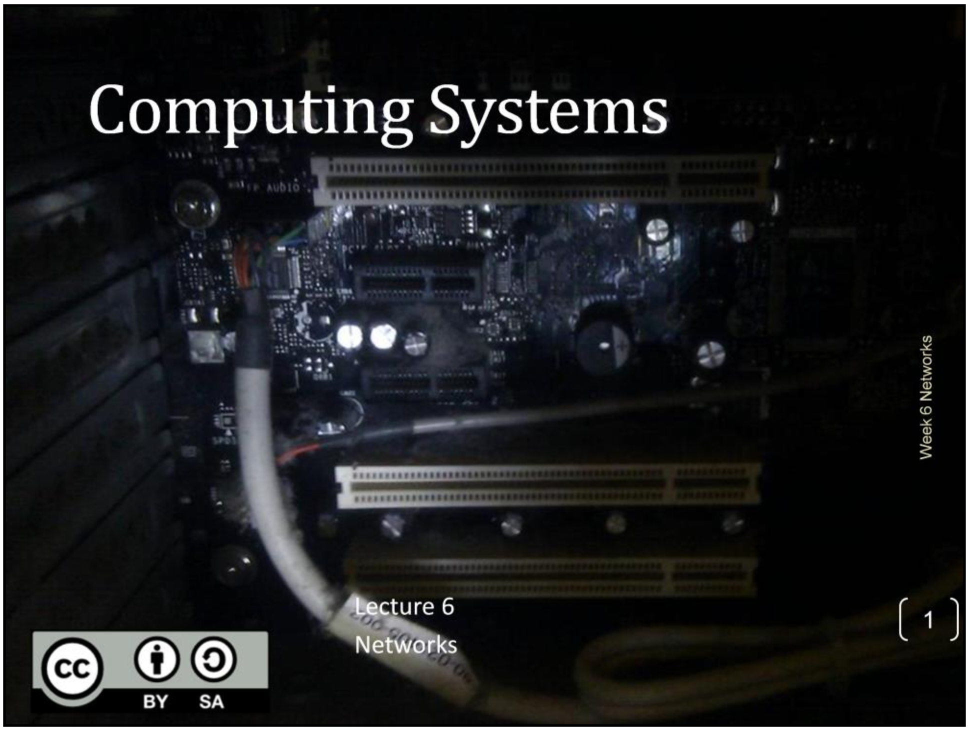

Computing Systems
Week 6 Networks
Lecture 6
Networks
1
BY SA


Starting with a view of some of the forms of current networking applications that exist..

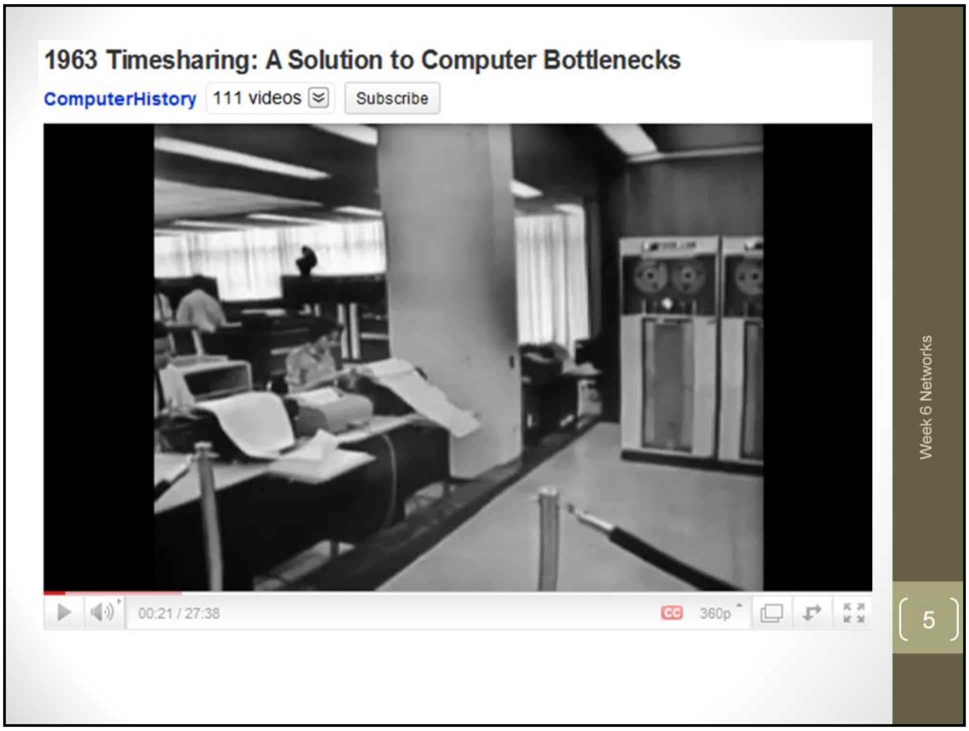
Multi-user on-line role playing games played at almost real time.


We can be on line on the internet and networked at any time.

Modern smartphones include a wide array of networking capabilities – mobile internet, bluetooth (short range connectivity, to headsets or even to allow the phone to act as an input device for a PC) and access to GPS (global positioning data – phone receives signals from satellites)

# Early Multi-User Computing

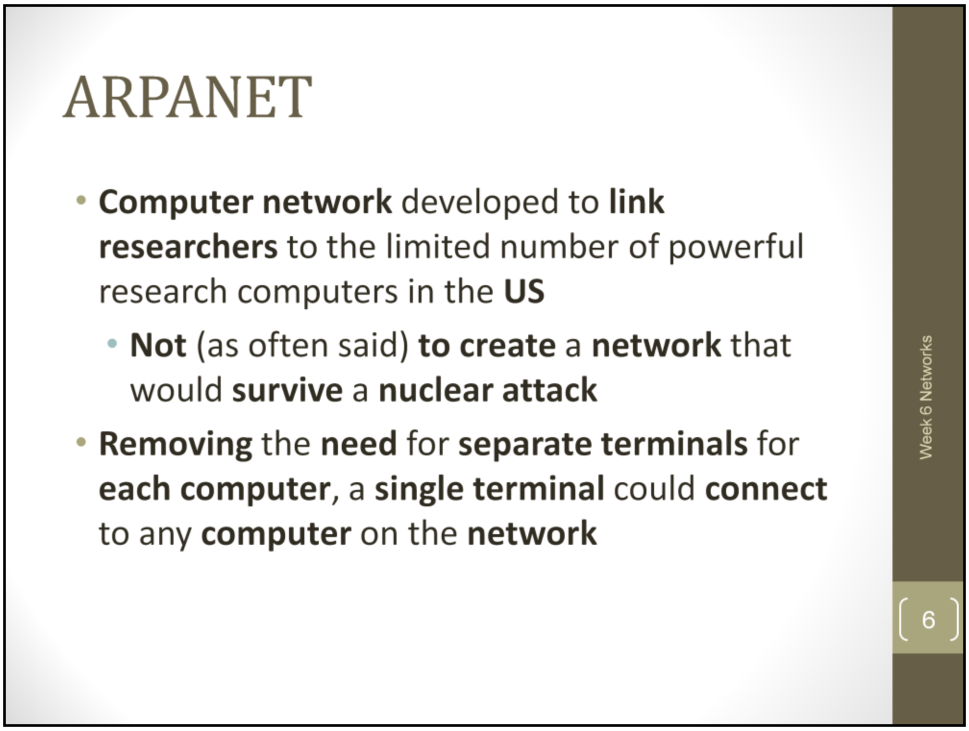
- **Early** computers were standalone
  - no communication between different computers
- Development of **terminals** allowed **multiple** users (possibly in different **locations**) to use the same **computer** at the same time
  - **Terminal** only used for **keyboard input** and **text output** (screen or printer)
- Later generations of terminals added graphics & local processing (e.g. **PLATO IV**)



