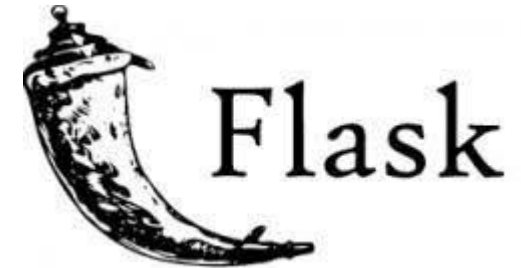


# Introduction to Agile Web Development

CITS3403 Agile Web Development

# Welcome to CITS3403 Agile Web Technology

- Focus on **programming** for the WWW and **agile software development**
- Includes
  - How the web works
  - Markup languages and protocols
  - Web Styling with CSS and Bootstrap
  - Document object models and event handling
  - Client-side scripting with Javascript, JQuery and AJAX
  - Flask Python web application development
  - Agile development and GIT
  - MVC architecture and object relational modelling
  - Deployment on Heroku
  - REST APIs
- Primarily use open source and free technologies
  - you can use it at home



# Course Information



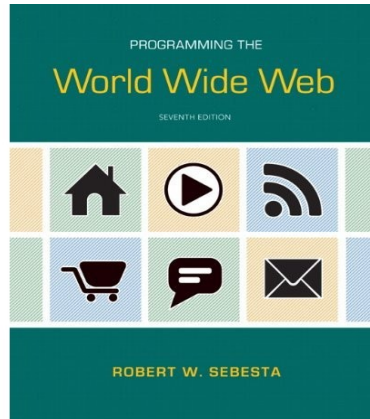
- Double-badged CITS3403 and CITS5505 – *contains mostly the same material...*
- Unit Coordinator & Lecturers:
  - Dr Tim French ([tim.french@uwa.edu.au](mailto:tim.french@uwa.edu.au))
- Lab Demonstrators:
  - Michael Stewart and Tom Smoker
- Teaching sessions
  - Lectures: Wednesday 2-4pm
  - CITS3403 Lab: Mon, Wed, Fri 4-6 Lab CSSE 2.01/ELEC151 (starting week 2) *BYOD is fine*
  - CITS5505 Laboratory, 12-1pm, CSSE:2.03 *BYOD is fine*
  - Consultation: Wed 11-12, Rm 2.14

Mid-semester test: Thursday April 4, Wilsmore Lecture Theatre, 4-5pm (**worth 20%**)
- Webpages:
  - LMS: [https://lms.uwa.edu.au/webapps/blackboard/execute/announcement?method=search&context=course&course\\_id=\\_43183\\_1](https://lms.uwa.edu.au/webapps/blackboard/execute/announcement?method=search&context=course&course_id=_43183_1)
  - Unit web page: <http://teaching.csse.uwa.edu.au/units/CITS3403/index.php?fname=lectures>
  - Help forum: <https://secure.csse.uwa.edu.au/run/help3403>

# Unit Timetable

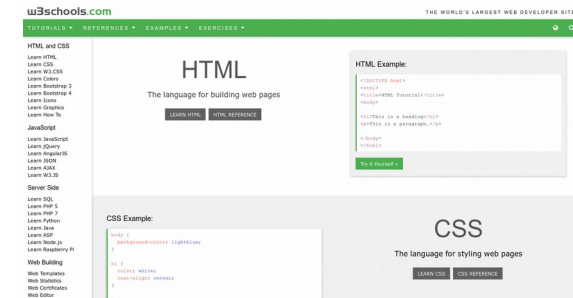
|          |  |  |  |  |  |
|----------|--|--|--|--|--|
| 11:00 AM |  |  |  |  |  |
| 12:00 PM |  |  | CITS5505 (SEM-1)<br>Laboratory<br>CSSE: [ 203]<br>Wks 10-16, 18-21 |  |  |
| 1:00 PM  |  |  |  |  |  |
| 2:00 PM  |  |  | CITS3403 (SEM-1)<br>Lecture<br>GGGL: [ G21]<br>Wks 9-16, 18-21     | CITS5505 (SEM-1)<br>Lecture<br>GGGL: [ G21]<br>Wks 9-16, 18-21 |  |
| 3:00 PM  |  |  |  |  |  |
| 4:00 PM  | CITS3403 (SEM-1)<br>Laboratory<br>CSSE: [ 201]<br>Wks 10-16, 18-21 |  | CITS3403 (SEM-1)<br>Laboratory<br>ELEC: [ 151]<br>Wks 10-16, 18-21 | CITS3403 (SEM-1)<br>Examinatio                                 | CITS5505 (SEM-1)<br>Examinatio                                     |
| 5:00 PM  |  |  |  |  | CITS3403 (SEM-1)<br>Laboratory<br>CSSE: [ 201]<br>Wks 10-16, 18-21 |
| 6:00 PM  |  |  |  |  |  |

