Imperial College London

Computer Networks and Distributed Systems

Part 1 - Introduction

Course 527 – Spring Term 2015-2016

Emil C Lupu & Daniele Sgandurra
e.c.lupu@imperial.ac.uk, d.sgandurra@imperial.ac.uk

Course Structure

1st half: Computer Networks - covers basic principles of networking through examples of real technology

2nd half: Distributed Systems – covers basic distributed systems architectures, remote (object) interactions, remote procedure calls, security

Whilst networks are concerned with communicating data from one endpoint to another (or several) distributed systems help us design systems whose components are hosted on different computers.

Course Structure

2 lectures + 1 tutorial each week

1 coursework

Electronic handouts available from CATE

Please ask questions!

Acknowledgements:

- Computer Neworks based on material by Peter Pietzuch, Dan Chalmers and Ian Harries
- Distributed Systems based on material by Morris Sloman

Course Attendance

B.Eng & M.Eng Electronic and Information Engineering
2nd year Required

B.Eng & M.Eng Mathematics and Computer Science
3rd Year Selective

M.Sc Computing Science Selective

Recommended Books

"Computer Networks", Andrew S. Tanenbaum, Prentice Hall, 2005 (5th Edition)

- Main reference and worth reading

"Distributed Systems: Concepts and Design", George Coulouris, Jean Dollimore, Tim Kindberg, Addison-Wesley, 2005 (5th Edition)

IEEE, IETF, ITU, OSI and W3C standards form basis of much of the material, but not designed as tutorials

Exam Questions

Not about low-level details

- Q: "What's the 50th bit in the IP packet header?"
- A: "It's the 'Don't Fragment' Flag"

But rather about principles and design trade-offs

- Q: "You need to design a transport layer for a network with the following characteristics… How would you do this?"
- A: "I'd use a reliable transport service, similar to TCP, because…"

Always explain your reasoning!

Some (simple) maths involved

Part 1: Computer Networks Introduction and Overview

Computer Networks Help us Answer Some Questions

How do I get bits down a wire?

How many computers can be connected to an Ethernet LAN?

How do we provide network connectivity to a laptop that moves about?

Why does it matter whether I use `java.net.Socket` or `java.net.DatagramSocket` when programming?

How do Windows PCs, Linux boxes and Macs communicate on the Internet?

Why is the Internet sometimes so slow?

Syllabus Overview

Introduce networking concepts and terminology
- Introduce OSI and TCP/IP engineering models
- Course loosely follows OSI Reference Model

Describe basic network standards and protocols
- Learn how design choices affect network behaviour

Describe how networks inter-connect

Illustrate how networks interact with applications

Part 1 – Contents

Basic terminology

Network types

Network topologies

Network protocol standards
- OSI Seven Layer Network Model
- TCP/IP Internet Model

Information and Data

Information
- Stimuli that have meaning in some context for receiver

Data
- Information translated into form more convenient to move or process by computer

Channel
- Path through which signals can flow

Network
- Graph of devices interconnected by channels

Node
- Device on network graph
- May refer to end-point (e.g. computer) or communications device (e.g. router)

Network Metrics

Bandwidth
- Data transferred per unit time (usually bits/second)

Delay or **Latency**
- Time a bit takes to get from source to destination (seconds)

Jitter
- Variation in delay (usually % of delay or value +/- seconds)

Loss
- Rate of loss of units of transfer (percentage, unit depends on what is being lost)

Bandwidth

Careful! Bandwidth also technical (EE) term

- Measure of frequency range of analogue channel

(Informally) used for **channel capacity**

- How much data can be sent through a channel?
- Refers to **transmission rate** (throughput)
- "This is a high bandwidth connection."

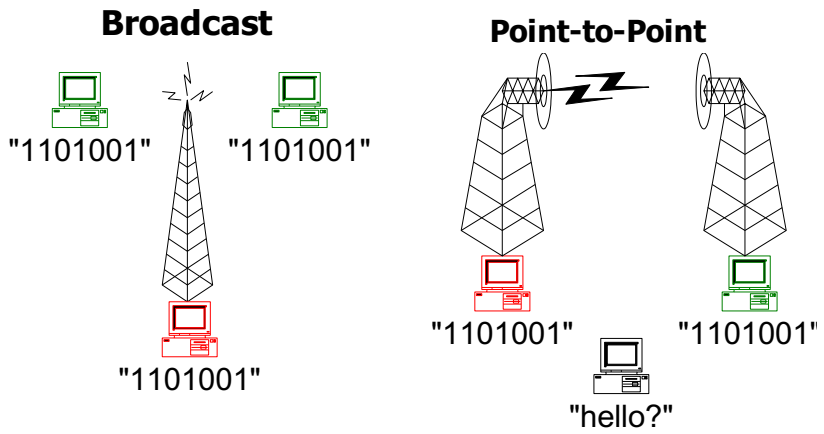
Classes of Communication

Many ways to describe a network

- Wires (or media) that form channels
- Behaviour of channels
- Range in physical and organisational terms
- Needs and capabilities of nodes

We need models to describe diverse networks

Types of Connections



From Connections to Networks

Most networks have >2 devices that connect dynamically

Individual wires between each pair of computers

- Simple but clearly not scalable

Shared wires between computers

- Only listen to messages addressed to you

Larger networks by having switches make dynamic connections over shared pool of channels

We'll come back to this later…

Types of Networks

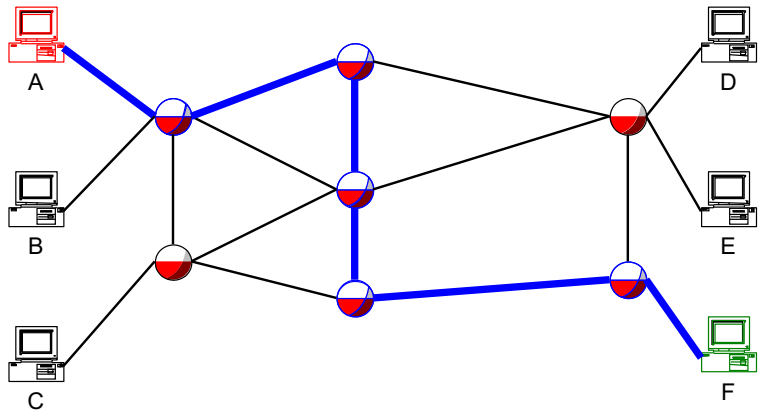
We now examine:

- Two forms of switch operation for networks
- Two types of service that networks can provide

Each valid but offer different behaviour

- Compare telephone network vs. computer network

Circuit Switching (CS)



Circuit Switching Features

One maintained path (**circuit**) (e.g. telephone call)

Three phases:

1. Circuit establishment
2. Data transfer
3. Circuit disconnection

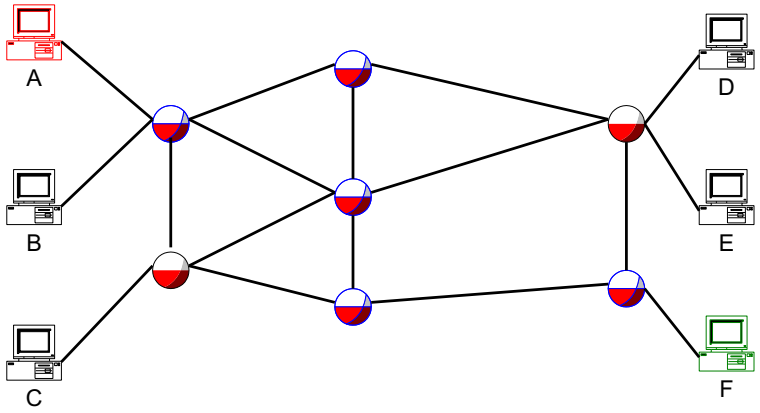
Overhead for call set-up, no overhead for use

Provides guaranteed resources

Connection breaks if any link or switch on route fails

Charging typically by time

Packet Switching (PS)



Packet Switching Features

Route calculated for each **packet** (e.g. postal service)
- Packets may arrive out of order
- Switches may store and forward packets

