


Week 9 - Friday

## Introduction to Networks


School of Information Technology and Electrical Engineering  
The University of Queensland



## Outline

- Networks
  - Network Structures/Topologies
  - Protocols and Services
  - Internets and The Internet
  - Introduction to IP/UDP/TCP
- Credits:
  - Tanenbaum, "Computer Networks"
  - Bryant and O'Halloran, "Computer Systems: A Programmer's Perspective"


2



## Why Use Computer Networks?

- Resource sharing
  - software, data, databases, equipment remotely accessible
- High Reliability
  - Alternative sources of supply
- Economy
  - Small computers have better price/performance than large computers
- Communication
  - Email, WWW, Instant Messaging, Chat
- Personal reasons
  - Access to information, entertainment
- Commerce
  - Business-to-business, business-to-consumer


4



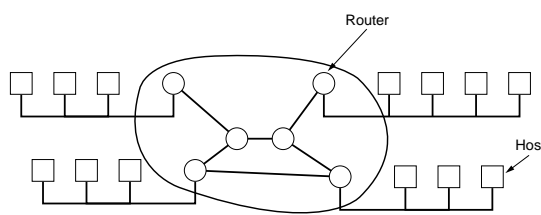
## Connectivity

- Networks provides connectivity between **nodes** over some **link** (or links)
- **Nodes**
  - hosts (computation nodes)
    - e.g. computers and other devices
  - routers (switching nodes)
- **Link: physical medium**
  - Various technologies, e.g. twisted pair wire, air
  - Various scales, e.g. room to planet sized
  - Various topologies
    - Point to point
    - Multiple access (broadcast)


5



## Network Structure – Routers and Hosts



6



## Scale of Computer Networks

Distance	Communication across...	
10 m	Room	} Local area network
100 m	Building	
1 km	Campus	
10 km	City	} Metropolitan area network
100 km	Country	} Wide area network
1,000 km	Continent	
10,000 km	Planet	
		} The Internet

7

COMP2303  
COMP7306

### Packet switched

- Networks can be divided into:
  - Circuit switched
  - Packet switched (store and forward)
- Most computer networks fall in the second category.
- Information is broken into packets which are moved through the network separately.

8

COMP2303  
COMP7306

### Point-to-Point Networks

- Pairs of nodes linked together - links have two ends
  - e.g. twisted pair cable, fibre-optic cable, microwave link
- Packets sent from origin router to destination router via intermediate routers
  - **Packet** = short message (with destination address)
  - Each router makes a decision about where to send a packet next
- Examples
  - Most WANS

9

COMP2303  
COMP7306

### Possible Point-to-Point Topologies

The diagrams illustrate six network topologies:
 

- Star**: A central node connected to multiple peripheral nodes.
- Ring**: A closed loop of nodes connected in a single path.
- Tree**: A hierarchical structure with a root node and multiple levels of child nodes.
- Complete**: Every node is connected to every other node in the network.
- Intersecting Rings**: Two or more rings that share one or more nodes.
- Irregular**: A network with no regular pattern of connections.

10

COMP2303  
COMP7306

### Multiple Access (Broadcast) Networks

- Single channel shared by all hosts
- Each host checks each packet's destination address
  - reject packets where address doesn't match
- Broadcast subnets also support **broadcast mode**
  - packet addressed to all destinations
- **Multicasting**
  - packets broadcast to a particular subset of hosts
- Examples
  - Satellite based networks
  - WiFi, bluetooth(?)

11

COMP2303  
COMP7306

### Broadcast Network Topologies

The diagrams illustrate four broadcast network topologies:
 

- Bus**: A single line (bus) with multiple computers connected to it.
- Ring**: A closed loop of computers connected in a single path.
- Satellite**: A satellite in orbit connected to multiple ground stations.
- Radio**: A radio tower connected to multiple receivers.

- Issues
  - Which host broadcasts?
  - If only one, which one?
    - Static or dynamic allocation

12

COMP2303  
COMP7306

### ethernet

- IEEE - 802.3
- Broadcast network.
- Stages (not from the beginning):
  - BNC connectors, T pieces – coaxial cable. All nodes strung together in a line.
  - Twisted pair cable (eg cat5), RJ-45 plugs – hubs
    - Link detection, still one segment
  - Switches

13

COMP2303  
COMP7306

## Networks of Networks

- An *internetwork* (**internet**) is an interconnected set of networks.
  - The Global IP **Internet** (uppercase "I") is the most famous example of an internet (lowercase "i")

14

COMP2303  
COMP7306

## Example Structure Revisited

15

COMP2303  
COMP7306

## Short Break

- Stand up and stretch

16

COMP2303  
COMP7306

## Protocols

- Definition: Set of procedures designed to achieve some higher purpose, sometimes in adverse circumstances
- In communications:
  - Set of rules governing exchange of data between two entities
  - Agreed rules of communication.
    - How to say things
    - What to expect when you do.