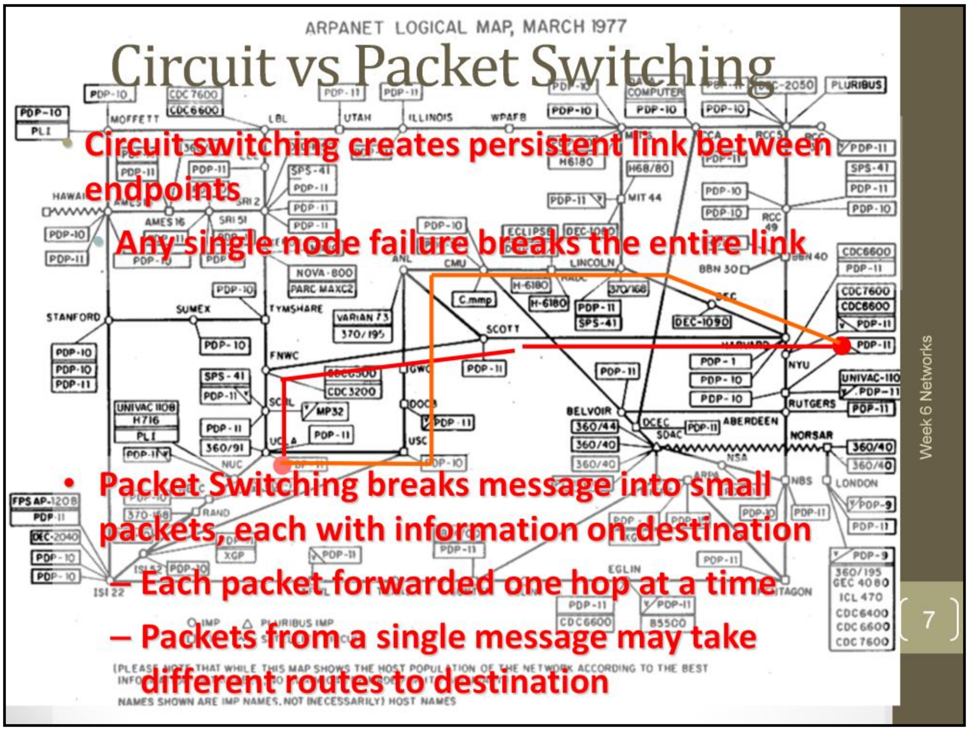

Click to launch video

In lecture watch only first minute of this – watch as much as you would like at home!

One computer with access from terminals




Not to survive nuclear attack, but to survive the loss of individual nodes without compromising the whole network.




Packets-switching break the overall message into small packets, each having information on the destination. Each packet is forwarded from the current node to the next node and sent onwards to the destination one hop at a time. The different packets that form a single message may take different routes to the destination.  Two routes are shown, if there is a network failure at a node the packets will be routed through a different node and sent round the network a different way.

ARPNET network in March 1977

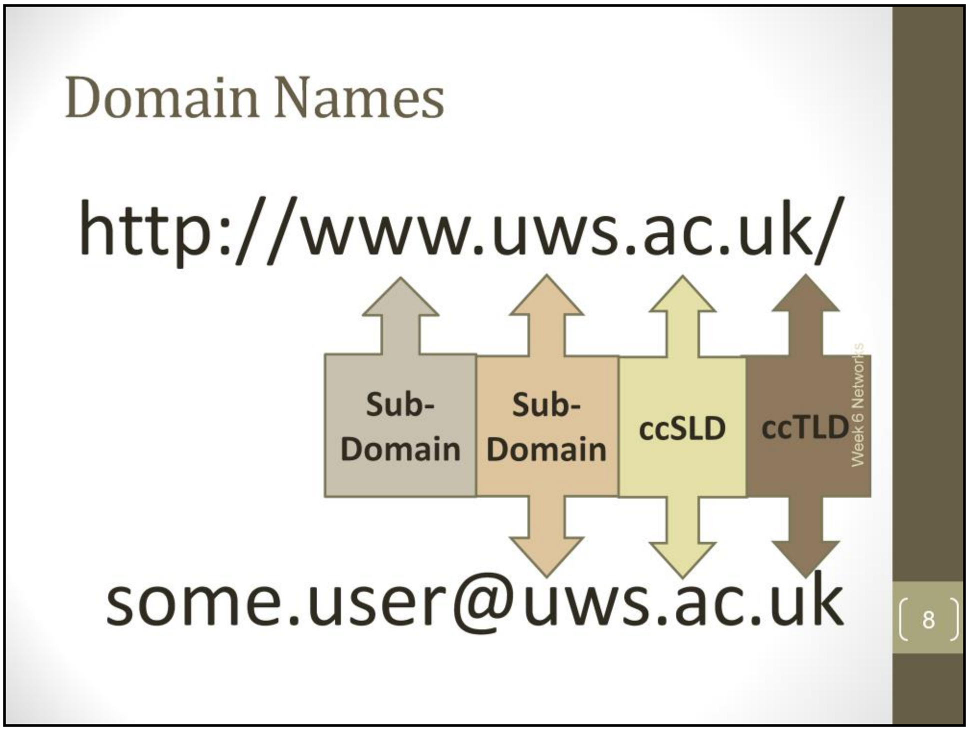
Connections from Hawaii to London

Early attempts at networking suffered if a single node on the network failed

> **Circuit switching** creates a point to point connection through an entire network; connection lost if any node fails → used in telephone calls

**Packet Switching** allows successful communication even where nodes fail

> **Message split into chunks**
>
> Each **chunk** passed around the network from node to node in direction of destination
>
> Donald Davies (NPL,UK)
>
> Also Paul Baran (US) – Message Blocks




Top link → web page URL

Bottom link → email address

Could have:

Student.uws.ac.uk

Staff.uws.ac.uk

ccTLD = **country code Top Level Domain**. Alternative TLDs include .com and .edu (mainly North America and international use)

ccSLD = **country code second level domain**. In some countries (including UK) only a restricted set of SLDs are available – e.g. .co.uk for companies, .ac.uk for academic sites, .gov.uk, etc etc. Other countries allow any SLD which can be used creatively eg. .ly is the TLD for Libya which is used for e.g. bit.ly (a url shortening service)