All data has addressing and control overhead
- But no initial overhead

Usually no guaranteed resources

Failures accommodated transparently
- Different routes may have different properties
- Packets may be lost/retransmitted due to failure

Charging typically by packet

Circuit Switching vs. Packet Switching

Circuit Switching
- Fixed bandwidth
- Unused bandwidth wasted
- Call set-up required
- Congestion may occur at call set-up (arrival rate = transmission rate)
- Overhead on call setup only
- In-order delivery
- Circuit fails if any link or switch fails

Packet Switching
- Variable bandwidth
- Uses only bandwidth required
- No call set-up
- Congestion may occur on any packet (causing delay and reordering)
- Overhead on every packet
- Out-of-order delivery
- New route found if any link or switch fails (some data may be lost)

Types of Connection Service

Network provides **connection service** to programs
- May be **connectionless** or **connection-oriented**

Uses underlying network to achieve this
- Network may be PS or CS
- Network doesn't determine service type provided
  - Software can add behaviour

Connectionless Service (CL)

No conceptual connection or maintained route
Unit of connection is **datagram** (packet)
No guarantee of order

Packet switched networks provide pure CL service
- Packets addressed by destination and routed accordingly
- Each packet handled separately
- No state at switches or set-up/tear-down calls

Connection-Oriented Service (CO)

Connection maintained between end-points

Unit of connection is the **circuit**

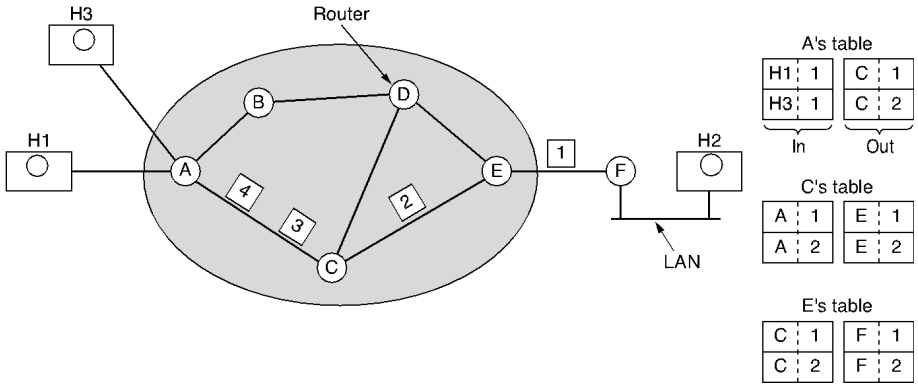
Order is preserved

Circuit switched networks provide pure CO service

- Circuit defines destination and route

Packet switched networks can provide CO service by using **virtual circuits**

Virtual Circuits I



Establish route using connection set-up packet

Soft connection using circuit

- Route packets by **circuit identifier**
- Each packet includes circuit identifier in short header

Virtual Circuits II

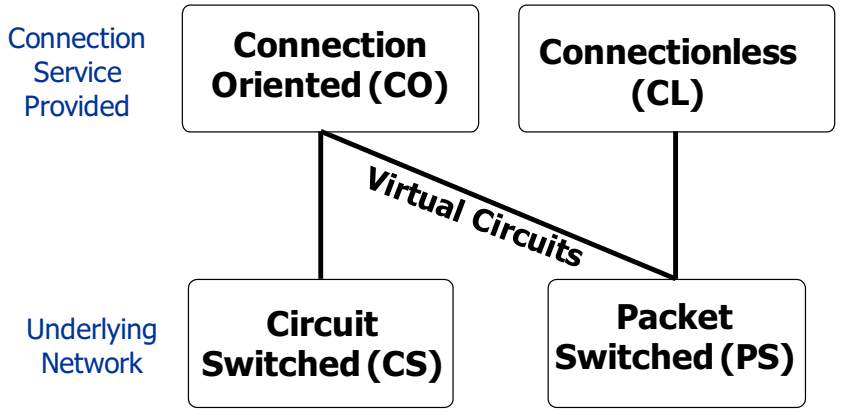
Less routing overhead per packet than PS

- Need to maintain circuit info at switches
- Set-up/tear-down overhead
- No dedicated resources but reservations may be possible

Order may be maintained (unlike CL)

Asynchronous Transfer Mode (ATM)

Summary: Classes of Network Connection



Scale of Networks

Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	Local area network
100 m	Building	
1 km	Campus	
10 km	City	Metropolitan area network
100 km	Country	Wide area network
1000 km	Continent	
10,000 km	Planet	The Internet

Local Area Networks (LANs)

Transmission through buildings
- Typically 80% of communications are local

Many and varied devices
- Different message sizes and rates
- Nodes may connect and disconnect, or fail
- Systems may compete or co-operate

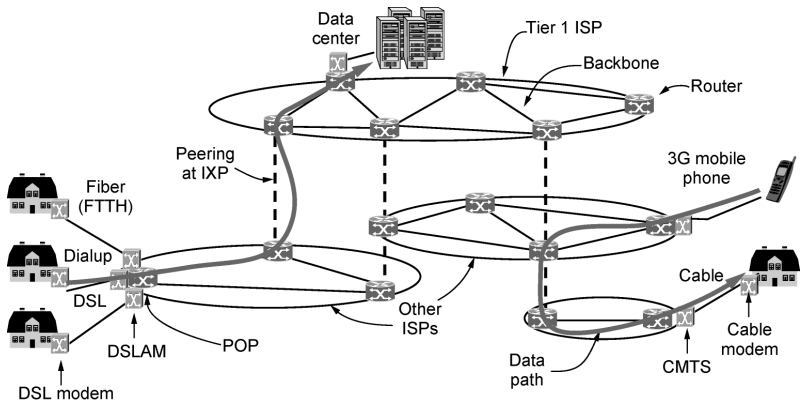
Typically under single administrative domain

Metropolitan, Wide-Area, Inter-nets

Formed from interconnected LANs
- Longer distances
- Costs of long cables, satellite links
- Delay and bandwidth restrictions due to distance

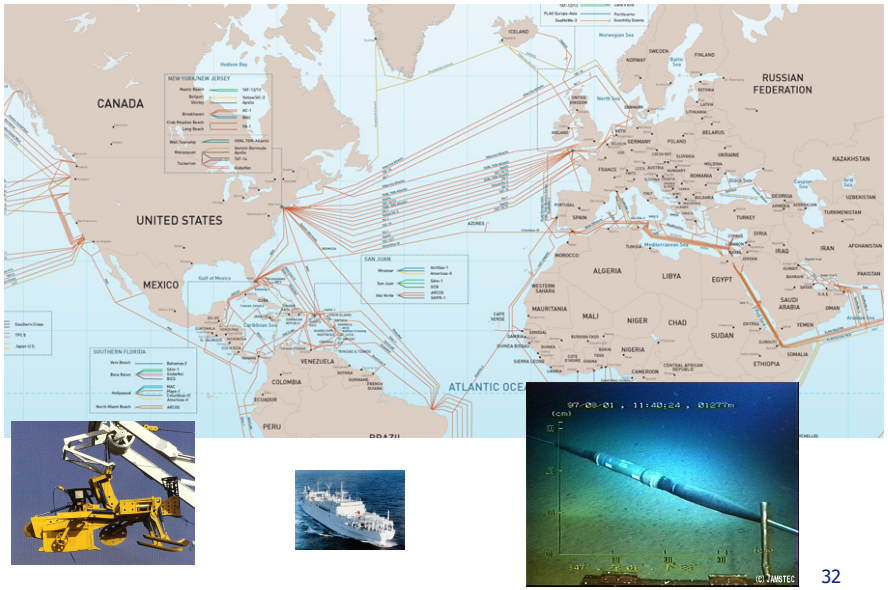
Politics of shared ownership and international connections

