COMP519
Web Programming
Autumn 2015

Internet and World Wide Web Protocols
The purpose of these notes is to give a brief overview of how devices connected to the Internet and World Wide Web communicate with each other, and information that we, as web programmers, need to know in order to perform certain programming tasks.
In the 1970s computer networks were ad hoc, following vendor specified designs and communication methods.

### Open Systems Interconnection model

- **Developed**
  by the ISO (International Organization for Standardization) in 1984.

- **Provides**
  an abstract model of networking.

- **Divides** the tasks involved in moving information between networked computers into seven task groups.

- **Each task group is assigned a “layer”**.

Each layer is reasonably self-contained, so (in theory)

- each can be implemented independently, and
- changes/updates to a layer need not effect other layers.
The goal of defining the Open System Interconnection Reference Model (the OSI Reference Model) was to ensure the interoperability of communication systems by using standard protocols.

A layer serves the layer above it, and is served by the layer below it.
Protocol Layers

Application layer
Describes how applications will communicate, e.g. HTTP, FTP, Telnet, SMTP.

Presentation layer
Describes the form of data being transferred and ensures that it will be readable by receiver, e.g. floating point formats, data compression, encryption.

Session layer
Describes the organization of large data sequences and manages communication session, e.g. coordinates requests/responses (“traffic flow”).

Transport layer
Describes the quality and nature of data delivery, e.g. how retransmissions are used to ensure delivery.

Network layer
Describes how a series of exchanges over various data links can deliver data across a network, e.g. addressing and routing.

Data Link layer
Describes the logical organization of data bits transmitted on a particular medium, e.g. frame sequencing, error notification.

Physical layer
Describes the physical and electrical properties of the communications media, e.g. voltage levels, data rates, max distances.
What is a protocol, after all?

Protocols define how messages are sent and received, i.e. the format and order of them, what happens when messages are transmitted and received.

Efficient protocols are what allows the Internet to operate in the decentralized manner that they do.

Packet routing using these protocols also allows for reasonably good error checking/correcting methods, and reliability of transmissions.
Layer Protocols

Across a network, processes at the same level can (seemingly) communicate, e.g. Web servers and browsers run at the application level, communicating via HTTP.

In reality, actual communication takes place at the physical layer. Upper layers can only communicate with those above and below.

At the source, as data is passed down the layers, the protocol for each layer adds control information to the data.

At the destination, as data is passed up the layers, the protocol for each layer strips and analyzes the control information for that layer.
The Internet Protocol Suite is a set of communications protocols used for information transfer on the Internet (and other networks).

Network layer: Internet Protocol (IP)
- Provides generalized packet network interface.
- Handles routing through the Internet.
- Connectionless and unreliable. (In contrast, the telephone system is a “connection” service.)

Transport layer: Transmission Control Protocol (TCP)
- Provides a virtual circuit over which two processes can communicate.
- Supplies logic to give reliable, connection-oriented session.
- FTP (file transfer) and HTTP are built on top of TCP.
The Internet Protocol (IP) divides information into packets (datagrams) for delivery from a source to a destination. This protocol defines the format of the packets and provides the “addressing system” for routing packets.

IP adds packet routing information, which includes the source address (IP address of the host sending the packet), a destination address (IP address of host to receive the packet), a “time to live” (an amount of time to remain before the packet can be discarded as undeliverable), etc.
IP is characterized as “unreliable” as it provides no guarantee of delivery or quality of service for transmitting information from source to destination. It is also a “connectionless protocol” in that each packet is individually addressed and routed based on the information in the packet itself (rather than using a fixed data channel).

In a loose sense, IP is designed to facilitate rapid communications by using packet routing to divide data into small portions and route them along (possibly different) paths to their destination.
The Transmission Control Protocol (TCP) adds information (on top of that the IP uses) that provides a virtual circuit, including message formatting, circuit management, flow control, and error correction.

TCP includes information such as source and destination ports, a sequence number (identifying its place in a sequence of packets), an acknowledgment number (specifying the next sequence number the receiver is expecting), a checksum (used for error checking of the header and data), etc.
TCP works by establishing a connection in a “handshake process” before beginning data transfer. Once the transmission is complete, the connection is terminated to release resources to be used again for other transmissions.

Utilizing IP’s packet routing technology, TCP provides the methodology to ensure reliable transmission.
The Internet Protocol Suite

As stated, IP provides the method of distributing data in packets, and can distribute packets to a destination via different routes, handling congestion in this manner.