# Circuit vs Packet Switching

- ***Circuit switching*** creates **persistent link** between **endpoints**
  - Any **single node** failure **breaks** the **entire link**
- ***Packet Switching*** allows **successful communication** even where **nodes fail**
  - ***Packet Switching*** breaks message into **small packets**, each with information on **destination**
  - ***Packets*** from a **single** message may take **different routes** to **destination**

## Data Packets

- Data between machines sent in *packets*
  - **Large data** transfers broken into **multiple packets**
- Each packet contains
  - **IP** (**Internet Protocol**) **addresses** of **destination** & **sender**
  - **Packet sequence number**
  - **Data contents**
  - **Cyclic Redundancy Check** (**CRC**) data to verify correct transmission of packet

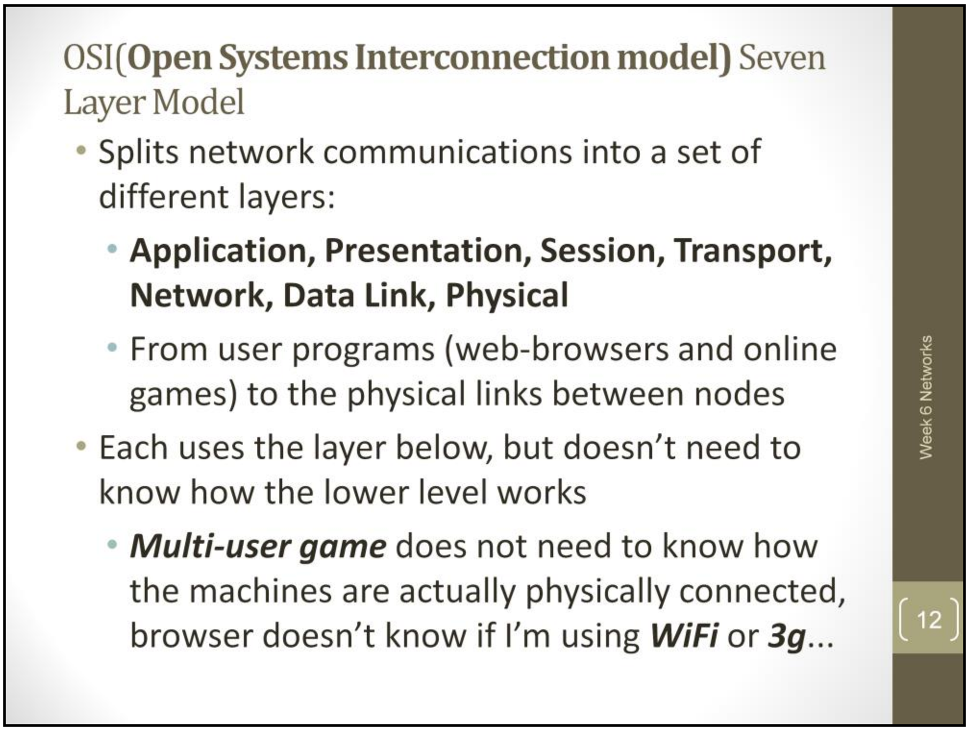



HCW p317

# Quick quiz

- Join the 'Socrative' app 'Room 642124' and try the quick quiz.

Week 4 Compilers and Interpreters 11




HCW p 334

**Physical layer** → wires or wireless connections used.

Each layer uses the layer below but does not need to know how it works.

**Application**→ only part USER SEES, user does not see most of the work the processor is doing.

→ Converts a MESSAGE'S DATA into BITS and attaches a HEADER identifying SENDER/RECEIVING COMPUTER.

**Presentation** → TRANSLATES MESSAGE into a language RECEIVING COMPUTER can understand (often ASCII – encodes TEXT as BITS), → COMPRESSES DATA, → possibly ENCRYPTS DATA, → adds another HEADER specifying LANGUAGE & COMPRESSION / ENCRYPTION SCHEMES.

**Session → OPENS communications, → sets BOUNDARIES (called BRACKETS) for the BEGINNING and END of the MESSAGE and establishes whether the message will be sent HALF DUPLEX(each computer takes turns in sending and receiving) or FULL DUPLEX(both computers SEND / RECEIVE at the same time).**

**Transport** → PROTECTS the data being sent, → it subdivides DATA into SEGMENTS and creates CHECKSUM tests (mathematical sums based on the contents of the data) that can be used later to determine whether the data had been scrambled, → it makes BACKUP COPIES of the DATA, → the TRANSPORT HEADER identifies each SEGMENT'S CHECKSUM and it's POSITION in the message.

**Network** → Selects a ROUTE for the MESSAGE, → it forms SEGMENTS into PACKETS, COUNTS them, and adds a HEADER containing the SEQUENCE of the PACKETS and ADDRESS of the RECEIVING COMPUTER.

**Data Link** → Supervises the TRANSMISSION, → It CONFIRMS the CHECKSUM, adds ADDRESSES and DUPLICATES the PACKETS, → this layer keeps a COPY of the PACKETS until it receives confirmation from the next point along the route that the PACKET has ARRIVED UNDAMAGED.

Physical → ENCODES the PACKETS into the MEDIUM that will carry them (e.g. analogue SIGNAL – if the message is going across a TELEPHONE LINE), → and SENDS the PACKETS along that line.

# OSI seven layer Model

Data

- **Application Layer**
- *Converts* a message's **data** into **bits** and *attaches* a **header** *identifying* the **sending** and **receiving computers**

Data

- **Presentation Layer**
- *Translates* the message into a **language** that the **receiving computer** can understand (often **ASCII**, a way of encoding text as bits). *Compresses & encrypts* the **data**. Adds another **header** specifying the **language** as well as the **compression** and **encryption schemes**.

Data

- **Session Layer**
- *Opens communications.* Sets boundaries for the beginning and end of the message. Establishes whether the message will be sent *half duplex* or *full duplex*. **Details** are placed into a **session header**.

Segments

- **Transport Layer**
- Protects the data being sent. **Subdivides** the **data** into **segments** and **creates checksum tests used** later to **determine** whether the **data** was **scrambled**. Makes **backup copies** of the **data**. The **header** *identifies* each **segment's checksum** and its **position** in the **message**.

Packets

- **Network Layer**
- Selects a **route** for the **message**. Forms **segments** into **packets**, **counts** them, and **adds** a **header** containing the **sequence** of **packets** and the **address** of the **receiving computer**.

Frames

- **Data Link Layer**
- **Supervises** the **transmission**. Confirms the **checksum**, and then addresses and duplicates the packets. Keeps a **copy** of each **packet** until it **receives confirmation** from the next **point** along the **route** that the **packet** has **arrived undamaged**.

Bits

- **Physical Layer**
- **Encodes** the packets into the **medium** that will **carry** them-such as an **analogue** signal, if the message is going across a **telephone** line-and **sends** the **packets** along that **medium**.

# Application Layer

- The first layer, the **application** layer, is the only part of the process a user sees, and even then the user doesn't see most of the work the application does to prepare a message for sending over a network.
- The layer *converts* a message's **data** into **bits** and *attaches* a **header** *identifying* the **sending** and **receiving computers**.