Overview of Internet Architecture

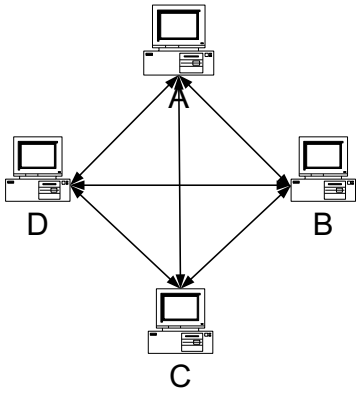


From: Tanenbaum and Wetherall, Pearson Ed. 2011

Internet Submarine Cables



LAN Topologies: Mesh



Fully connected graph

- Requires n(n-1)/2 links
- Doesn't scale to many nodes

No routing, as all nodes connected to each other

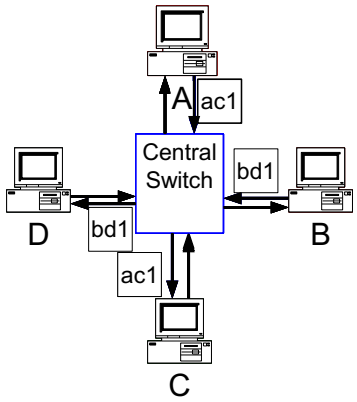
LAN Topologies: Star

Central switch

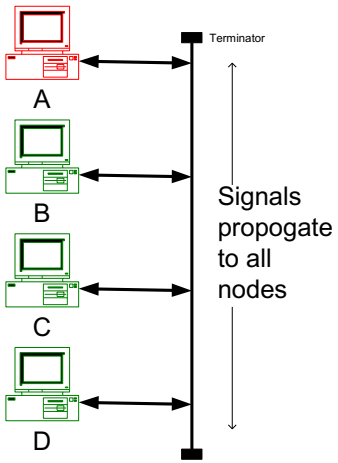
- Used by all nodes
- Links have simplicity of single communicating pair
- Switch must store/forward packets

Full bandwidth of link available to each node

- Assuming switch can keep up



LAN Topologies: Bus

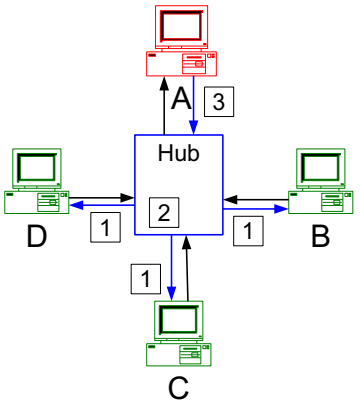


Bus is shared medium

- Divide data into frames (packets) to share link fairly
- Must address frames
- Must avoid and/or cope with frames colliding

Need **medium access control**

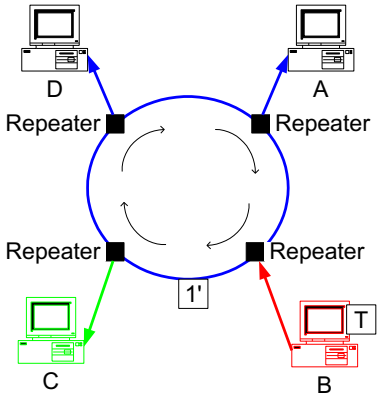
LAN Topologies: Bus with Hub



Central hub
- Used by all nodes
- Forwards all frames onto all outgoing links

Like bus with central connection
- Hubs can be formed into tree to extend bus size

LAN Topologies: Ring



Data circulates on ring in one direction
- Divide data into frames with addresses
- Permission to send given by permission **token T**
- Source node removes frame on return

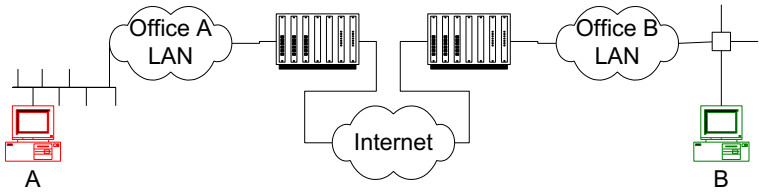
Comparing LAN Topologies

Switch
- Dedicates connections to communicating node
- Not as cheap as some alternatives

Ring
- Ring with tokens has simple equal use system

Bus/Hub
- Link bandwidth shared between all nodes
- Good for multicast/broadcast
- Require medium access control
- Require switched division to scale

Typical Wide-Area Connection Path



Subnets are smaller networks that form connection
- Subnets segment traffic
- Limit propagation of signals

Support for underlying networks
- With different physical technologies
- With different administrative ownerships

Network Abstractions

Applications view network as black box service
- Hide the details of the network
- Many parameters are orthogonal

☛ How do we describe a complete network architecture?

General-purpose networks are complex
- Different networking technologies
- Equipment provided by multiple manufacturers
- Managed by different people

☛ How do we define intended behaviour?

Standards

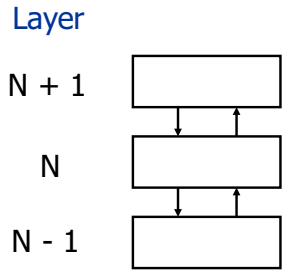
Standardised ways of connecting systems
- Hardware and software (protocol) standards
- Freeze technology and require backwards compatibility
- Do not prescribe implementation

Many standard bodies exist
- e.g. ISO, ITU, IEEE, IETF, W3C, …

Different types of standards
- Open (published, free) vs proprietary standards
- e.g. industry provides (de-facto) standards

Network Stack Model

Model network as **layered stack**
- Layer N provides well-defined service to Layer N+1
- Layer N uses Layer N–1 for communication



Layering provides modularity
- Layers do not process data from higher layers
- May replace implementation of layers

But too many layers lead to inefficiency

Design issues at a layer

Error Correction

Multiplexing

Addressing

Routing

Congestion

Flow Control

Quality of Service

Real Time

Security

… layers may deal with only some of these issues

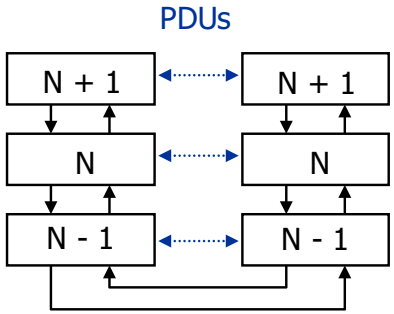
Protocols

Protocol: "an agreement between parties on how communication is to proceed"

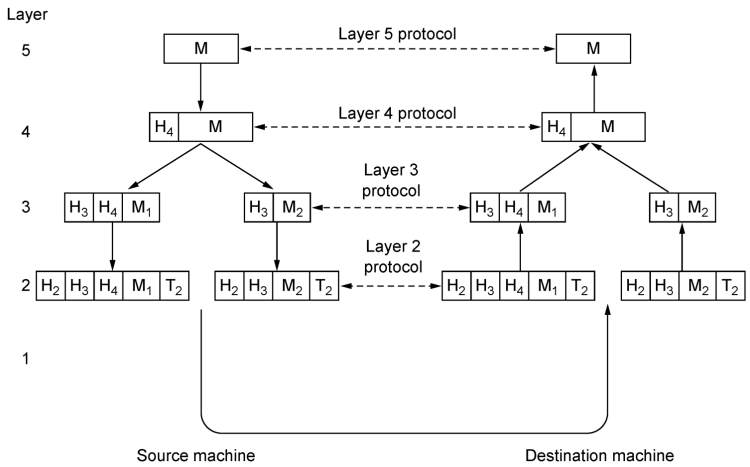
- Defines msg formats, relationships between msgs, …
- Reuse protocol implementations across apps

Entity at one host exchanges **protocol data units** (**PDU**) with peer entity at another host

- Actual connection only at lowest layer



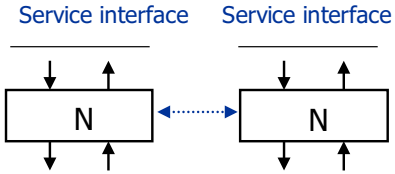
Protocols and Data Encapsulation

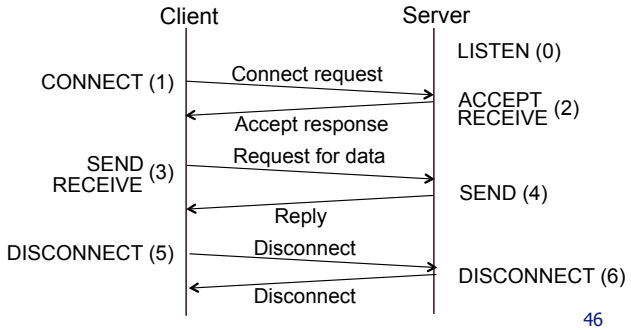


Example information flow supporting communication in layer 5.