17

COMP2303  
COMP7306

## Key Elements of a Protocol

What's in a protocol?

- **Syntax**
  - Data format
  - Signal levels
- **Semantics**
  - Meaning of data, e.g. control information
- **Timing**
  - Speed matching, sequencing

18

COMP2303  
COMP7306

## Protocol Hierarchies

- To reduce design complexity, most networks organized as series of layers or levels
- Purpose of layer
  - Offer **services** to higher layers
  - Abstraction
    - Shield higher layers from implementation details

19

COMP2303  
COMP7306

## Network Architecture

- Definition: A set of layers and protocols
- Must contain enough information to build
- Some protocol architectures do not specify interfaces, only protocols
  - In such cases interfaces can differ on different machines
- **Protocol Stack**
  - Set of protocols used by a certain system, one protocol per layer

20

COMP2303  
COMP7306

## Entities

- Definition: Active elements in each layer
  - Can be software (e.g. library code or separate process)
  - Can be hardware (e.g. integrated circuit)
- **Peer entities**
  - Same layer – different machines
  - Sometimes called *peer processes*

21

COMP2303  
COMP7306

## Services vs Protocols

Distinction is important

- Service (interface)
  - Operations available to a higher level entity
- Protocol
  - Rules for communication between peer entities

22

COMP2303  
COMP7306

## Services vs Protocols

23

COMP2303  
COMP7306

## Service Categories

- Connection
  - Connection-oriented
  - Connectionless
- Reliability
  - Reliable
  - Unreliable

24

COMP2303  
COMP7306

## Connection-Oriented and Connectionless Service

- Connection-oriented
  - Modeled after telephone system
  - Service user
    - establishes connection
    - uses connection
    - releases connection
- Connectionless
  - Modeled after postal system
  - Each message
    - contains full destination address
    - routed through system independently

25

COMP2303  
COMP7306

## Reliability

- Services can be reliable or unreliable
- Reliable
  - receiver needs to acknowledge receipt
- Unreliable
  - No acknowledgement necessary

26

COMP2303  
COMP7306

## Virtual Protocols

- No data directly transferred between peer entities
- Each layer passes data and control information to layer below, until reach lowest layer
- Actual communication occurs via physical medium

27

COMP2303  
COMP7306

## Headers and Enveloping

- Header
  - Additional information attached to message data by a protocol
- Protocol worries about header, not message
- Like placing a message in an envelope
- Layered protocols -> nested envelopes
- Some protocols might add a *tail* also

28

COMP2303  
COMP7306

## Headers and Enveloping (cont.)

29

COMP2303  
COMP7306

## The Notion of an internet Protocol

- How is it possible to send bits across incompatible LANs and WANs?
- Solution: *protocol software* running on each host and router smoothes out the differences between the different networks.
- Implements an *internet protocol* (i.e., set of rules) that governs how hosts and routers should cooperate when they transfer data from network to network.

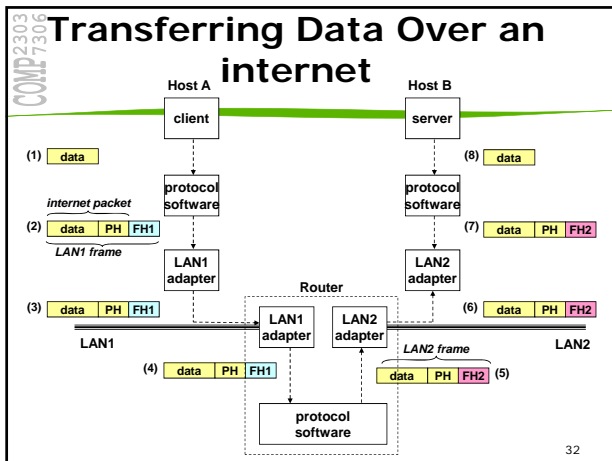
30

COMP2303  
COMP7306

## What Does an internet Protocol Do?

1. Provides a naming scheme
  - An internet protocol defines a uniform format for **host addresses**.
  - Each host (and router) is assigned at least one of these internet addresses that uniquely identifies it.
2. Provides a delivery mechanism
  - An internet protocol defines a standard transfer unit (*packet*)
  - Packet consists of *header* and *payload*
    - Header: contains info such as packet size, source and destination addresses.
    - Payload: contains data bits sent from source host.