# Presentation Layer

- The **presentation** layer *translates* the message into a **language** that the **receiving computer** can understand (often **ASCII**, a way of encoding text as bits). This layer also *compresses* and perhaps *encrypts* the **data**. It adds another **header** specifying the **language** as well as the **compression** and **encryption schemes**.

# Session Layer

- The **session** layer *opens communications*. It *sets* boundaries (called **brackets)** for the beginning and end of the message and establishes whether the message will be sent *half duplex***,** with each computer taking turns sending and receiving, or *full duplex***,** with both computers sending and receiving at the same time. The **details** of these decisions are placed into a **session header**.

# Transport Layer

- The **transport** layer protects the data being sent. It **subdivides** the **data** into **segments** and **creates checksum tests** – mathematical sums based on the contents of data – that can be **used** later to **determine** whether the **data** was **scrambled**. It also makes **backup copies** of the **data**. The **transport header** *identifies* each **segment's checksum** and its **position** in the **message**.

# Network Layer

- The **network** layer selects a **route** for the **message**. It forms **segments** into **packets**, **counts** them, and **adds** a **header** containing the **sequence** of **packets** and the **address** of the **receiving computer**.

# Data-Link Layer

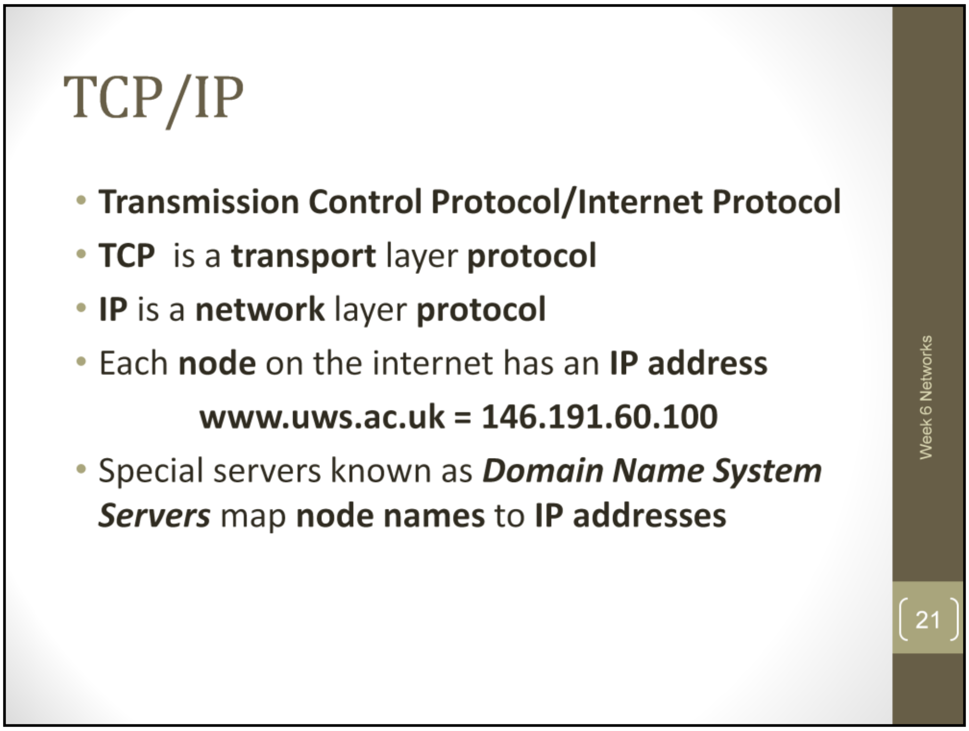
- The **data-link** layer **supervises** the **transmission**. It **confirms** the **checksum**, and then addresses and duplicates the packet**s**. This layer keeps a **copy** of each **packet** until it **receives confirmation** from the next **point** along the **route** that the **packet** has **arrived undamaged**.

# Physical Layer

- The **physical** layer **encodes** the packets into the **medium** that will **carry** them-such as an **analogue** signal, if the message is going across a **telephone** line-and **sends** the **packets** along that **medium**.

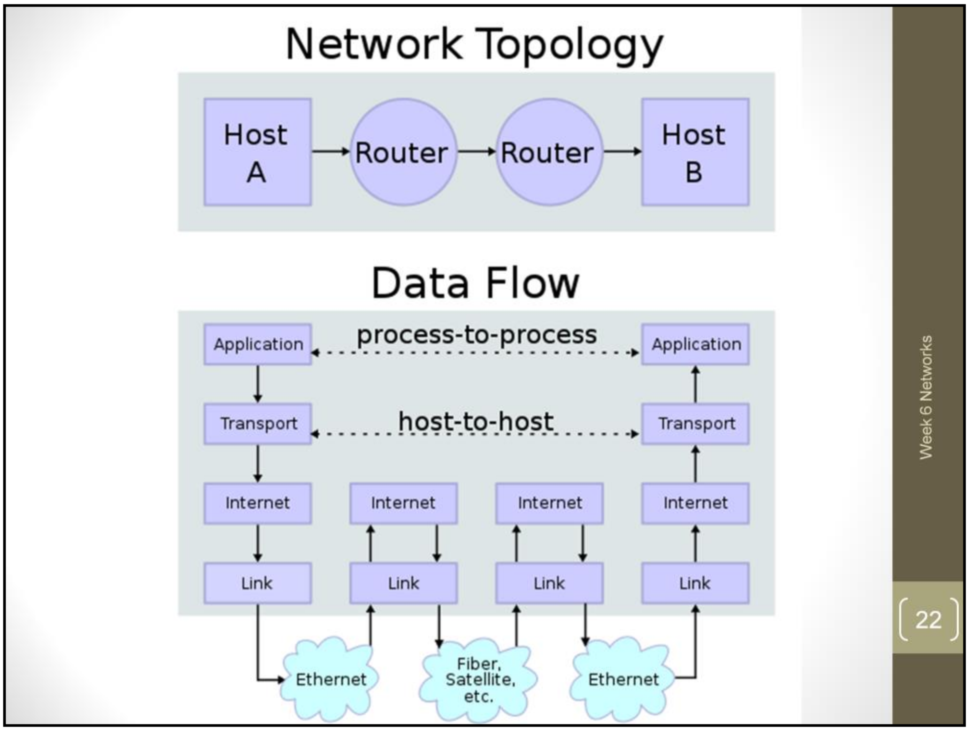
Week 6 Networks

20




All machines on the internet have an IP address.

The current IP addressing system (IPv4) is running out of addresses, the new system (IPv6) is coming online – but not all online yet. Can move a web server from one machine to another machine with a different IP address, we can still get the communication we want to get.




Applications don't necessarily know much about what is happening in the lower layers. One application makes a connection to another, while *under the hood* a range of other protocols are used to create the connections and transfer data

In this example the two host machines don't need to know about the particular networks in use

This diagram shows the Internet Protocol Suite of layers.