# Recommended Readings

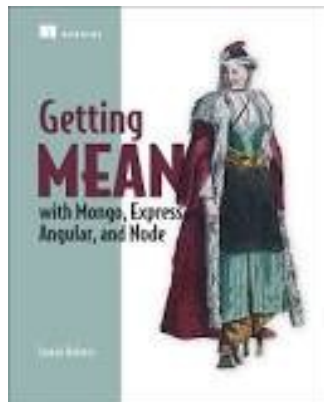


- **General Reference**

Robert W. Sebesta,  
*Programming the  
World Wide Web  
2013*,  
7th Edition, Pearson/  
Addison Wesley.



- **For Labs Part I**
- W3Schools, available at:  
<https://www.w3schools.com/>



- **Not used this year**

Simon Holmes,  
*Getting MEAN with  
Mongo, Express,  
Angular, and Node*  
1st Edition, Manning  
Press, 2015.



- **For Labs Part II**

Miguel Grinberg  
Flask Mega-Tutorial  
Available at:  
[https://  
blog.miguelgrinberg.com/  
post/the-flask-mega-tutorial-  
part-i-hello-world](https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world)

# Assessment

- Mid-semester test: Thursday, Week 6.
  - 20% or final grade
  - 4-5 questions, written answers
- Project: Monday, May 20.
  - 30% of final grade
  - Done in pairs
  - Build a voting/ranking web-site. Lab work will step through this process.
- Final exam: End of semester examination period
  - 50% of final grade
- Please ensure you have consulted the Unit Outline for information on:
  - unsatisfactory progress
  - late assessment penalties
  - plagiarism policy
  - including ACE and academic misconduct
  - Faculty marks adjustment policy

# A Brief History of the Internet

- DoD created the Advanced Research Projects Agency (ARPA) in 1958 (now DARPA)
- ARPA wanted
  - communications, program sharing, remote computer access
  - robust - continue to work if some nodes “taken out” by malicious forces
- ARPAnet - late 1960s and early 1970s
  - about a dozen ARPA-funded research labs and universities
  - graduate students played a large part in its development!
  - didn’t live up to intentions - mostly text-based email, limited reach
  - but the snowball had started rolling...
- Non-ARPA-funded Universities wanted in so other networks were created in the late 70s and early 80s
  - BITnet (Because It’s Time Network), initially electronic mail and file transfer
  - CSnet (Computer Science Network), primarily email

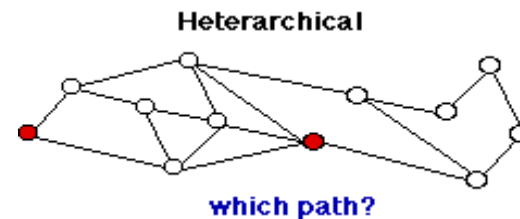
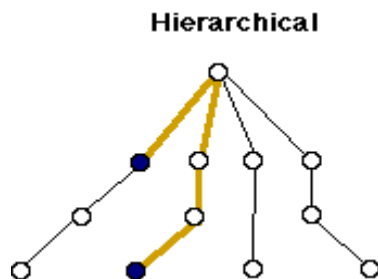
# NSFnet: the birth of the internet