TCP provides for virtual circuits, error detection and correction, automatic repeat requests, and other reliable end-to-end communications, using the IP protocol for transmission (TCP is designed for accurate, not necessarily timely, communications).

TCP also handles reassembling packets at the destination, as they can arrive out of order due to the IP routing scheme to avoid congestion.

TCP/IP together comprise the Internet Protocol Suite, and provide for the most commonly used functions on the Internet, namely e-mail, file transfer, and are the basis of the HTTP protocol for web page documents.
Data Encapsulation and Transmission

message
segment
Datagram
Frame
IP addresses

IP addresses are numerical labels assigned to computers in a network, used for identification and addressing (passing information).

IP addresses under IP version 4 are 32 bits long.

10010011 10000110 00000010 00010100
written as a dotted sequence
147.134.2.20

IPv6 extends address sizes to 128 bits.

Its extensions support authentication, data integrity, and confidentiality (as well as providing the many more IP addresses necessary to accommodate the explosion of smartphones, tablets, laptop computers, etc).
Domain Name System

Rarely do applications deal directly with IP addresses.

A hierarchical system of domain names can be used instead, e.g. top level domains: edu, com, gov, org, net, . . .

Commonly: hostname.subdomain.domain (possibly many subdomains), e.g. csc.liv.ac.uk

Domain names serve as “listings” in a “phone book” for the internet, i.e. when you enter a domain name in a web browser, a “look up” is done to find the corresponding IP address.

Domain names also simplify things in that if the physical network changes (e.g. IP addresses are modified), the domain names can remain the same while the “phone book” entries are updated with the new IP address.
Domain Name Servers

A domain name server is a machine that keeps a table of names and corresponding IP addresses (i.e. this performs the “look up” to match domain names to their corresponding IP address).

As of Oct 2014, there are 504 root servers in the world.

While there are only 504 root servers, there are multiple physical servers (that mirror them) that operate in different geographic locations to provide reliable service in case of hardware failures.

When an application specifies a host name, go to local domain name server and try lookup in its stored cache. If not stored there, then local DNS requests address from a root server. The root server determines appropriate name server and forwards request.
Routing Protocols

Routers (or gateways) are special purpose machines on the Internet that determine the path for packets from source to destination.

When a router receives a packet, inspects the destination address. It looks up that address in a routing table and based on the contents of the table, forwards the packet to another router (or to its final destination if possible).

There are various protocols used to determine routing methods, such as a “hop path” metric (i.e. the number of steps from the source to the destination), and procedures to detect abnormal events such as unreachable hosts and network congestion.
The World Wide Web
The World Wide Web is the world’s largest client/server system. Communication occurs via message passing.

- Within a web browser, a client selects a URL of a desired page.
- The browser requests a resource from a server.
- The server responds with message containing the type of resource (HTML, gif, pdf, zip, ...), and supplies the contents.
- The browser uses “type” info to correctly display the information (e.g. HTML).
- If the page contains other items (images, applets, ...), the browser must request each item separately.
HTTP

Hypertext Transfer Protocol (HTTP): Application-layer protocol for distributed, collaborative, hypermedia information systems. It presumes an underlying and reliable transport layer protocol (such as TCP).

Generic, stateless, object-oriented.

HTTP can be used for many tasks, such as name servers and distributed object management systems. It is the underlying language of the Web.
HTTP (cont.)

HTTP/1.0 allows only connectionless message passing.
▶ Each request/response requires a new connection.
▶ To download a page with images requires multiple connections.
▶ Can overload the server, require lots of overhead.

HTTP/1.1 provides persistent connection by default.
▶ Once client and server connect, remains open until told to close it (or it timeouts).
▶ Reduces number of connections, saves overhead.
▶ Client can send multiple requests without waiting for responses e.g. can request all images in a page at once.
GET request

Most URL’s (Uniform Resource Locator) have the form:

    protocol://serverName URL
    e.g.   http://www.csc.liv.ac.uk/ martin/

To retrieve a document via HTTP from a server, we can issue a GET request using the HTTP protocol:

    GET URL HTTP/1.1
    Host: serverName

The Web server receives the GET request message and then can respond to it.

A request of this type is automatically generated by a web browser when you select a URL. Such a request could also come from a link checker, a search engine robot, . . .

Alternatively, a request can come directly from a telnet connection using port 80.
GET example

bash-3.1$ telnet www.csc.liv.ac.uk 80
Trying 10.128.0.3...
Connected to www.csc.liv.ac.uk (10.128.0.3).
GET /~martin/index.html HTTP/1.1
Host: www.csc.liv.ac.uk

Server response has assorted header information, followed by the webpage content itself.

