COMPSCI 314 S2C

# Modern Data Communications

- Introduction
- Basic concepts
- Assignment 1



## COMPSCI 314 S2 C 2010
## Modern Data Communications

Lecturers

- Brian Carpenter – Room 587, brian@cs.auckland.ac.nz
- Clark Thomborson - Room 593, cthombor@cs.auckland.ac.nz
- Nevil Brownlee – Room 590, n.brownlee@auckland.ac.nz

Test Date

Friday 27 August, 9:00 – 10:00 am

Assignments due

*(via the CS DropBox, dates subject to revision)*

Friday 20 August
Friday 24 September
Friday 15 October

Course web pages - keep an eye on them!

http://www.cs.auckland.ac.nz/courses/compsci314s2c/



## Other matters

- Class representative
  Your name could be here: ________________

- Tutor
  Habib Naderi

- Assignment extensions
  - We will consider extensions to an assignment due date only for
    1. Illness or other unforeseeable emergency
    2. Conflicts with other assignments, if the request is made within *one* week of the assignment being issued
  - We will not be sympathetic if told about conflicts at the last minute. Please plan your work.



## There are no stupid questions...


. . . only stupid answers

- Questions

  If you're stuck or don't understand, please contact the tutor or any of your lecturers.

  Or consider using the class forum.

- Email

  Email must include the course number (314) and your UPI.

  A name is nice, too.

# Approach to material

- We (mostly) follow the textbook (Shay, 3rd edition)
- The lectures will provide in-depth discussion and comment on the course material.
  - Students who miss the lectures tend to get lower grades
  - You should also read the relevant sections in the textbook
- The course does *not* cover *all* of the textbook. The sections that are covered are shown on the lecture outline, as it appears on the *lectures* page linked off the *course web page*
- *Changes* to the course outline and/or content will be notified on the course web page
- We assume that students already have some understanding of Data Communications.
  - If not, you may need to read ahead in Shay
  - Assignment 1 is designed to bring you up to speed



# Approximate plan of course

| Week starting | Tuesday **Lib B15** | Thursday **Lib B15** | Friday **Eng1 401** | |
|---|---|---|---|---|
| 19 July 2010 | 1 Introduction | 2 Signals | 3 Codes | |
| 26 July 2010 | 4 Analog & Digital | 5 Analog & Digital | 6 Compression | |
| 2 Aug 2010 | 7 Compression | 8 Data Integrity | 9 Data Integrity | |
| 9 Aug 2010 | 10 Data Integrity | 11Encryption | 12 Authentication | |
| 16 Aug 2010 | 13 Flow Control | 14 Flow Control | 15 LAN link control | ***Ass 1 due 20 Aug*** |
| 23 Aug 2010 | 16 Ethernet | 17 Ethernet | - Test - | ***TEST: Fri 27 Aug*** |
| 30 Aug 2010<br>6 Sept 2010 | — Mid Semester Break — | | | |
| 13 Sept 2010 | 18 Wireless, 802.11 | 19 Bridges | 20 Switches | |
| 20 Sept 2010 | 21 Routing | 22 Routing | 23 IPv4 basics | ***Ass 2 due 24 Sept*** |
| 27 Sept 2010 | 24 IPv4 and DNS | 25 IPv6 | 26 TCP | |
| 4 Oct 2010 | 27 UDP, sockets | 28 Applications | 29 P2P applications | |
| 11 Oct 2010 | 30 HTTP, the web | 31 Loose ends | 32 Course overview | ***Ass 3 due 15 Oct*** |
| 18 Oct 2010 | *No lectures – just lots of time to study* | | | |

*Changes are possible at any time - check the web site !*



# Main objectives

- Explain how signals and data are sent over various media.
  - You need this to understand how networks are constructed and made reliable and secure.
- Introduce the principles of Local Area Networks.
  - You need this to understand how large-scale networks, like the Internet, are built up from LANs.
- Explain how network protocols, especially TCP/IP, are designed and fitted together
  - You need this to understand the basis for the Internet and for distributed computing.
- ➔ Our focus is on *how things work*, especially on the underlying principles and protocols – we won't look at 'how to configure a router,' etc.