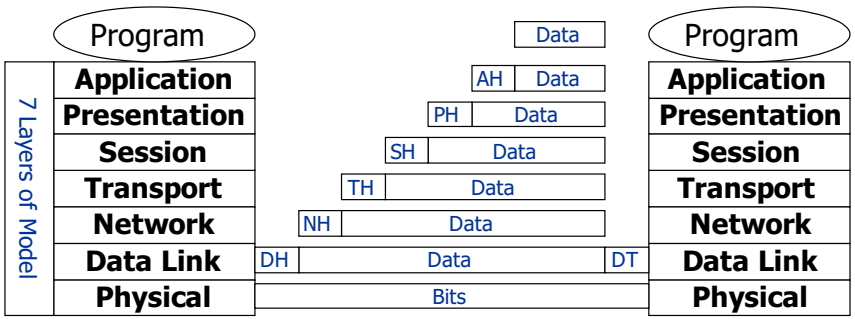
Protocols and Service Primitives

Primitive	Meaning
LISTEN	Block waiting for an incoming connection
CONNECT	Establish a connection with a waiting peer
ACCEPT	Accept an incoming connection from a peer
RECEIVE	Block waiting for an incoming message
SEND	Send a message to the peer
DISCONNECT	Terminate a connection



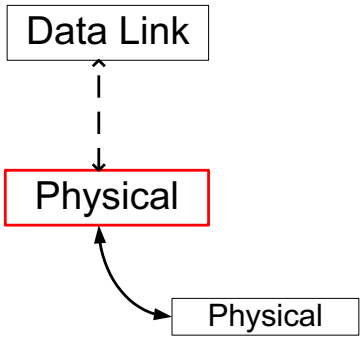


OSI Reference Model



PDUs

OSI – Physical Layer

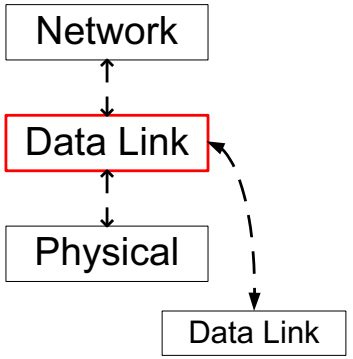


Transmission of bit-stream over medium

Encodes data according to signalling standards

Connectors and cables defined

OSI – Data Link Layer



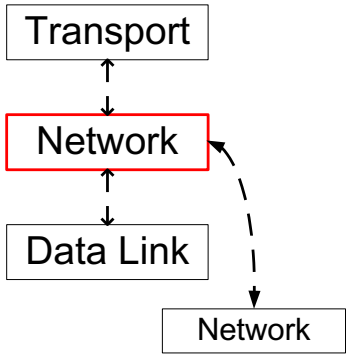
Arranges data into bit stream for sending over physical link

- Data encoded in transmission frames
- Low-level flow and error control for single hop

Possible services to network layer

- Unacknowledged CL
- Acknowledged CL
- Acknowledged CO

OSI – Network Layer

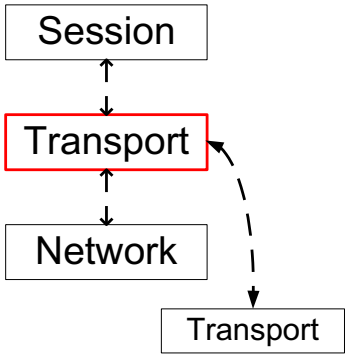


Provides end-to-end transmission of data

- Set-up and termination of connections (CO)
- Global addressing and routing (CL)
- Hides differences in underlying networks

Uses data link layer to provide transmission over single hops

OSI – Transport Layer



Provides transparent transfer service

- End-to-end flow control and error recovery
- Can be more reliable than underlying network

OSI – Session and Presentation Layers

Session Layer

- Enhances transport for sessions with special services
- e.g. dialogue synchronisation, exception handling

Presentation Layer

- Manages syntax and semantics of data exchanged
- e.g. data encryption, authentication, and compression
- e.g. data marshalling, byte ordering, …

☛ We don't look at session and presentation layers much in this course.

OSI – Application Layer

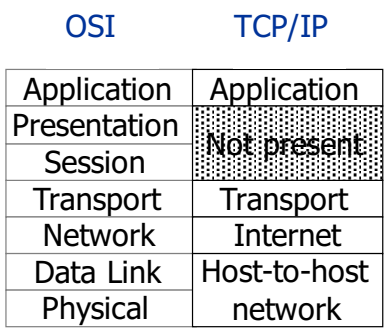
Provides interface to application

- But does not include the application!
- Network functionality specific to given application
- Most users only have contact with app layer

Protocols for common application interactions

- e.g. file transfer, e-mail, web

TCP/IP Model



Developed by DoD for ARPANET

- Still used in Internet
- Designed to be resilient to failures

Presentation and session functions not seen as necessary

Host-to-host network largely undefined

End-to-End Principle (Saltzer, Reed, Clark)

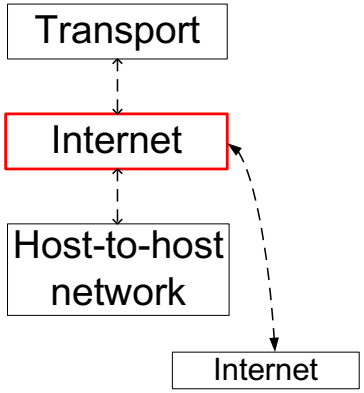
"Communications protocol operations should take place at ends of protocol connection"

- Data link control happens at ends of wires
- Network control happens at ends of subnets
- Transport control happens at ends of connections

Results in efficiency and transparency

- Each layer doesn't do unnecessary work
- Intermediate nodes don't process higher layers
  - Could result in unexpected behaviour

Internet Layer



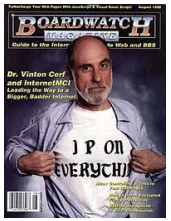
Packet-switched (PS), connectionless (CL), inter-networking layer

Delivery to destination

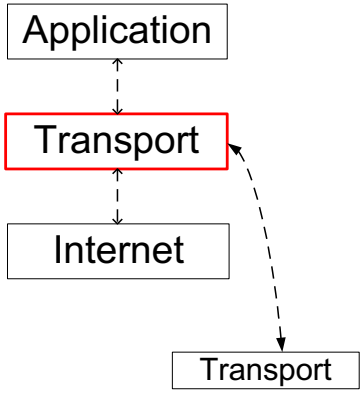
- Routing, congestion control
- Hides different physical networks

IP protocol realises layer

- Defines packet format



Transport Layer

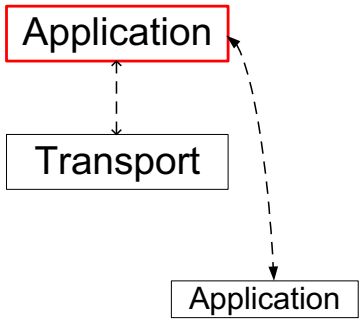


End-to-end connections

- Flow control
- Error recovery

TCP and UDP realise layer

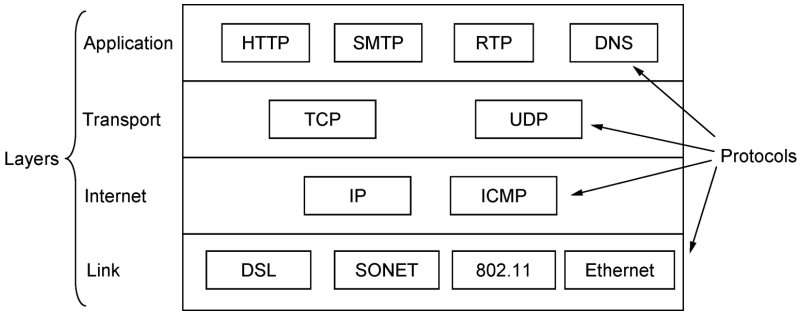
Application Layer



Protocols for application interaction

- HTTP (web)
- SMTP (e-mail)
- DNS (host naming)
- FTP (file transfer)
- NNTP (usenet news)