Transport layer deals with the host to host connections.

```
C:\Users\admin>tracert 146.191.60.100
Tracing route to uws.ac.uk [146.191.60.100] over a maximum of 30 hops:
  1   67 ms   <1 ms    1 ms  XXXXXXX-XxX [192.168.2.1]
  2   36 ms   40 ms   34 ms  xx.xxx.xxx.x
  3   24 ms   16 ms   25 ms  renf-cam-1b-v125.network.virginmedia.net [80.4.64.245]
  4   44 ms   18 ms   44 ms  renf-core-1b-ae2-0.network.virginmedia.net [195.182.176.181]
  5   40 ms   26 ms   19 ms  leed-bb-1b-as3-0.network.virginmedia.net [213.105.175.225]
  6   32 ms   36 ms   27 ms  leed-bb-1a-ae0-0.network.virginmedia.net [62.253.187.185]
  7   27 ms   24 ms   20 ms  brnt-bb-1b-as2-0.network.virginmedia.net [213.105.175.26]
  8   51 ms   29 ms   28 ms  brnt-tmr-1-ae5-0.network.virginmedia.net [213.105.159.50]
  9   59 ms   51 ms   56 ms  telc-ic-1-as0-0.network.virginmedia.net [62.253.185.74]
 10   32 ms   26 ms   18 ms  linx-gw2.ja.net [195.66.226.15]
 11   69 ms   38 ms   67 ms  ae1.lond-sbr3.ja.net [146.97.35.173]
 12   33 ms  158 ms   32 ms  as0.read-sbr1.ja.net [146.97.33.165]
 13   36 ms   29 ms   32 ms  so-5-0-0.warr-sbr1.ja.net [146.97.33.90]
 14   42 ms   46 ms   46 ms  ae14.glas-sbr1.ja.net [146.97.33.122]
 15   40 ms   46 ms   41 ms  clydenet.glas-sbr1.ja.net [146.97.40.98]
 16   46 ms   43 ms   47 ms  paisley-ge1-1-glasgowpop-ge8-13-v200.clyde.net.uk [194.81.62.202]
 17   32 ms   37 ms   36 ms  146.191.225.1
 18   40 ms   67 ms   39 ms  146.191.241.9
 19   40 ms   36 ms   32 ms  146.191.241.5
 20   34 ms   35 ms   30 ms  uws.ac.uk [146.191.60.100]
Trace complete.
```

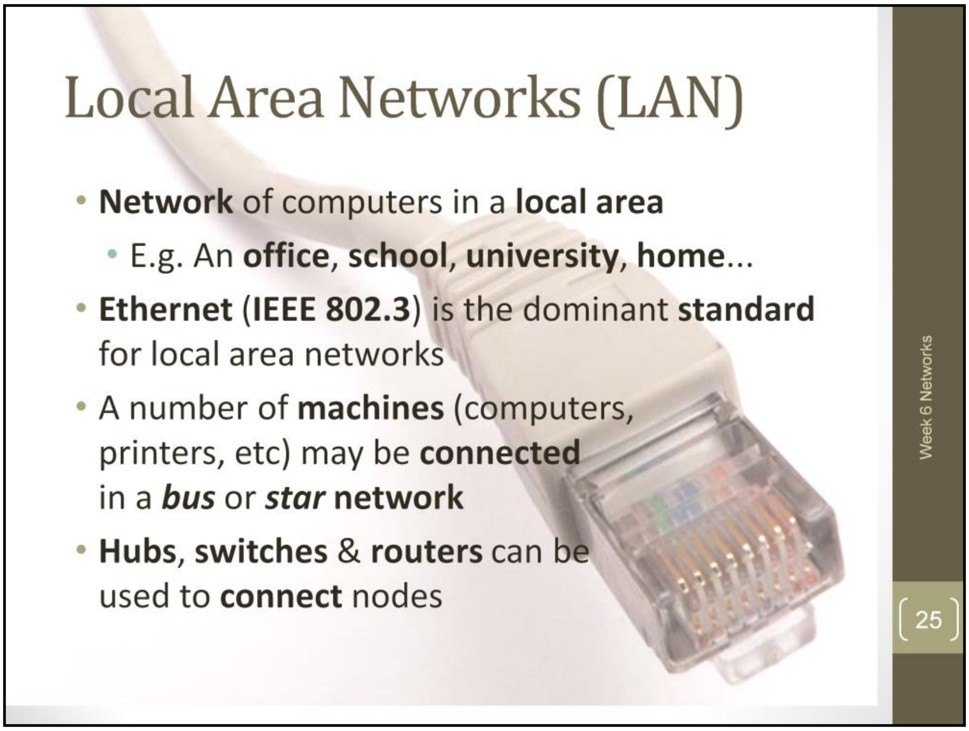


Trace route from my home machine (in the Glasgow area) to the UWS homepage (hosted on a server in the Glasgow area)...

Route goes via (1) my home WiFi router onto the (2-9) Virgin network backbone, apparently through Leeds and further south. Connects to the academic (10-15) JA.NET network in London (line 11) before travelling back up to the UK to the clydenet network and the UWS internal network (15-20)

# Quick quiz

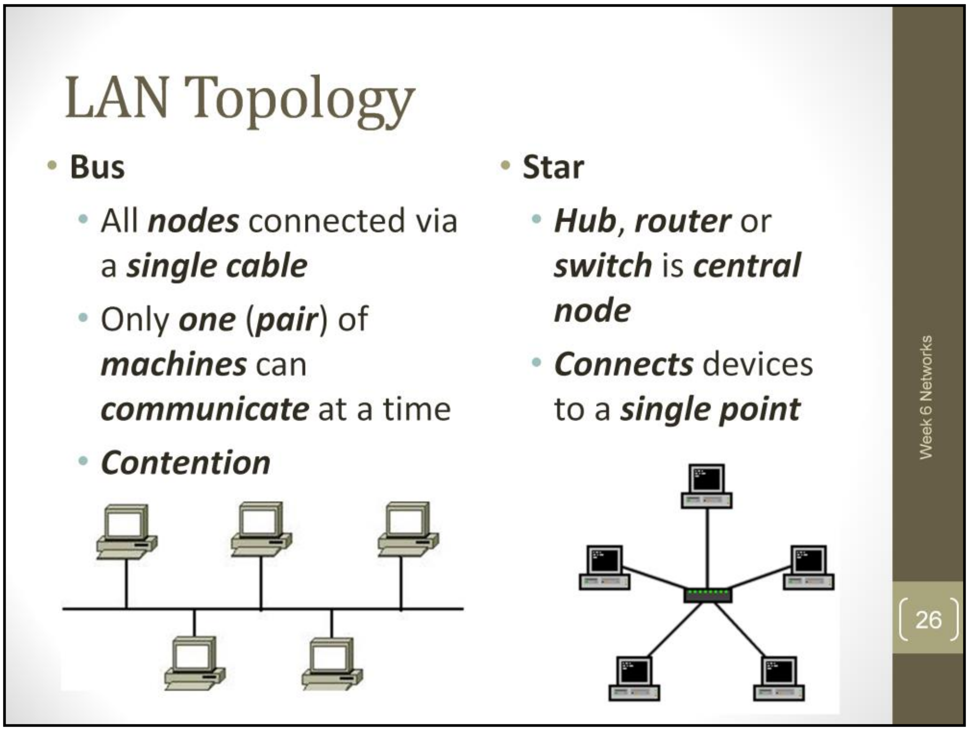
- Join the 'Socrative' app 'Room 642124' and try the quick quiz.




