- Server** - name and version of the server software
- Date** - the current date
- Last-Modified** - date document was last changed
- Location** - new location for redirection responses
- Content-Length** - No. of bytes of data
- Content-Type** - MIME type of data
- Set-Cookie** - contains cookie info
- Lots of others ...

14

COMP2303 / COMP7306

## HTTP Response

HTTP Response Line  
Header Section  
Body

- Body (or Response Data)
  - Follows the header section (and a blank line)
  - Not needed for **HEAD** request
  - Can be 0 bytes to billions...
- Connection
  - Usually server will close connection after data finished
    - If **Connection: Keep-Alive** has been specified then channel will remain open waiting for another request

15

COMP2303 / COMP7306

## HTTP - Summary

- Request
  - Method Document-Address HTTP-version
  - Request Headers
  - Body (Request Data)
- Response
  - HTTP-version Status Status-Description
  - Response Headers
  - Response Data

16

COMP2303 / COMP7306  
and 08 P.11.11.11.11

Week 11 - Tuesday

## Network Applications

School of Information Technology and Electrical Engineering  
The University of Queensland

COMP2303 / COMP7306

## Outline

- Today
  - Network Applications - DNS, Email, Network Utilities
- Credits:
  - Tanenbaum, "Computer Networks"

18

COMP2303  
COMP7306

## DNS – Domain Name System

- Strings are used to name hosts
  - e.g. agave.students.itee.uq.edu.au
- Network only understands numbers
  - Need conversion mechanism
- Original ARPANET
  - Hosts file – listed all hosts and IP addresses
    - Doesn't scale
    - Need to decentralize

19

COMP2303  
COMP7306

## DNS Hierarchy

- Subdivide and delegate authority
- Repeat – end up with tree like structure
- Internet – hierarchy based on structure of organisations – not on physical network connections

20

COMP2303  
COMP7306

## DNS Hierarchy (cont.)

- There are additional top level domains also: .biz, .info, ...

21

COMP2303  
COMP7306

## Mapping Domain Names to Addresses

- Name server
  - Program that supplies name-to-address translation service
- Client – *name-resolver*
  - Uses one or more servers
- Resolution
  - Start with local server
  - Query passed on if answer unknown
  - Answer cached locally for some time

22

COMP2303  
COMP7306

## DNS Query

- DNS Queries – based on UDP
- Figures to be drawn
  - Recursive
  - Non-recursive
- nslookup demonstration

23

COMP2303  
COMP7306

## Email

- Email systems have two parts
  - MTAs – Message Transfer Agents
    - Daemons, e.g. sendmail
  - UAs – User Agents
    - Mail readers

24

COMP2303  
COMP7306

## User Agent

- aka Mail reader
- Program used by end-user
- Supports
  - composing
  - retrieving
  - replying
  - mailbox manipulation

25

COMP2303  
COMP7306

## Sending Email

- Envelope
  - Encapsulates the message
  - That information needed to transport message, e.g.
    - Destination address, priority
- Message
  - Header
    - Control information for mail readers
  - Body
    - Information for human recipient

26

COMP2303  
COMP7306

## RFC 822

- Internet standard for ASCII mail transmission
- *Doesn't clearly distinguish envelope from header fields*
- Messages consist of
  - header lines of form  
Field-name: value
  - blank line
    - indicates end of headers
  - body
- MTA's use envelope and/or header fields to transport messages

27

COMP2303  
COMP7306

## Message Transport Header Fields

Headers relevant to message transport:

- To:
- Cc:
- Bcc:
- From:
  - Creator or author
- Sender:
  - Who actually sent it (may not be author)
- Received:
  - Each MTA adds a line along the way
- Return-Path:

28

COMP2303  
COMP7306

## Other Headers

- Most common:
  - Date:
  - Reply-To:
  - Message-Id:
  - In-Reply-To:
  - Subject:
- Any can add X- headers
  - e.g., X-This-can-be-any-name: value

29

COMP2303  
COMP7306

## Message Body

- ASCII
  - 7-bit
- Other data forms need to be encoded
- MIME
  - Multipurpose Internet Mail Extensions
  - Allow existing mail protocols and programs to be used

30

COMP2303  
COMP7306

## MIME

- Five relevant headers:
  - MIME-Version:
  - Content-Description:
    - Human-readable string
  - Content-Id:
    - Unique Identifier
  - Content-Transfer-Encoding:
    - How message is packaged
  - Content-Type:
    - Nature of the message, e.g.
      - text/plain, video/mpeg, image/gif
      - application/octet-stream (this is anything)
      - multipart/mixed, multipart/alternative

MIME is used for HTTP also

31

COMP2303  
COMP7306

## Transfer Encoding

- Five schemes
  - ASCII text (7 bit characters), lines < 1000 characters long
  - 8 bit characters, line length < 1000
    - violates original protocols, but mostly works these days
  - Binary (8-bit characters)
    - No guarantee that messages will arrive correctly
  - Base64 encoding
  - Quoted-printable encoding

32

COMP2303  
COMP7306

## Base64 encoding

- Also called "ASCII armour"
- Groups of 24 bits (3 bytes) broken into four 6-bit units
- Each 6-bit unit encoded as ASCII character
  - 0-25 encoded as A-Z
  - 26-51 = a-z
  - 52-61 = 0-9
  - 62 = +
  - 63 = /

33

COMP2303  
COMP7306

## Base64 Example

34

COMP2303  
COMP7306

## Quoted-printable Encoding

- For messages that are mostly ASCII
  - Base 64 inefficient
- Encodes
  - characters above 127 (7-bit limit)
  - non-printable ASCII charactersas
  - equals sign, followed by
  - two hexadecimal digits representing character code

35

COMP2303  
COMP7306

## Message Transfer

- SMTP
  - Simple Mail Transfer Protocol
  - Standard port: 25
  - Simple ASCII protocol
  - Commands:
    - HELO, MAIL FROM:, RCPT TO:, DATA:
- Most common implementation
  - Sendmail

36

COMP2303  
COMP7306

## Final Delivery

- POP3 = Post Office Protocol
  - Allows user to log-in, fetch messages, delete messages, log-out
  - ASCII text based
- IMAP = Internet Message Access Protocol
  - Central repository of email accessed from anywhere

37

COMP2303  
COMP7306

## Network Utilities

Selection to be discussed/demonstrated in class

- ifconfig
- ping
- traceroute
- netstat
- arp
- whois
- nslookup
- tcpdump

38