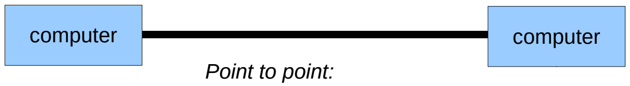


# Overview

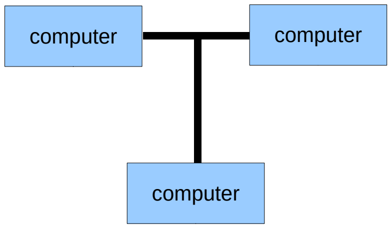
*See Chapter 1 of Shay*

- Data communications are usually implemented using various layers in a protocol stack
  - Each software layer will be a set of data structures, processes and procedures implementing specific network protocols
  - Hardware layers are similar, but made out of electronic or optical components and circuits
  - A protocol is a set of rules about how to send bits
- The need for layers arises from the structure of networks and the needs of applications.

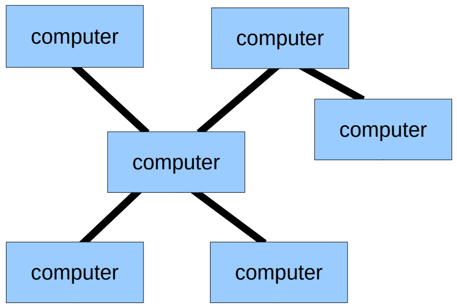
# Network types




*Point to point:*
*How to send bits?*




*Shared link:*
*How to send bits to the correct destination?*




*Mesh:*
*How to send bits indirectly to the correct destination?*



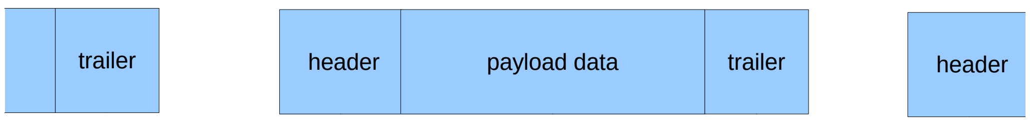
# Addressing

- As soon as we have more than two devices talking to each other, we need to have an address for each device.
  - Where devices are on the same cabling system or wireless system, usually a *Local Area Network*, the addresses are called *physical, hardware* or *link-level* addresses.
  - Where devices are indirectly connected, the addresses are logical addresses usually called *network* addresses.



# Packets

- Data flows over the network in *packets.*
  - Packets are sometimes called *datagrams*.
  - Packets are known as *frames* when considering the physical layer.
  - There are gaps between the frames.
  - Packets allow many users to share one network
  - Packets include *headers* and optional *trailers* as well as useful data (called the *payload*).
  - Headers include source and destination *addresses*.