- NSFnet - 1986
  - National Science Foundation (NSF)
  - originally for non-DoD funded places
  - initially connected five supercomputer centers
  - spread to other academic institutions and research labs
  - 1988/89 - commercial infiltration - mail, ISPs
  - by 1990, it had replaced ARPAnet for non-military uses
  - soon became the network for all (by the early 1990s)
  - other networks created gateways and eventually merged
    - JANET, BITnet, Usenet,...
  - >1M computers around the world by 1992
  - NSFnet eventually became known as the Internet



# Dynamic routing

- For robust networks, a transmission protocol must find new routes to a destination as preferred routes fail.
- This is achieved by **dynamic routing**, where the routes are selected at the time of transmission, after considering current network conditions.
- Dynamic routing requires a network architecture devoid of critical sites, whose failure will bring down the entire network. That is, the network cannot be hierarchical.



# The Largest Inter-connected network – the Internet



- If the computers that make up a network are close together (within a single department or building), then the network is referred to as a **local area network (LAN)**.
- A network that covers a wide area, such as several buildings or cities, is called a **wide area network (WAN)**. The largest **WAN** in existence is the **Internet**.
- Today the Internet has grown to include hundreds of millions of interconnected computers, smart phones, televisions, printers, fridges and networks
- The physical structure of the Internet uses fiber-optic cables, satellites, phone lines, and other telecommunications media

# In summary, the Internet is

- A network connects computers so they can communicate, exchange information, and share resources.
- The Internet is an infrastructure, in particular a global computer network, supporting data transmission.
- The **Internet is a network of interconnected networks**. If part of its infrastructure is destroyed, data can still flow (in principle) through the remaining networks.
- The Internet uses high-speed data lines (backbones) to carry data. Smaller networks connect to the backbone, enabling any user on any network to exchange data with any other user.

# How the Internet works

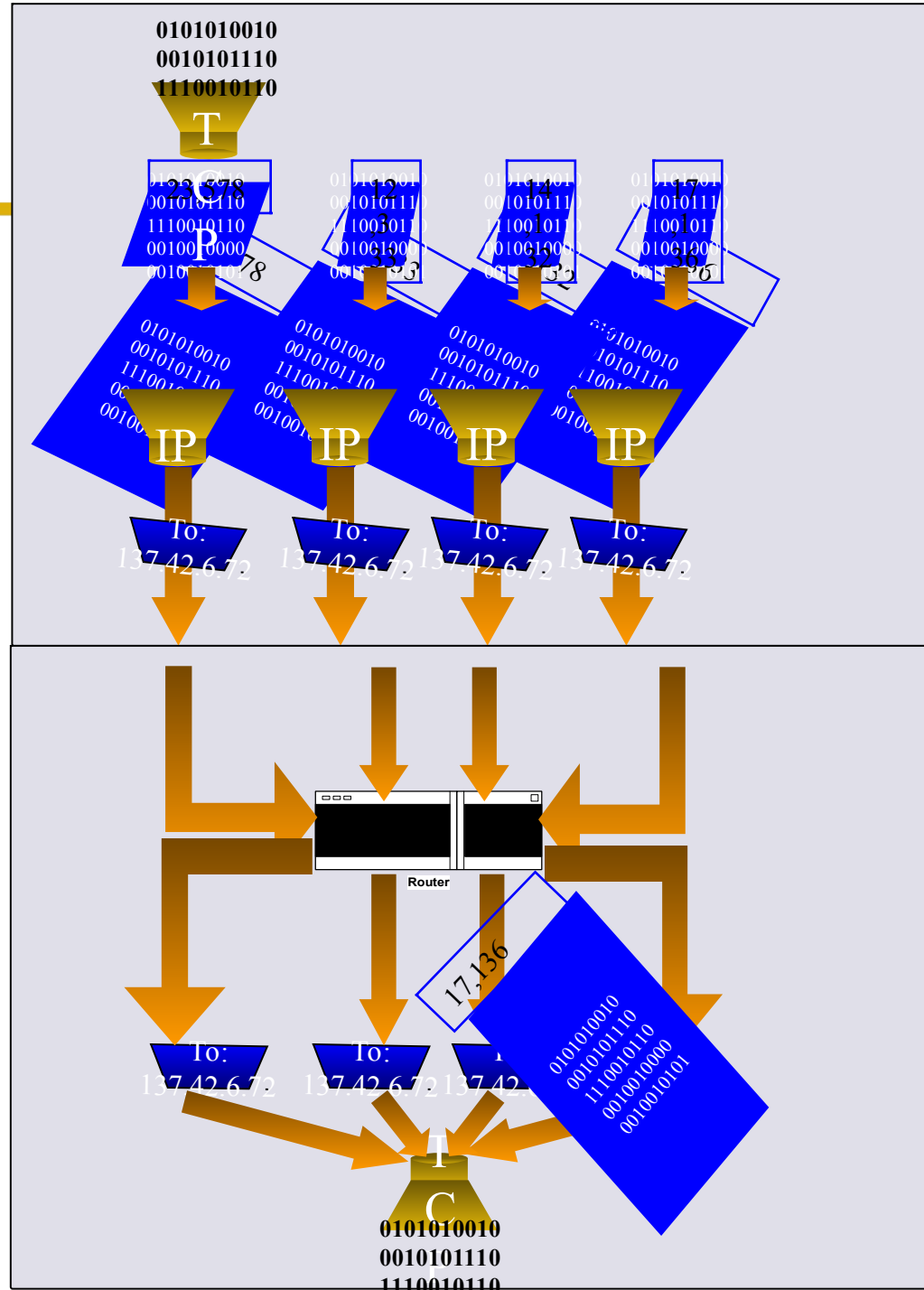
- TCP/IP
- Addressing Schemes
- Domains and Sub-domains
- Routing Traffic Across the Internet
- Client Server Architecture