Example Protocols



Comparing Reference Models

OSI Model	TCP/IP Model
The standard model	Concepts lack generality
Can be complex, not all layers always used	Host-network layer poorly defined
OSI protocols unpopular and poor implementation	TCP/IP protocol most widely used

- Computing (and this course) tends to use OSI model but Internet protocols

Imperial College London

Computer Networks and Distributed Systems

Part 2 – Computer Networks: Physical Layer

Course 527 – Spring Term 2015-2016

Emil C Lupu & Daniele Sgandurra
e.c.lupu@imperial.ac.uk & d.sgandurra@imperial.ac.uk

Part 2 – Contents

Physical Layer
- Properties of communications media
- Signalling, modulation and multiplexing
- Overview of common physical layer technologies

Think about impact on design decisions of layers above

Physical Layer

Provides communications path between nodes

Uses standards
- Agreed ways of connecting devices and signalling
  - Be able to interpret signals
  - Must deal with limitations of physical world

Not going into EE details (or physics)!

Properties of Wired Connections

Signals travel through wires at fixed speed
- Medium can carry signals at many frequencies

Attenuation: signals get weaker over distance

Signals may suffer from interference
- Shielded wires help with attenuation & interference
- Twisting also helps with interference
- Often wires require termination

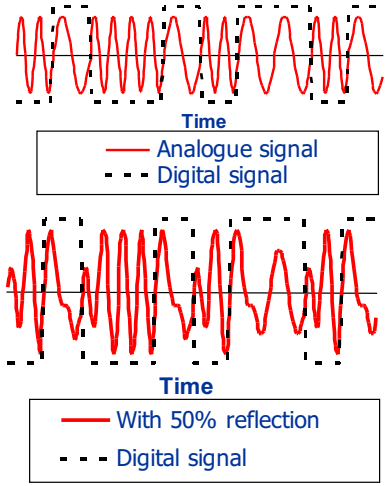
Network goes only where you lay it
- Wires costs money, fibre-optics cost even more

Reflection and Termination

Reflection from ends of wires causes interference

Need **termination** to absorb signal at ends

- e.g. coax-based Ethernet and SCSI use terminators



Properties of Wireless Connections

Signal travels through wireless at fixed speed

- Medium can carry signals at many frequencies
- Different radio frequencies disperse differently

Radio signal suffer from attenuation and interference

- From other transmitters and from reflected signals
- Need to manage power to avoid interference

Radio signal goes wherever it can

- Radio bandwidth subject to regulation
- Environment can block radio waves

Modulation

Modulation: transform information signal into signal more appropriate for transmission on physical channel

Data and signal may each be digital or analogue

- Digital ➔ only values are zero and one
- Analogue ➔ continuous range of values

Digital Data ➔ Digital Signals	NRZ, Manchester
Digital Data ➔ Analogue Signals	ASK, FSK, PSK
Analogue Data ➔ Digital Signals	PCM (pulse code)
Analogue Data ➔ Analogue Signals	AM, FM, PM

Baseband vs. Broadband

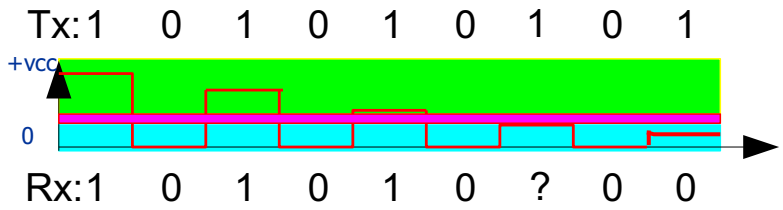
Baseband network (Ethernet, serial, …)

- Medium directly transmits digital/analogue data
- Uses single frequency band (0…f Hz)
- Very simple
  - e.g. Ethernet, serial, …

Broadband network (television, ADSL, …)

- Modulate analogue carrier wave to transmit data
- Can choose good frequency for channel
- Can use multiple bands (f1..f2 Hz, f3..f4 Hz, …)
- Can share channel among multiple users

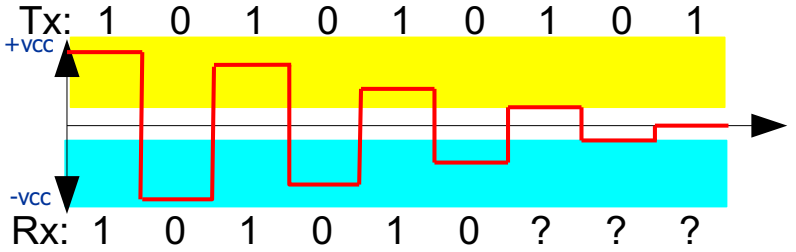
Line Coding: TTL Signals



Binary value represented by state
- "High" voltage defines a 1
- "Low" voltage defines a 0
- Undefined between levels

As signal degrades with distance:
- 1 becomes undefined and then becomes 0

Differential TTL Signals e.g. RS232



Binary value represented by state
- "Positive" voltage defines a 1
- "Negative" voltage defines a 0
- Undefined around 0

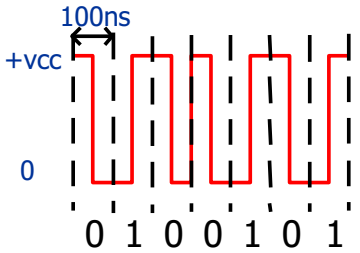
Value becomes undefined as signal degrades
- But never incorrect as polarity not lost

Synchronisation: Clocks

Receiver must identify which bit of data is being sent
- Easy if sending 0101010
- Harder if sending 0001100
  - Could be heard as 001100 if timing wrong

Need **synchronisation** between sender + receiver
1. Slow data rate so slight inaccuracy doesn't matter
2. Separate signal with clock in it
3. Modify signal so that clock is built in

Manchester Encoding



Ethernet encoding (inverse)
- 0: high-to-low
- 1: low-to-high

Thomas encoding
- 0: low-to-high
- 1: high-to-low

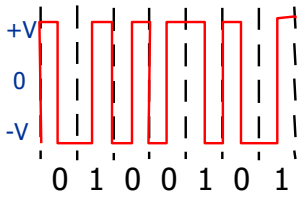
XOR signal with clock 1 clock cycles/bit
- Every bit has at least one transition

Binary value represented by type of transition
- Signal changes simplify clock synchronisation
- Signal changes enable fast detection of signal
- Requires twice bandwidth of simple binary encoding
- Transition at start has no meaning.

Differential Manchester Encoding

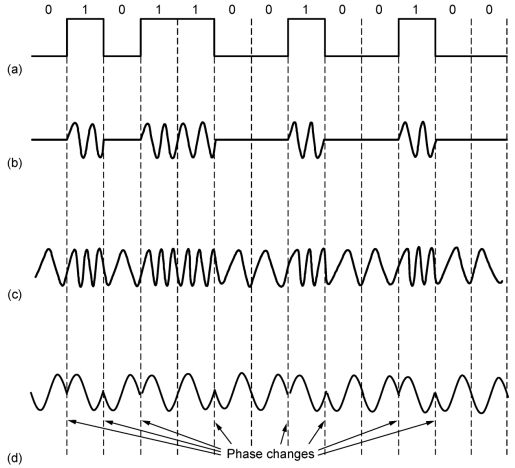
Binary value represented by presence/absence of transitions

- Every bit has at least one transition
- Better noise immunity
- Polarity doesn't matter
- But requires more complex equipment



0: transition at start
1: no transition at start

Broadband Modulation



(a)A binary signal. (b)Amplitude shift keying (BASK). (c)Frequency shift keying (BFSK). (d)Phase shift keying (BPSK).