31



- ### Other Issues
- We are glossing over a number of important questions:
    - What if different networks have different maximum frame sizes? (segmentation)
    - How do routers know where to forward frames?
    - How are routers informed when the network topology changes?
    - What if packets get lost?
  - Not topics for COMP2303/7306
    - More details in COMS3200 (Computer Networks I)
- 33

- ### Global IP Internet
- Most famous example of an internet
  - Based on the TCP/IP protocol family
    - **IP** (Internet protocol)
      - Provides basic naming scheme and unreliable delivery capability of packets (datagrams) from host-to-host
    - **UDP** (Unreliable Datagram Protocol)
      - Uses IP to provide unreliable datagram delivery from process-to-process
    - **TCP** (Transmission Control Protocol)
      - Uses IP to provide reliable byte streams from process-to-process over connections
  - Accessed via a mix of Unix file I/O and functions from the **sockets interface** (next week)
- 34

### Break

35

### Useful UNIX

36

- ### IP (Internet Protocol)
- Connectionless model of delivery
  - **Best-effort** delivery – no guarantees of success
  - **Packets**
    - Also called **datagrams**
    - Consists of
      - Header
      - Body part ("text")
    - Header consists of
      - 20 byte fixed part
      - 0-40 byte optional part
- 37

COMP2303  
COMP7306

## Internet Protocol (IP) Header

Version	HLen	Service Type	Total length			
Identification		DF	MF	Fragment offset		
Time to live	<b>Protocol</b>		Header checksum			
Source address						
Destination address						
Options (0 or more words)						

- We're interested in the **highlighted** header fields

38

COMP2303  
COMP7306

## IP Header: Protocol

- 8 bits
- Range: 0 to 255
- Identifies higher level protocol to which packet should be passed, e.g.
  - 6 = TCP
  - 17 = UDP
- RFC 1700 – defines protocol numbers

Version	HLen	Service Type	Total length			
Identification		DF	MF	Fragment offset		
Time to live	<b>Protocol</b>		Header checksum			
Source address						
Destination address						
Options (0 or more words)						

39

COMP2303  
COMP7306

## IP Header: Source and Dest. Addresses

- 32 bits each
- Dest. address
  - Used to route packet to intended destination
- Source address
  - Destination can choose whether to receive
  - Destination knows whom to reply to

Version	HLen	Service Type	Total length			
Identification		DF	MF	Fragment offset		
Time to live	<b>Protocol</b>		Header checksum			
<b>Source address</b>						
<b>Destination address</b>						
Options (0 or more words)						

40

COMP2303  
COMP7306

## IP Addresses

- 32-bit IP addresses are often written in **dotted-decimal** notation
  - each of 4 bytes written in decimal
    - e.g. 130.102.2.15
  - This notation used for human consumption
- Some addresses have special meanings
  - e.g., broadcast to all hosts on a particular network
  - More details in week 12

41

COMP2303  
COMP7306

## What does IP give us?

- Best-effort packet delivery
  - No guarantees ...
- Packets may be
  - dropped
  - reordered
  - duplicated
  - size-limited
  - delayed

42

COMP2303  
COMP7306

## What do Applications Want?

- Not to have to worry about the underlying network (routers etc)
- Messages
  - transported from one host to another
  - guaranteed delivery
  - delivered in same order as sent
  - arbitrary size

43

COMP2303  
COMP7306

### What do Applications Want? (cont.)

44

COMP2303  
COMP7306

### Transport Protocols

- We'll look at two transport protocols
  - UDP = User Datagram Protocol
  - TCP = Transmission Control Protocol

45

COMP2303  
COMP7306

### UDP: User Datagram Protocol

- Connectionless Transport Protocol
- Allows applications to send datagrams to other applications
  - Basically an interface to IP
- No need to establish connection
- 8 byte header + data

46

COMP2303  
COMP7306

### Why would we want to use UDP?

- Don't care if packet lost
  - e.g. Streaming
- Small messages
  - No connection establishment overhead
- No congestion control
  - Can send things as fast as possible
- Simple implementation
  - No connection state information