# Things to be done in a network

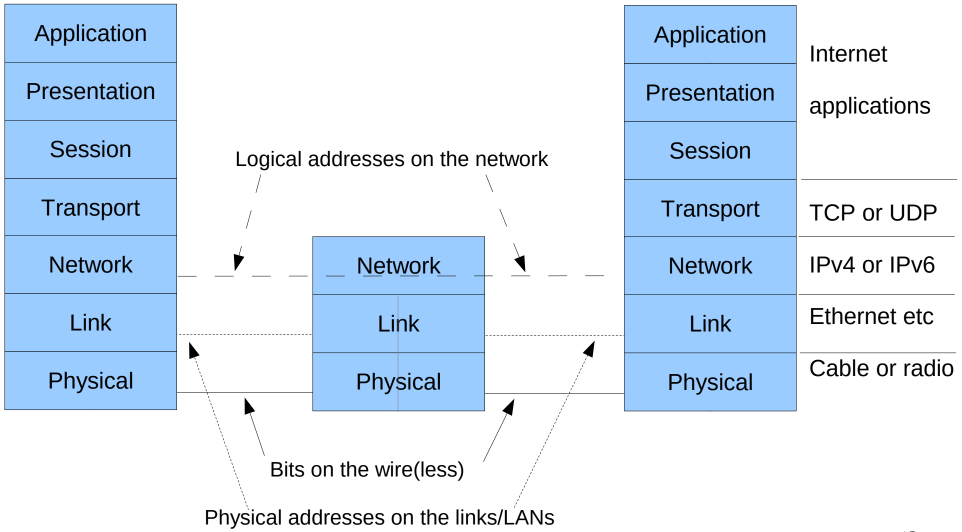
1. Transmit bits from one place to another (Physical)
2. Assemble bits into bytes and messages, check for reliable transmission, deliver to correct destination (Link)
3. Send messages indirectly between end-nodes in mesh-type network (Network)
4. In a mesh network, handle lost packets, broken links etc (Transport)
5. Handle extended sessions between endpoints, LANs, etc.
6. Resolve differences between data representation in different computers
7. Do something useful (User application)

These are the seven layers of the “Open Systems Interconnection” (OSI) communications model.

TCP/IP (Internet) combines layers 5-7, into a single Application layer.

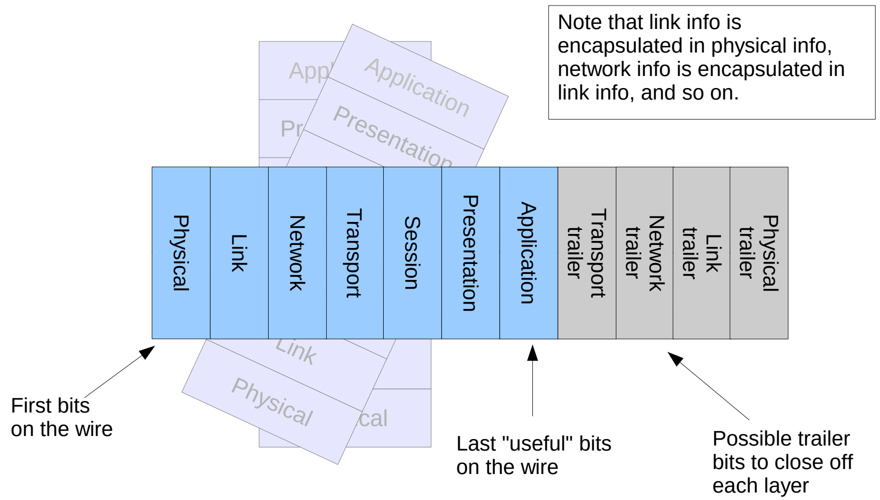
We focus on the Internet protocols.

# The formal model






# Data packets on the wire (or on a radio link)






# A real packet that you might see

| Ethernet preamble | Ethernet link header | IP header | TCP header | HTTP header+data | Ethernet trailer |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5,6,7 | ...2,1 |

(most layers don't require trailers in practice)