HTTP/1.1 200 OK
Date: Mon, 08 Oct 2011 10:01:15 GMT
Server: Apache/2.0.58
HP-UX_Apache-based/Web_Server (Unix) mod_perl/1.99_16 Perl/v5.8.7 DAV/2 PHP/5.0.4
Last-Modified: Mon, 01 Oct 2011 14:55:16 GMT
ETag: "ec3f-1122-9fd83d00"
Accept-Ranges: bytes
Content-Length: 4386
Content-Type: text/html

<!DOCTYPE html>
<head>
    . . .
    . . .
</head>
<body>
    . . .
    . . .
</body>
</html>

Connection closed by foreign host.
Response Header Fields

The first line of the server’s response contains a status code.

- **200 OK**  
  Request was processed successfully

- **301 Moved permanently**  
  Document has been moved

- **304 Not modified**  
  If cached version is up-to-date

- **400 Bad request**  
  Syntax error in client’s request

- **403 Forbidden**  
  Client is not allowed access (e.g. protected)

- **404 Not found**  
  File could not be found

- **500 Internal server error**  
  Server failed

- **503 Service unavailable**  
  Server is overloaded
Other response header fields

In addition to the status code, the server’s response may include:

- **Date**: response time (in GMT)
- **Server**: identification info on the server
- **Last-modified**: time document was last changed (in GMT)
- **Content-length**: size of document, in bytes
- **Content-type**: file format (e.g., html, gif, pdf)
- **Expires**: prevents browser from caching beyond date
bash-3.1$ telnet www.csc.liv.ac.uk 80
Trying 10.128.0.3...
Connected to www.csc.liv.ac.uk (10.128.0.3).
Escape character is '^^>'.
GET /~martin/foo.html HTTP/1.1
Host: www.csc.liv.ac.uk

HTTP/1.1 404 Not Found
Date: Mon, 08 Oct 2011 10:10:35 GMT
Server: Apache/2.0.58
HP-UX_Apache-based_Web_Server (Unix)
mod_perl/1.99_16 Perl/v5.8.7 DAV/2
PHP/5.0.4
Accept-Ranges: bytes
Transfer-Encoding: chunked
Content-Type: text/html

<html>
<head>
<title>404 Not Found</title>
<link rel="stylesheet" href="./fonts.css" type="text/css">
</head>
<body>
 . . .
<td id="main_content">
<div id="topmargin"><a name="skipnav" tabindex="1"></a></div>
<!- Beginning of main page content ->
<h1>Page Not Found</h1>
<p class="text">The requested URL was not found on this server.</p>
 . . .
</html>

Connection closed by foreign host.

If file not found, response includes 404 status code and generic error page.
Other Request Methods

HEAD     similar to GET, but requests header information useful for checking to see if a document exists, how recently it’s been updated

POST    similar to GET, but encodes inputs differently useful for submitting form contents to a CGI program, and is also often used in PHP scripts for submitting information

PUT     upload a document to the server  
         (new in HTTP/1.1)

DELETE  delete a document from the server  
         (new in HTTP/1.1)
GET and POST methods (more to come...) 

When we discuss CGI programming, we will see more about how the GET and POST methods are used to pass information to a server.

Information is extracted from that sent to the server in different ways depending upon the method used to send the data.

PHP provides special built-in associative array variables ($_GET[ ] and $_POST[ ]) that provide easy access to information that a server receives via one of these requests.
Browsers cache pages to save downloading.

Maintain temporary storage (cache) for recent pages.

When a page is requested, check to see if already in cache.

If not in the cache, issue GET request. When response message arrives, display page and store in cache (along with header info).

If already stored in the cache, the browser might send a GET request with “If-Modified-Since” header set to the data of the cached page.

When response message arrives, if status code 200, then display and store in cache. If status code 304, then display cached version instead.
HTTP message passing is transaction-based, stateless.

Many e-commerce applications require persistent memory of customer interactions.

E.g., amazon.com remembers your name, credit card, past purchases, interests.

Other sites use “shopping carts” for purchases, logins for access, etc.

Netscape’s solution: cookies

A cookie is a collection of information about the user, and a browser can store some information on your machine that it can later request.
Server can download a cookie to the client’s machine using the “Set-cookie” header in a response.

Set-cookie: CUSTOMER=Alex_Thompson; PATH=/; EXPIRES=Thursday, 29-Jan-2016 12:00:00

When user returns to URL on the specified path, the browser returns the cookie data as part of its request.

Cookie: CUSTOMER=Alex_Thompson