Hubs →An **Ethernet hub**, **active hub**, **network hub**, **repeater hub** or **hub** is a device for connecting multiple Ethernet devices together and making them act as a single network segment. A hub works at the physical layer (layer 1) of the OSI model.

Switches →

Routers → A **router** is a device that forwards data packets between computer networks, creating an overlay internetwork. A router is connected to two or more data lines from different networks. When a data packet comes in on one of the lines, the router reads the address information in the packet to determine its ultimate destination




Contention occurs when two or more computers try to send packets on the network at the same time

Hubs are relatively simple: any signal received is sent to all connected nodes; nodes ignore packets not meant for them.

Both bus and hub based networks can suffer from contention. Nodes can act as packet sniffers too.

Switches send data to specific nodes (can broadcast to all nodes too)

Routers can often act as switches but might not allow broadcast. Routers also provide onward connections to other networks (e.g. Internet)

Local networks can be connected by bridges to other local networks or the internet.

# Home and Small Office

- Small home & office networks typically connect to internet via:
  - **Dial-in**: modem uses standard phone line and exchange. *Limited* **bandwidth**.
  - **DSL** (**Digital Subscriber Line**) requires updated machines at local exchange to offer higher speeds over a standard phone line
    - **Asynchronous DSL**: **Higher** *download* than *upload* **bandwidth**
  - **Cable Internet** uses **cable television infrastructure** to provide **internet access**



HCW p326- 331

# DSL vs Cable Internet

**DSL**

- **Good availability**
- Phone and internet on **one connection**
- Transmission **rate affected** by **distance** to **exchange**, quality of phone line, local exchange equipment
- Big **variance** in **performance**

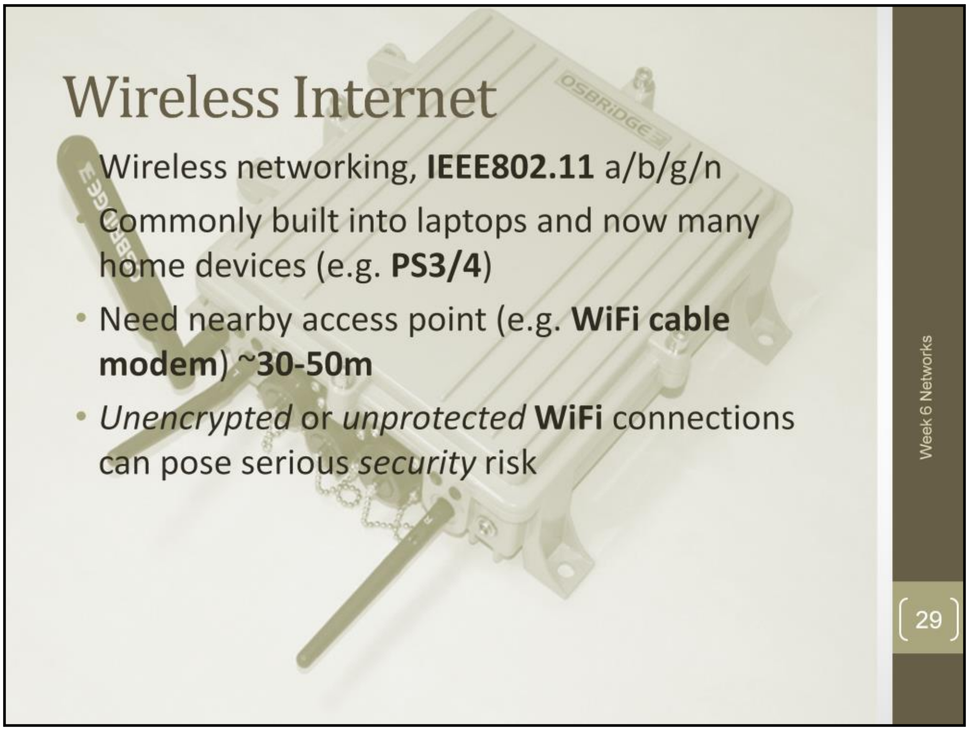
**Cable Internet**

- **Limited availability**
- **Separate connection**
- **Cable** originally designed for **broadcast** – **high use** can cause **network congestion**
- Typically **higher performance** than **DSL**

Week 6 Networks

28

Cable → Was not originally developed for upload. Can cause congestion.



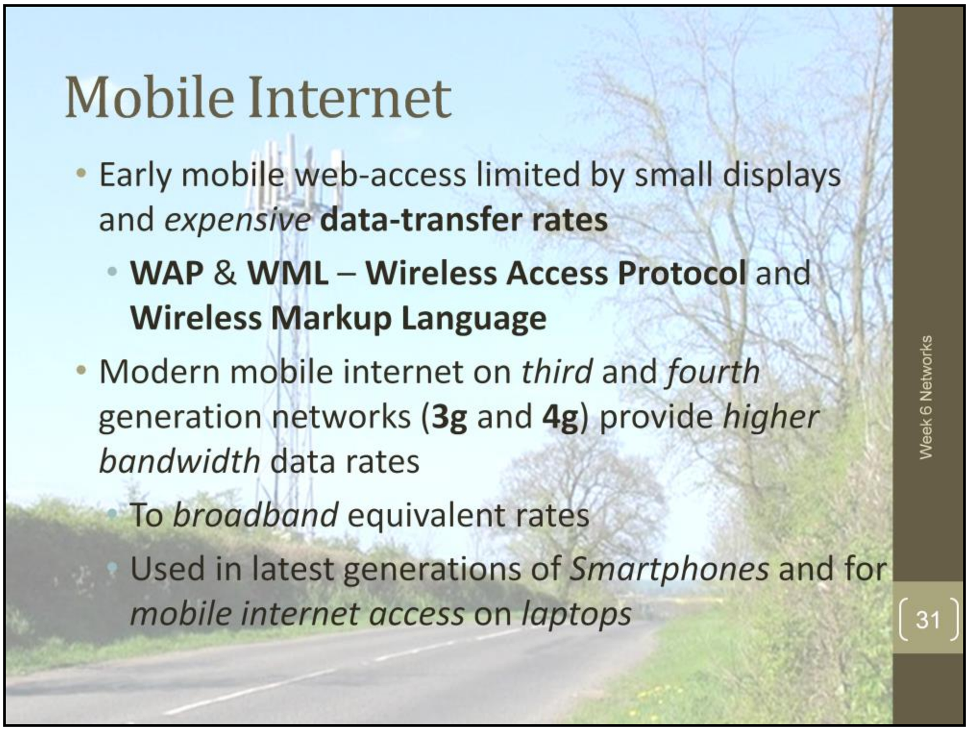

HCW p352-3

IEEE802.11 /n → current standard.