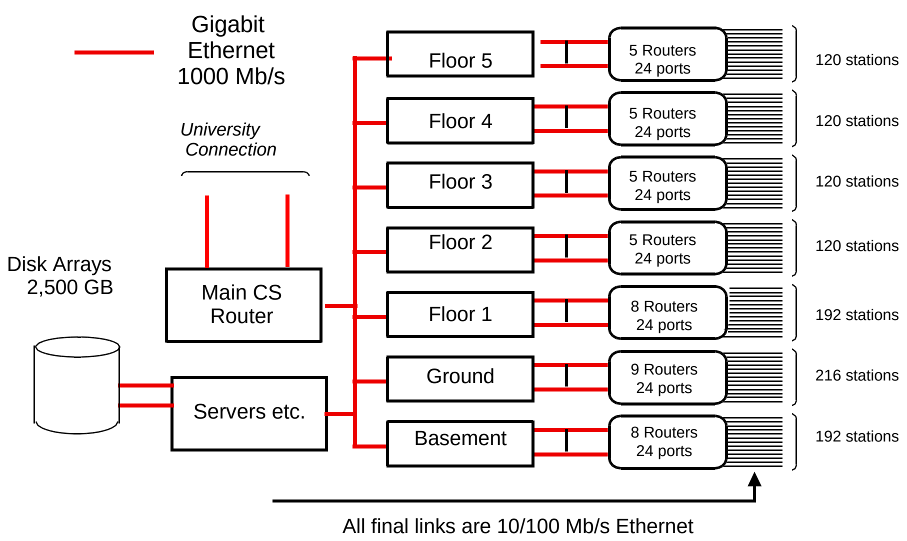
- Ethernet physical layer encapsulates
  - Ethernet link layer encapsulates
    - IP encapsulates
      - TCP encapsulates
        - HTTP



# A real network

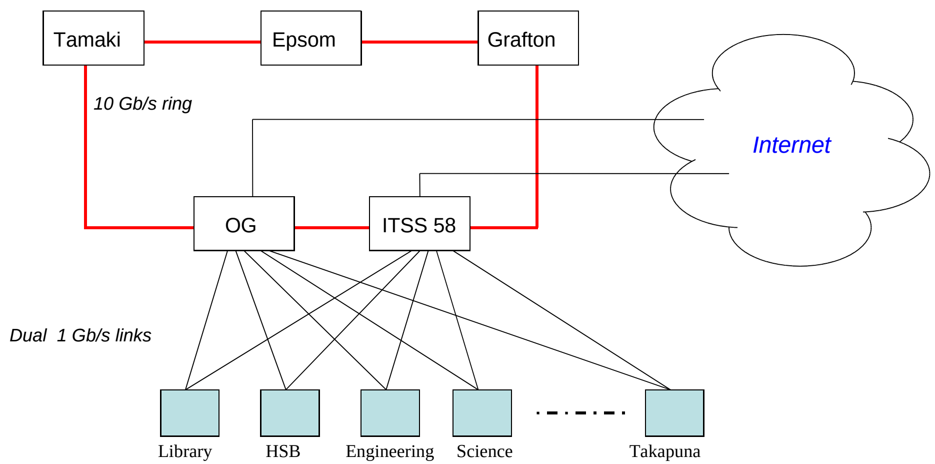
- Here's a (very brief) overview of the University of Auckland network, as it was a few years ago.
- Details change, but the principle is the same.

## Simple view of Computer Science Network, 2003






# The U Auckland Network, early 2007






# Communications basics

*See Shay 2.1-2.3*

- Data is sent from / received by an *interface* on a device (e.g. a PC)
- It may be sent directly, using *baseband* transmission, or it may be mixed with a carrier signal, i.e. sent using *modulated* transmission
- The time taken to transmit one bit ( '0' or '1') is called the *bit cell period*. Within each such period, a receiver must decide whether the incoming bit is '1' or '0'



## Basic information on transmission of bits

Bits, sent as electrical or optical signals, always travel at a 'propagation speed' of

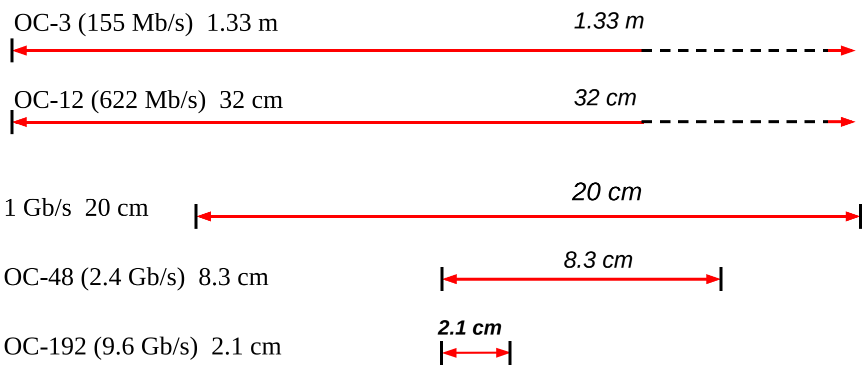
- 300,000 km/s in "free space" (radio, satellites, etc) (30cm per nanosecond)
- About 200,000 km/s on copper or fibre-optic cables (20cm per nanosecond)
  - Light, radio, and signals on a wire are all types of electromagnetic wave, so these speeds are in fact the speed of light in each case.