# How the Internet works - Protocols

- Network communication is made possible only if computers “speak” a common language. The rules and procedures for controlling timing and data format are the protocols and they,
  - **signal** another computer requesting communication. (client)
  - **identify** the requesting computer. (server)
  - **transmit** messages in blocks. (server)
  - **retransmit** if messages fail to arrive. (server)
  - **detect** errors and recovers. (client)
  - **signal** transmission is complete. (client)
  - **terminate** the connection. (server)

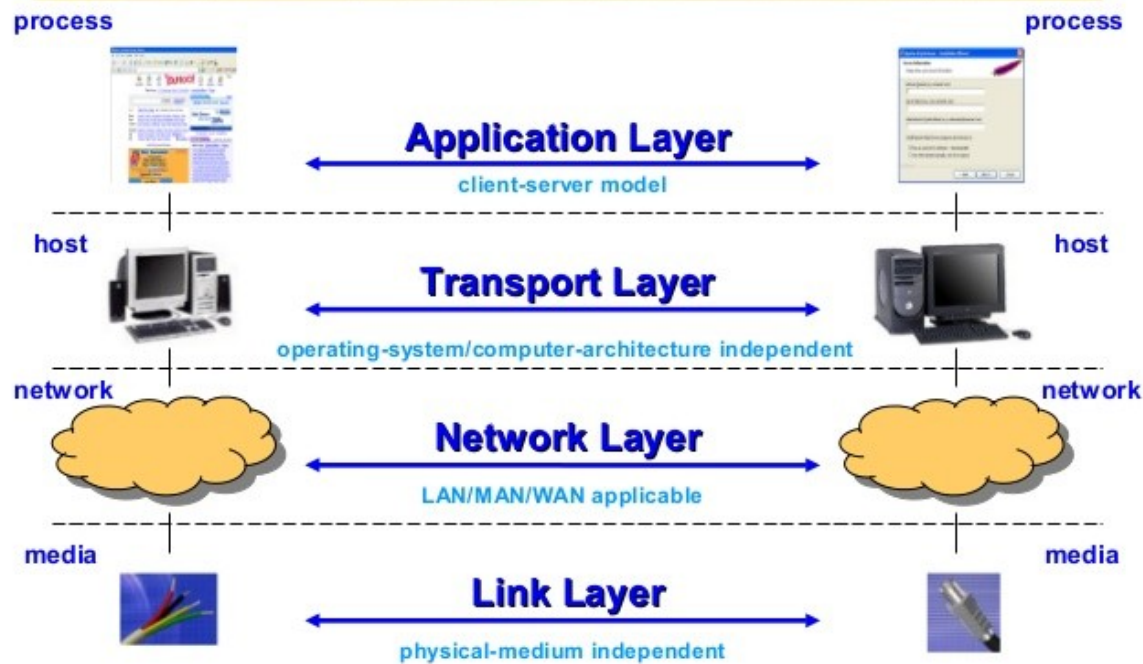
# How the Internet Works – TCP/IP

- Every computer and network on the Internet uses the same protocols - the Transmission Control Protocol/Internet Protocol, or TCP/IP.
- No matter what type of computer system you connect to the Internet, if it uses TCP/IP, it can exchange data with any other type of computer.
- TCP/IP was developed to tolerate unreliable sub-networks and the protocol guarantees proper transmission of data, since the physical network can't.