E.g. to transmit digital data over analogue channel

Use **carrier signal** (periodic wave form) and vary:

- amplitude
- frequency
- phase

Combination of amplitude and phase often used

More Terminology

Baud

- Rate at which signal level (modulation) changes (signal elements per second)

Data rate

- Rate of data transmission (bits per second)

Multiplexing: Sharing Channels

Signal occupies bandwidth in channel

- But it need not occupy whole channel
- e.g. many radio stations operate in parallel

Multiplexing

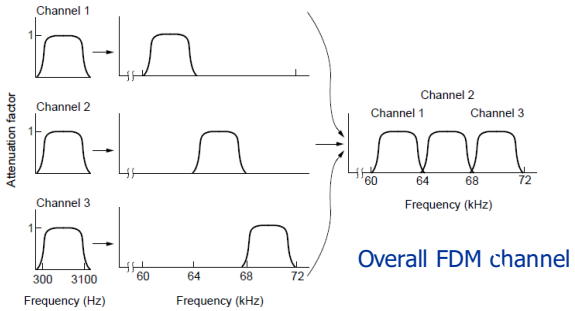
- We examine three techniques for sharing a medium

Frequency Division Multiplexing (FDM)

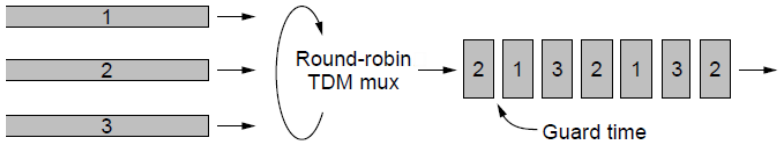
Encode different signals by sending at different frequencies
- e.g. Radio, TV, GSM, …

Need guards bands because filters imprecise

Someone must allocate frequencies to users



Time Division Multiplexing (TDM)



Subdivide channel into fixed time **slots**

Encode many signals by sending at different times
- Examples: phone calls in trunks, TV schedule, …

Issues in TDM

Whole bandwidth channel usable for duration of slot
- But input signals must have bandwidth less than medium bandwidth / number of channels

Introduces delay while waiting for slot
- Gap between slots must not interfere with requirements

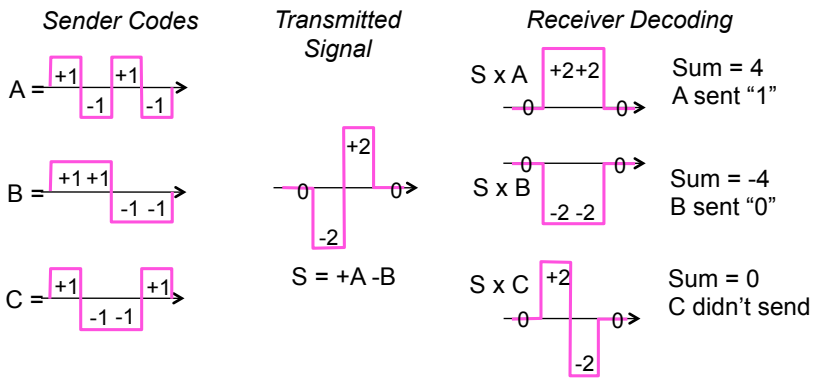
Someone must allocate time slots
- Needs synchronisation to keep track of slots
- Fixed allocation bad for bursty data

Code Division Multiple Access (CDMA)

Imagine many groups having conversations in same room
- TDM ➔ taking turns to talk
- FDM ➔ talking in isolated groups not heard by others
- **CDMA** ➔ everyone talking in different languages

Stations transmit over entire frequency spectrum
- Transmission divided in intervals (chips)
- Stations combine data bits with own code sequence
- Interference between signals occurs
- Separation made using coding theory

Examples: UTMS, satellite transmission, …



Common Connection Standards

☛ No need to know all the details of wiring!

Issues in CDMA

Only practical for communication with central station

- Interference needs to be controlled
- Requires sophisticated signal power management
  - "Everyone can talk as long as no-one talks too loud"

Flexible allocation of channel resources

- Soft degradation as number of stations increases

PC Parallel Port

25 pin connector

- 8 data bits at a time
- 4 control lines to printer, 5 signals from printer

TTL voltage signals

- 0 to 0.8V = OFF = 0 2.0 to 5.0V = ON = 1

100 kBytes/sec max transfer speed

- Note: Bytes as its parallel

5-15m max cable length

- 1 can become 0 with long wires

PC Serial Port (RS-232)

9 or 25 pin connector

- 2 data lines (send and receive)
- 2 control lines, 4 status signals

Differential TTL signals

- +3.0 to 12v = space = 0 -3.0 to –12v = mark = 1

256kb/s max transfer speed

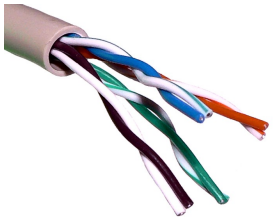
15m max cable length

- Even with attenuation polarity maintained

Analogue Phone Lines

Twisted pair cable

- Send/receive wires are twisted together to reduce interference/radiation
- Different versions (shielded/unshielded, CAT3, CAT5)



Use **modem** to send data using tones

- Telephone system has filters to limit range of tones
  - Only permits 300Hz–3kHz (human voice)
- Approx 2400 distinct tones/sec (2400 baud)
- 56k bits/sec best practical data rate

Digital Phone Lines

ISDN Basic Rate Interface (BRI)

- 2 x 64kb/s bearer/data (B-channels)
- 1 x 16kb/s control/signalling (D-channel)

Two twisted pair cables

ISDN Primary Rate Interface (PRI)

- 23 (US) or 30 (EU) x 64kb/s data channels
- 1 x 64kb/s control
- T1=1.544Mb/s (US), E1=2.048Mb/s (Europe & Asia)

Many others, including OC3 = 155.52Mb/s (optical)

ADSL

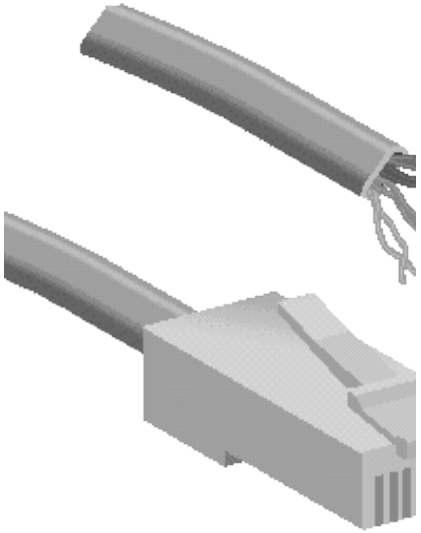
(Asymmetric) Digital Subscriber Line

- Data rate: 128kb/s - 8Mb/s
- "Always-on" behaviour
- Slower sending than receiving

Uses digital phone system

- Remove voice filter to increase bandwidth
- Subdivide channel into frequency bands and use good ones
- Limited range from exchange (typically 5km)

Ethernet (802.3 10/100Base-T)

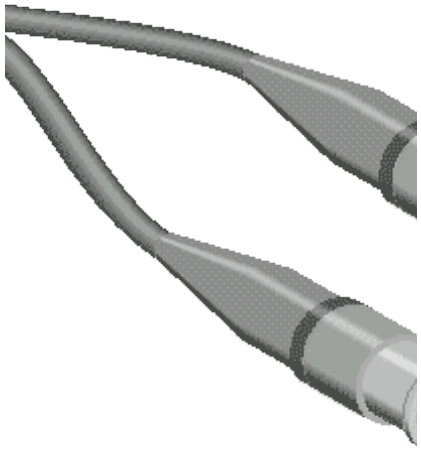


100Base-TX most common cabling in office LANs today (allows 100Mb/s)