WEP- Wired Equivalent Privacy

# Wireless Security

- Wireless transmissions can be '***heard***' by any listening device
  - **Not** just the **access** point
- **Unencrypted** messages can be easily copied
  - **WEP**: **Wired Equivalent Privacy**
    Deprecated (not recommended for use) – **one encryption** key for all **packets**
  - **WPA** & **WPA-2**: Wi-Fi Protected Access – uses a different *encryption* key for every data packet


Mobile Internet
Early mobile web-access limited by small displays and expensive data-transfer rates
WAP & WML – Wireless Access Protocol and Wireless Markup Language
Modern mobile internet on third and fourth generation networks (3g and 4g) provide higher bandwidth data rates
To broadband equivalent rates
Used in latest generations of Smartphones and for mobile internet access on laptops
Week 6 Networks
31

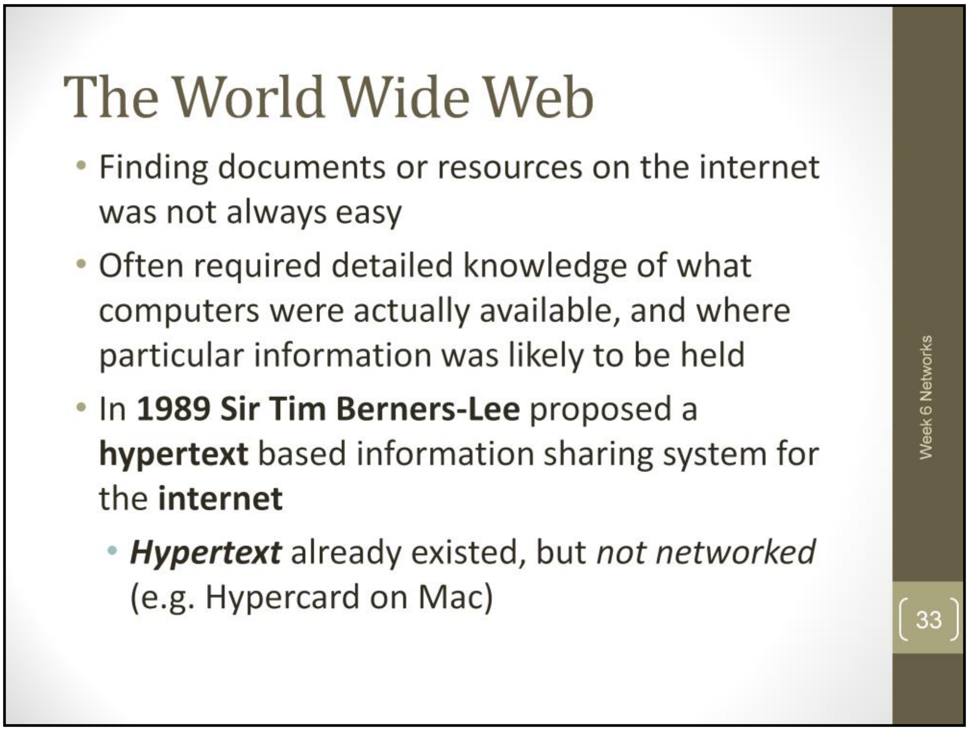
HCW p354-357

| Feature | iOS | Android | webOS | Windows Phone | BlackBerry OS | Bada |
|---|---|---|---|---|---|---|
| **Company** | Apple | Open Handset Alliance (Google) | HP/Palm Inc | Microsoft | RIM | Samsung |
| **OS Family** | Mac OS X/Unix-like | Linux | Linux | Windows CE 7 | Mobile OS | *nix |
| **CPU** | ARM | ARM, MIPS, Power Architecture, x86 | ARM | ARM | ARM | ARM |
| **Programmed in** | C, C++, Objective-C | C, C++, Java | C | Many, .NET (Silverlight/ XNA) | Java | C++ |
| **License** | Proprietary + open source elements | Open source + closed source elements | Open source + closed source modules | Proprietary | Proprietary | Proprietary |
| **Official SDK platform(s)** | Mac OS X | Multi-platform | Multi-platform | Windows | Windows | Windows |



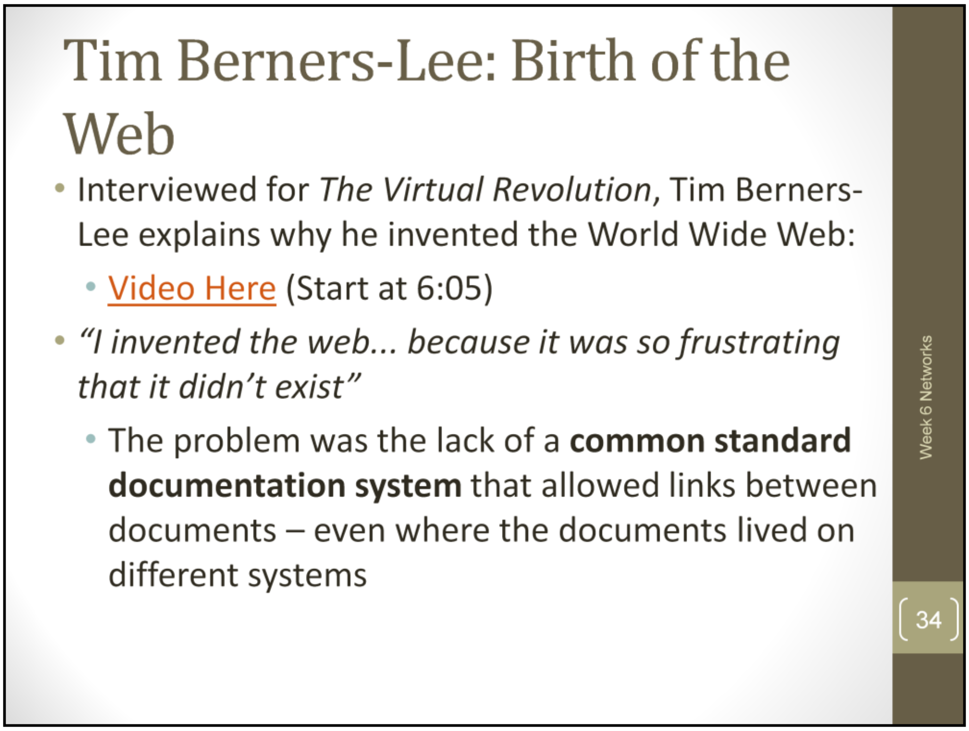

Table excerpted from http://en.wikipedia.org/wiki/Mobile_operating_system CC-BY-SA Wikimedia Foundation, multiple contributors

SDK platform refers to what hardware is used to develop for the phone, ` MAC OS X is the only official hardware that can be used to develop for iOS using the Software Development Kit (SDK)




1989: Tim Berners-Lee, a systems developer at CERN (Geneva) becomes frustrated at the difficulty of retrieving documents and information from different systems  - proposes a *hypertext* based system to be run on-top of the internet:

The World Wide Web




In the first six minutes of the video, Tim discusses how he got interested in computers more generally.

Perhaps of interest to this class as he followed computers as the technology developed: working first with simple relays and transistor circuits, then building his own simple computers out of ICs. Eventually working at CERN → The **European Organization for Nuclear Research** developing systems that would allow other (incompatible) systems to communicate and share data.