- For transmission not needing guarantees (even unreliable networks are very reliable) one can use User Datagram Protocol (UDP). Data transmitted by UDP arrive faster, with none of the error detection or correction overheads that are in TCP/IP.



# How the Internet Works – TCP/IP

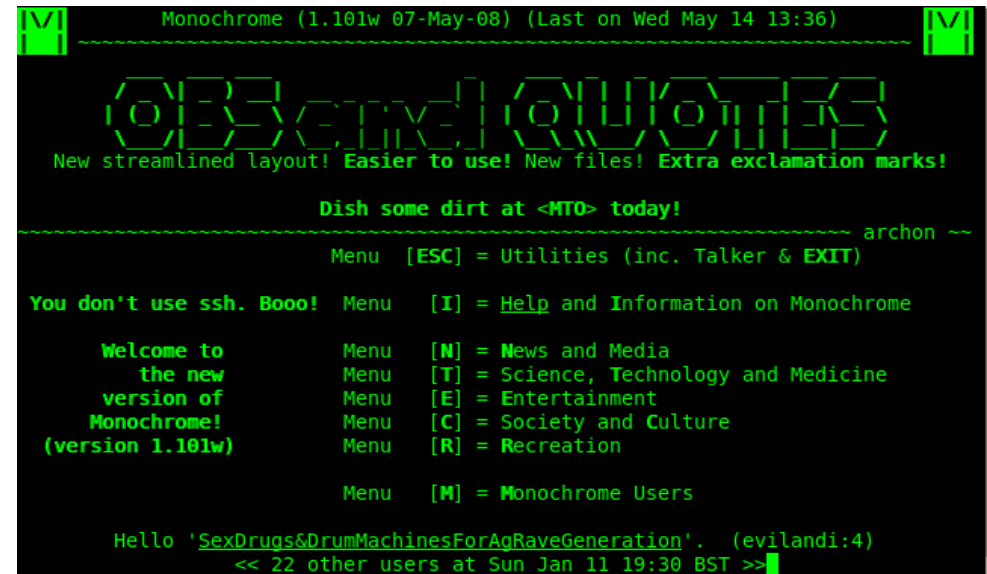
## TCP/IP Network Architecture





# Internet Applications: Bulletin Board Systems (BBS)

- Early interactive software, late 1970s to 1980s
- Users login to:
  - exchange messages through mail or public message boards
  - read news and bulletins
  - upload/download software
  - even on-line games
  - accessed using modem and phone line
  - precursor to today's WWW



# Usenet

- Idea conceived by Duke University grad students in 1979
- Unlike BBS, distributed network of servers (eg. each university)
- Large number of forums called *newsgroups* (not just news - users posts)
- Threaded discussions
- Formed social communities
- Precursor to Internet forums



Source: Benjamin D. Esham, Wikimedia Commons  
[http://en.wikipedia.org/wiki/File:Usenet\\_Big\\_Nine.svg](http://en.wikipedia.org/wiki/File:Usenet_Big_Nine.svg)