47

COMP2303  
COMP7306

### UDP: Header

32 Bits	
Source port	Destination port
UDP length	UDP checksum

- Ports
  - Identify end-points within machines
    - e.g. processes
  - Source port is optional
- UDP Length
  - Includes header
- UDP Checksum
  - For checking if corruption occurs
  - Optional – store 0

48

COMP2303  
COMP7306

### Short Break

- Stand up and stretch

49

COMP2303  
COMP7306

## TCP: Transmission Control Protocol

- Connection-oriented
- Reliable (over unreliable IP internetwork)
- Byte-stream
  - Not message stream
- Full-duplex
- Point-to-point (end-to-end)
  - Not multicast or broadcast
- Each machine has a *TCP transport entity*
  - user process or part of kernel

50

COMP2303  
COMP7306

## TCP Responsibilities

- Sender
  - Accepts data streams from local processes
  - Breaks data into pieces < 64k bytes
  - Sends each piece as IP datagram
  - Time-out and retransmit if no acknowledgement received
- Receiver
  - IP datagrams containing TCP data are passed to TCP entity for reconstruction
  - Acknowledge receipt
  - Reassemble datagrams in proper sequence

51

COMP2303  
COMP7306

## TCP Service Model: Sockets

- Sender and Receiver create sockets
- Sockets have addresses (IDs)
  - IP address + **port number**
- Multiple connections possible through one socket
  - Connections identified by socket ID of each end

52

COMP2303  
COMP7306

## Port Numbers

- 16 bits: 0 – 65535
- Below 1024
  - Well known ports
  - Reserved for standard services, e.g.
    - 23 - Telnet
    - 21 - FTP
    - 80 - HTTP
  - Look in /etc/services on a UNIX box

53

COMP2303  
COMP7306

## An Explanation of Ports

- Figure to be drawn in class

54

COMP2303  
COMP7306

## TCP: Byte Stream

- Message boundaries are not preserved
- Example
  - Sending process writes four 512 byte chunks
  - Receiving process may get
    - one 2048 byte chunk (see below)
    - two 1024 byte chunks
    - many other possibilities
- TCP doesn't care what the bytes are

(a) (b) 55

COMP2303  
COMP7306

## TCP Protocol

- TCP entities exchange **segments**
  - Data + 20 byte header, including:
    - Source & destination port numbers
    - Sequence number (for reordering)
    - Acknowledgements
- Size limited by
  - IP packet size
  - Network – maximum transfer unit (MTU)
- Segments with zero data are valid

56

COMP2303  
COMP7306

## TCP Connections

- Identified by
  - Source Port
  - Source IP address
  - Destination Port
  - Destination IP address

57

COMP2303  
COMP7306

## TCP/UDP and IP

- Remember: IP underlies both TCP and UDP
- Figure to be drawn in class

58

COMP2303  
COMP7306

## Client-Server Model

- Most network applications are based on the **client-server model**:
  - A *server* process and one or more *client* processes
  - Server manages some *resource*
  - Server provides *service* by manipulating resource for clients

```

    graph LR
      Client((Client process)) -- "1. Client sends request" --> Server((Server process))
      Server --- Resource[(Resource)]
      Server -- "2. Server handles request" --> Resource
      Server -- "3. Server sends response" --> Client
      Client -- "4. Client handles response" --> Client
  
```

*Note: clients and servers are processes running on hosts (can be the same or different hosts).*

59

COMP2303  
COMP7306

## Hardware and Software Organisation of an Internet Application

```

    graph TD
      subgraph ClientHost [Internet client host]
        direction TB
        Client[Client] --- UI[Sockets interface system calls]
        UI --- TCP_IP[TCP/IP]
        TCP_IP --- HI[Hardware interface interrupts]
        HI --- NA[Network adapter]
        subgraph UserCode [User code]
            Client
        end
        subgraph KernelCode [Kernel code]
            TCP_IP
        end
        subgraph HardwareFirmware [Hardware and firmware]
            NA
        end
      end
      subgraph ServerHost [Internet server host]
        direction TB
        Server[Server] --- UI
        UI --- TCP_IP
        TCP_IP --- HI
        HI --- NA
        subgraph UserCode
            Server
        end
        subgraph KernelCode
            TCP_IP
        end
        subgraph HardwareFirmware
            NA
        end
      end
      ClientHost --- GI[Global IP Internet]
      ServerHost --- GI
  
```

60

COMP2303  
COMP7306

## Coming Up

- Week 10 – Friday
  - HTTP
- Week 11
  - Network Programming
- Week 12
  - Internet & Ethernet
- Week 13
  - Network Applications (Other protocols)

61