# A Brief History of the Web

- **1990**: First web-server, **http://info.cern.ch**
- **1993**: Mosaic Browser brought **graphical browsing** to **Mac** & **PC** computers
- **1995**: First edition of Bill Gates' book "The Road Ahead" doesn't mention the web at all
- **1999-2001**: DotCom Boom and Bust
- **Today**: The **Ubiquitous** Web



**Ubiquitous** Web → everywhere used for everything

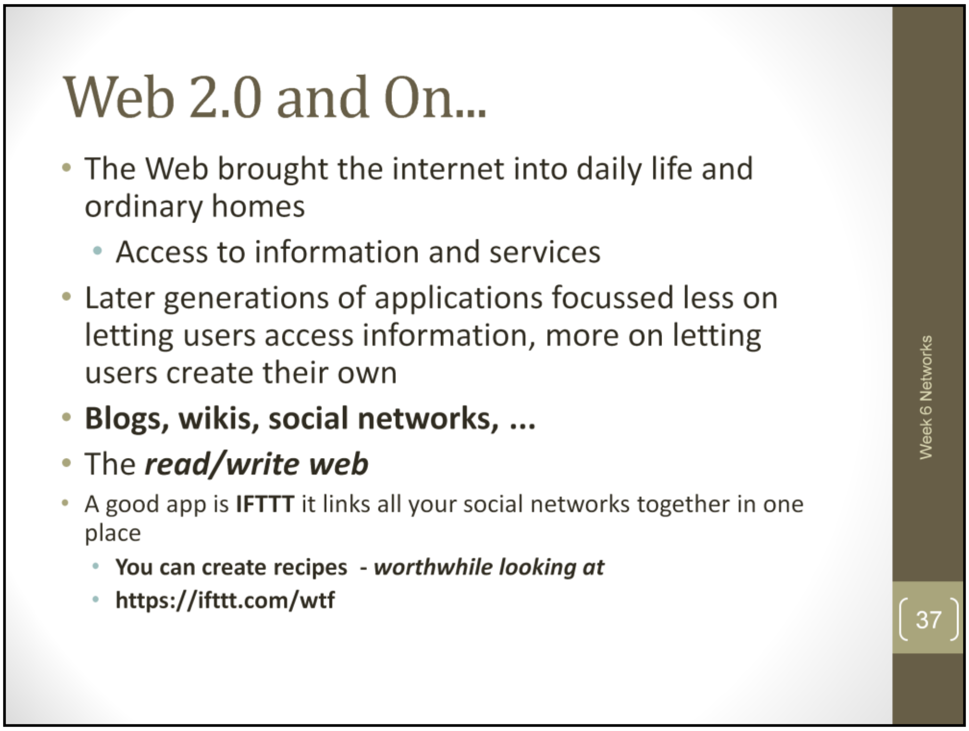
# Web Terms

- **HTTP**: **HyperText Transport Protocol** – the protocol used by browsers and servers to **request** and **provide** pages
- **HTML**: **HyperText Markup Language** – mark-up language used to **write** web-pages
- **URL**: **Universal Resource Locator** – each document or resource on the internet can have a **unique URL** that anyone to access that file

Week 6 Networks

36

HCW p 368




Good questions here are to ask who has a Facebook account, who uses Wikipedia, who has ever edited or created a page on Wikipedia, etc...

# Client/Server

- One machine on internet acts as a ***server***
- One or more machines as ***clients***
- Examples:
  - **Chat** or **Instant messaging**: User runs client program, which connects to a server run by **AOL/Microsoft**, etc
  - Some multi-user games require one user to run as a *server* (dedicated or as *client* as well)
  - **MMO** games (**World** Of **Warcraft**)

Week 6 Networks

38

World of Warcraft server.

# Peer To Peer

- Made (in)famous by **Napster** & illegal file-sharing
- Many legitimate uses
- Many **P2P** applications use a subset of **peers** to act as **directories** to allow peers to identify resources or other peers on the network
  - E.g. **Skype supernodes**
  - **BitTorrent** peers can be both ***seeders*** **(providers of files)** and ***trackers*** **(directories)**

Week 6 Networks

39

See HCW p340-343

Napster made this infamous with illegal file sharing.

# Distributed Server Networks

- Some applications may use maintain *client/server distinction* – but use distributed networks of servers
- E.g. **OpenSim Hypergrid**
  - **OpenSim** is an open source **3D virtual** world platform: anyone can run a server to create their own small virtual world
  - **OpenSim** virtual worlds can be linked
  - *Users* can travel between *worlds* from a *single client*

Week 6 Networks

40

**OpenSim Hypergrid → open source 3D virtual world.**

# Quick quiz

- Join the ‘Socrative’ app ‘Room 642124’ and try the quick quiz.

# Further Reading

- Wikipedia:
  - History of the Internet, TCP/IP, DSL, History of the World Wide Web, etc.
- PCH:
  - Chapter 14, Computer Communications

# Required Reading For Next Week

- HCW:
  - Chapter 9: p132-143 (Bitmaps & Vector Images)
  - Chapter 19 (Digital Cameras)
  - Chapter 20 (Multimedia Sound)
  - Chapter 21 (Multimedia Video)
  - Chapter 29 (Internet Video/Audio)

This work by Daniel Livingstone at the University of the West of Scotland is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/ or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

**Image Credits:**

Title Image – CC-BY-SA Daniel Livingstone, 2011

World of Warcraft screenshot – CC-BY Juanpol, http://www.flickr.com/photos/juanpol/8232592/

Mobile phone GPS – CC-BY HawaiianMama http://commons.wikimedia.org/wiki/File:GPS_on_smartphone_cycling.JPG

Ethernet RJ-45 Connector – CC-BY-SA David Monniaux , http://en.wikipedia.org/wiki/File:Ethernet_RJ45_connector_p1160054.jpg

Network Topology: Bus, Public domain, from Wikimedia Commons, Star CC-BY-SA Head, http://commons.wikimedia.org/wiki/File:Netzwerktopologie_Stern.png

Network Topology/DataFlow CC-BY-SA Kbrose/Cburnett http://en.wikipedia.org/wiki/File:IP_stack_connections.svg

OSBridge WiFI access point, Public Domain by Rewt, http://en.wikipedia.org/wiki/File:3GN_.jpg

Client/Server network, Gnu LGPL by Maurio Bieg, http://commons.wikimedia.org/wiki/File:Server-based-network.svg

Peer to Peer Network, Public Domain by Maurio Bieg http://commons.wikimedia.org/wiki/File:P2P-network.svg

Telephone mast, CC-BY-SA Andrew Tatlow, http://commons.wikimedia.org/wiki/File:Mobile_telephone_mast_-_geograph.org.uk_-_165771.jpg

Smartphone market share, CC-BY-SA Eraserhead1, http://en.wikipedia.org/wiki/File:Smartphone_share_current.png

Product names, logos, brands, and other trademarks featured or referred to within the these notes are the property of their respective trademark holders. These notes have been produced without affiliation, sponsorship or endorsement from the trademark holders.