# Power to the People - the “Killer App”

Early internet mainly used by people in Uni's and research labs

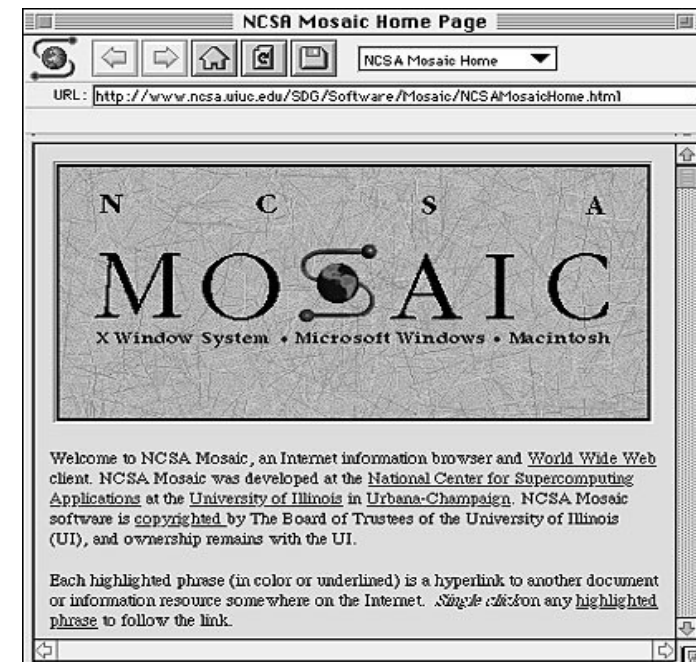
- 1991, CERN publicised new World Wide Web project
- Invented by Tim Berners-Lee and colleagues in 1989
- used TCP (Transmission Control Protocol) and DNS (Domain Name System)
- first browser WorldWideWeb on NeXTSTEP
- httpd (HyperText Transfer Protocol daemon) web Server



# Power to the People - the “Killer App”



- 1993 National Center for Supercomputing Applications (NCSA) at University of Illinois releases V1.0 of Mosaic browser
  - written by a student, Marc Andreessen, and Eric Bina
  - first multimedia browser (mixed images and text)!
- Explosion in internet use!
  - growth of web usage in 1000s of percent
  - changed internet use forever
  - the “killer app” of the 90s



# Client-Server Architecture

## What does a client do?

- Once wired, the user accesses the Web via software, called a browser (e.g. Firefox or Chrome).
- Browsers locate and display information from the Web.
- Communication is by an agreed transmission language or protocol, eg. HTTP (HyperText Transfer Protocol).
- The user requests a Web page through the browser, which communicates this to the server.
- The browser waits for the Web page to be delivered, typically a text file containing HTML instructions.
- The intricate graphics and formatting results from the browser rendering that page in the format defined in the file.

# Client-Server Architecture:

## What does a server do?

- The server's job is somewhat easier.
- The server is software running on a computer, and it responds to client requests for Web pages.
- The Web pages exist on its local file system.
- The server retrieves and then transmits the files to the client.



# What does the “cloud” do?

- A **network** is a structure linking computers together for the purpose of sharing resources such as printers and files
- Users typically access a network through a computer called a **host** or **node**
- A computer that makes a service available to a network is called a **server**
- A computer or other device that requests services from a server is called a **client**
- One of the most common network structures is the **client-server architecture**



# Anatomy of a URL

<http://www.domain.edu.au:1000/path/to/file?parameters=true#fragment>

- **The protocol used.** Typically http, ftp, https, ...
- **The domain name.** A domain name server maps this to an IP address
- **The port number.** Servers have ports 0-65535, but http defaults to port 80.
- **The path (route) to the file to execute.** The file is typically an html file, but it could also be php, text, pdf.
- **The parameters of the request.** These are specified as a set of key value pairs.
- **The fragment.** This anchors to a location in a page.
- There are also hidden parts of the request including the browser name and cookies.