10Base-T is phone wire, allows 10Mb/s, found in older networks

100m max segment length
- 1024 connections per segment (with hub)

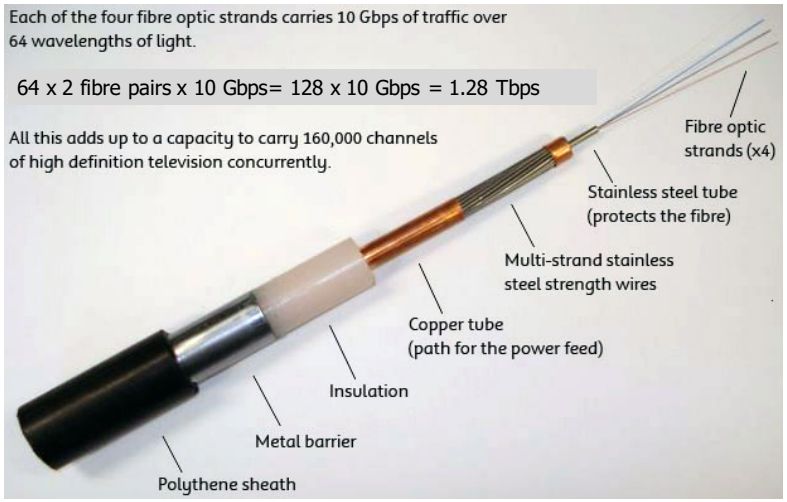
Fibre Optics (10/100Base-F)



Commonly used for:
- Backbones
- High speed networks
- Environments with high electrical noise
- Highly secure networks
  - Taps hard to make

2km max segment length
- Max 1024 connections per segment

Internet Cables



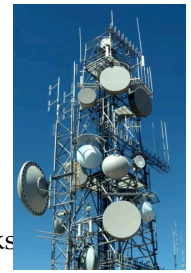
Microwave (Satellite Links)

Satellite acts as relay between two ground stations

Uses two frequencies: **uplink** and **downlink**
- Useful frequency range 1-10GHz
- Needs well-aligned parabolic dish with clear sight
- Satellite signal is typically broadcast over a wide area

Has long latencies (approx. ¼ second)
- Noticeable in speech
- Problems where protocols assume far lower latencies

Also used for point-to-point terrestrial links
- Good for long distance/rapid deployment links



Wireless Ethernet

IEEE 802.11, 802.11b, 802.11a, 802.11g, 802.11n, …

1Mb/s – 54Mb/s (and more)

2.4 GHz and 5 GHz
- Frequency band not restricted

500m range (at 1Mb/s in open)
- Affected by walls, microwave ovens, …

☛ *More later on this*

Mobile Telephones

Operate as cells transmitting to/from base station

GSM (2G)
- Based on TDM & FDM
- Widespread
- 9.6kb/s maximum

GPRS (2.5G)
- Development of GSM
- 115kb/s possible; 28kb/s typical

UMTS (3G)
- Based on CDMA
- 115kb/s – 2Mb/s range
- 384kb/s downstream normal

Bluetooth

Networking of personal devices
- Deliberately short range (typical max. 10m)
- Aims to be power efficient
- Provides serial link abstraction

Data rates between 57.6kb/s - 723.2kb/s
- Centralised TDM: master/slave design
- Frequency-hopping spread spectrum
- Interferes with 802.11b

Imperial College London

Computer Networks and Distributed Systems

Part 3 – Data Link Layer

Course 527 – Spring Term 2015-2016

Emil C Lupu and Daniele Sgandurra
e.c.lupu@imperial.ac.uk, d.sgandurra@imperial.ac.uk

Part 3 – Contents

Data Link Layer

Arranges data into bit stream for sending over physical link
- Defines communication between two physically connected network nodes
- Must cope with different physical layer technologies

Two sub-layers:

Logical Link Control (LLC)
- Low-level flow and error control for single hop

☛ *Not really covered in this course*

Media Access Control (MAC)
- Framing, addressing and channel access

Data Link Layer Services

1. Unacknowledged connectionless service
   - Independent frames with no logical connection
   - No recovery from loss but fast
   - Common in LANs at data link layer e.g. Ethernet

2. Acknowledged connectionless service
   - Each frame in acknowledged
   - Out of order delivery possible
   - Good for unreliable channels such as wireless e.g. 802.11

3. Acknowledged connection-oriented service
   - Connection established before data is sent
   - Each frame numbered and guaranteed to be delivered exactly once and in order
   - Provides reliable bit stream

LLC: Error Detection and Correction

Detection

Serial connections use 8 data + 1 parity bit
- Makes total number of 1s odd (or even)
- Detects all single (and odd numbered) bit errors, misses even bit errors

Cyclic Redundancy Check (CRC)
- Hash-based checksum (often implemented in H/W)

Correction

Forward Error Correction (FEC)
- Add more redundancy ➔ greater capacity to detect/correct bursts of errors
- e.g. use 5 bit codewords to encode 2 data bits

Data Framing

Why are frames necessary? Min size? Max size?

Need to group bits into separate messages
- Large **frames** have less overhead, but:
  - Have greater chance of collision
  - Cost more to retransmit if error detected

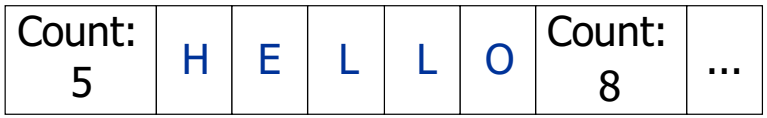
Need to add meta data to control protocol
- Addressing, length, frame type, CRC, …

Need to provide error detection/correction
- Physical layer may introduce errors by adding, removing, or modifying bits

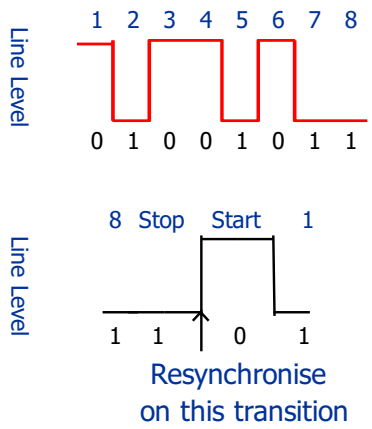
How to break bit streams into frames?

Insert gaps
- But timing hard to guarantee. How big a gap?

Count characters
- Include length field to delimit data
- Often used with another framing method



Serial Line Framing



Counting for framing
- Data transmitted at agreed rate
- But clocks may not be accurate

Use **start/stop** bits
- At least 1 transition per byte
- Bit asynchronous and byte synchronous

Framing: Flags

Start and end flags
- Special signal at start & end of frame: "FLAG"
- Search for flag if receiver loses track

FLAG	Header	Payload data from network layer	Trailer	**FLAG**	**FLAG**	Header

Uses **byte stuffing**
- Identify data with same bit pattern as the flag
- Use escape sequence to identify "next byte is data"

Stuffing

A	**ESC**	FLAG	B

C	**ESC**	**ESC**	D

IEEE 802.3 and Ethernet

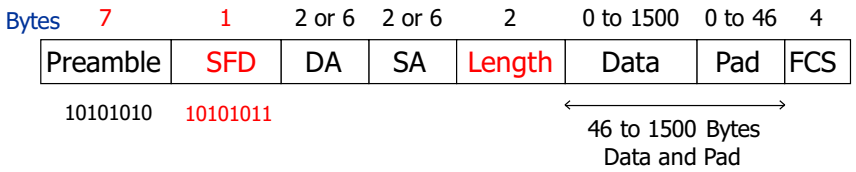
Originally developed by Xerox
- Became open standard
- These days it's almost a marketing term…

Uses Manchester encoding for line transitions

Operates over various physical media
- Data link layer is separate to physical layer
- But physical layer affects parameters of Ethernet

Two standards: IEEE 802.3 and Ethernet

IEEE 802.3: Frame Format



Ethernet standard slightly different
- Doesn't have SFD field
- Replaces Length with Type field

Preamble

- 7-byte alternating 0s and 1s to establish synchronisation
- Framing then by timing, spaces between frames (96 bytes) plus counting from length field