A 'faster' link has the bits arriving *more often* (say 1000 per microsecond, rather than 100 per microsecond), but they *never travel any faster*.

- The distance from New Zealand to North America, South America, Japan or Singapore is close to 10,000 km.
- 10,000 km at 200,000 km/sec takes 1/20 second = 50 milliseconds.
- The delay or "latency" from New Zealand to almost anywhere except Australia is therefore at least 50 ms. This delay cannot be reduced!
- Communications satellites orbit at 30,000 km, so the latency via satellite is 60,000 km at 300,000 km/sec = 1/5 second = 200 msec.

# Distances between bits, on optical fibre

Assume propagation speed of 200,000 km/s in glass fibre
(These distances are nearly correct if the page is printed on A4 paper).






# Communication Media: Conductive Metal

- Co-axial Cable (2.3)
  - Centre conductor, surrounded by a metal screen
  - Signal carried by the centre conductor, screened from electrical *noise*
- Twisted Pair (2.2)
  - Carries *balanced* signals, so as to minimise electrical *noise*
  - Cheaper and easier to install and use than co-ax
  - UTP cable has 4 pairs in an outer covering
  - Cat (Category) 5 UTP used for 100 Mb/s, cat 6 for 1 Gb/s



# Communication Media: Optical Fibre (2.3)

- Uses thin (about 50 micron) glass fibre to carry pulses of light
- Fibre is either *graded index* or *step index,* restricting the light's *propagation mode* so as to confine it inside the fibre
- Attenuation in fibre is low, making it suitable for long-haul (70 km or more) links
- Submarine cables can use optical amplifiers. For example, Southern Cross connects Sydney-Auckland-Fiji-Honolulu-Los Angeles
- Immune to electrical noise



# Communication Media: Wireless (2.4)

- Use electromagnetic waves to carry the signal in air (terrestrial) or free space (satellite)
- Wireless LANs (*802.11*) commonly used to link laptop PCs to an Internet *access point*
  - Range usually inside a room or building, say 50m
  - One access point can handle many laptops
- 802.11 can be used (with directional antennas) for much longer hops, so as to form regional networks
- Bluetooth used to link devices without wires
  - Cell 'phone to laptop, mouse to PC
  - Range about 10m or less

# Assignment 1

- Due August 20 and you can start immediately
    - worth 5% of the final grade
- Understand the basic concepts of protocol layers and encapsulation by practical observation in the CS lab.
- Read and understand the tutorial document (13 pages) linked as "support material" at http://www.cs.auckland.ac.nz/courses/compsci314s2c/assignments/
- Learn to use the packet capture tools provided.
- Capture data for each part of the assignment, and analyse the results to answer the questions.
    - don't forget to include captured data in your PDF file



# Background for Assignment 1

- Revise the material about TCP/IP from CompSci 215, or look ahead in Shay Chapters 9, 10 and 11.
- Packets include a sequence of *headers* corresponding to protocol layers.
- If we can inspect the packets, we can see and understand the headers.
- The CS lab machines have some tools for 314 students to allow capture and inspection of your packets.
    - capture packets to and from your IP address only, for privacy reasons



# Tools installed on CS lab Windows

- wincap - a simple library supporting the windump command line utility.
- wireshark - a packet capture tool with a GUI.
- Using the tutorial, try out some packet captures like the examples in sections VI and VII.
- Then you should be ready to start the assignment.