SFD (start frame delimiter)

- 10101011 indicates start of frame
- Allows receiver to miss start of preamble and still synchronise
- Compatibility with 802.4 and 802.5 (Token Ring)

Destination address

- 16 or 48 bits (depending on implementation)
- Host(s) intended to receive
  - Single host (unicast)
  - Group address (multicast)
  - Global address (broadcast)

Source address

- 16 or 48 bit address of sender

Type (Ethernet only)

- Identifies higher level protocol

Length (IEEE802.3 only)

- Bytes in this frame (optional)

Data

- Speaks for itself
- Includes higher layer headers

Pad

- 0-46 bytes to ensure frame long enough to enable collision detection (*a few slides away*)

FCS (Frame Check Sequence)

- CRC, based on all fields except preamble, SD and FCS
- Enables error detection

Ethernet Addresses

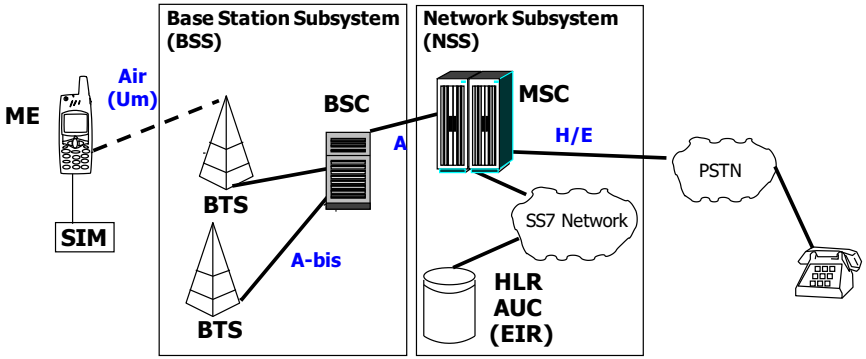
Usually, Ethernet addresses are 48 bits:

Bit 47: 0 = ordinary addr 1 = group addr
Bit 46: 0 = global addr (fixed in hardware),
1 = local addr (assigned by admin)
Bits 23-45: Vendor code (IEEE assigned)
Bits 0-23: Unique code, set by vendor

$2^{46} \Rightarrow 7 * 10^{13}$ possible global addresses

Written as 6 pairs of hex digits
- e.g. 00:11:85:7A:BC:E4

Global System for Mobile Communication (GSM)



SIM: Subscriber Identity Module
ME: Mobile Equipment
BTS: Base Transceiver Station
BSC: Base Station Controller
MSC: Mobile Switching Center
HLR/AUC/EIR: various databases

More info: https://styx.uwaterloo.ca/~jscouria/GSM/gsmreport.html

GSM Station Types

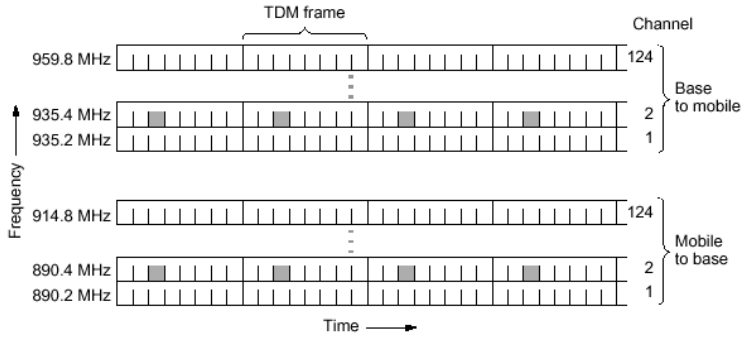
Mobile Equipment (ME) (terminal)
- Fixed (e.g. in cars; max power 20W)
- Portable (max power 8W)
- Handheld (max power 2W)
  - Power down to 0.8W as technology evolves

Base Transceiver Station (BTS)
- Defines cell
- Many to deploy; need to be rugged, reliable

Base Station Controller (BSC)
- Manages radio resources for 1+ base stations
  - Channels, frequency hopping, handovers

Mobile Service Switching Centre (MSC)
- Acts like PSTN switch
- Mobile subscription handling
  - Registration, authentication, location updating, handovers and call routing

Multiple Access Control (FDM/TDM)



Single mobile phone connection

TDM extends number of users each frequency band can accommodate with FDM

GSM: Channels and Frames

25 MHz frequency band in each direction

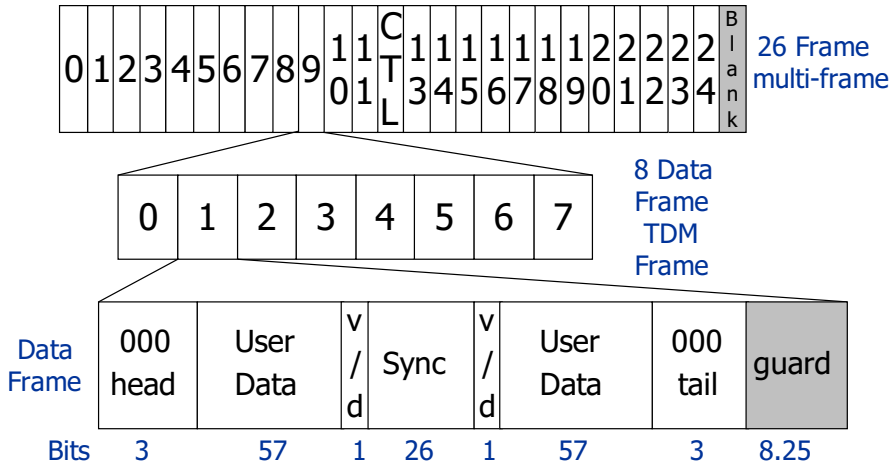
124 frequency channels (FDM) in band

- 200 kHz wide + guard channel to avoid interference

GSM Framing

- 8 data frames in 4.615ms TDM frame
- 26 TDM frames in 120ms multi-frame
  - 24 TDM frames (slots 0-11, 13-24)
  - 1 control data (slot 12)
  - 1 reserved (slot 25)

GSM Framing



Data Frame Structure

Data frame is 54μs

- Two 3-bit **head/tail** fields during power ramping
- Two 57-bit user **data** fields + 1 **data/voice** bit
- 26-bit **sync** field to synchronise frame boundaries and manage multi-path fading
- 8.25bit (30μs) **guard** to separate signals of MEs while signals ramp up

ME can send one data frame every 4.615 ms

- Downlink and uplink separated by 3 frames
- MEs need not transmit & receive at same time

GSM Error Detection/Correction

Some bits of voice data more important than others

Voice data very delay sensitive

3 levels of error correction/detection codes

- Include none on least important bits
- Add forward error correction (FEC) to others
- Replace lost important bits with previous sample

Summary: GSM

Wireless (digitised) voice network
- Also supports data circuits

Communication always via base station controller

FDM/TDM allocation of medium
- Stations assigned bandwidth by controller

Different levels of FEC for voice bits

☛ *There's a lot of GSM which we've not covered here!*

IEEE 802.11 vs. GSM

IEEE 802.11	GSM
Connectionless	Connection oriented
Packet based – Allows contention for medium	Frequency/time slots – Fixed bandwidth for data connection
Distributed and centralised MAC	MAC done by base station
Error/ACK scheme designed for data	Error correction designed for voice

☛ Very different beasts!

Medium Access Control

Physical channel supports multiplexing scheme

But how do we allocate communications channels?
- Contention
- Fairness
- Access latency

Static allocation vs. dynamic allocation

Static Allocation

In Part 2, we looked at TDM, FDM and CDMA
- Static ways for stations to access fixed part of medium

Properties
- Connection-oriented service
- Guaranteed, allocated bandwidth
- Bounded latency to transmit

Dynamic Allocation

But in many computer networks the following applies:

- Most stations do not want to transmit at once
  - Don't waste bandwidth on silent stations
- Need to ensure fair access to medium
  - Would like bounded delay to transmit
- Single transmitter on medium is simpler electronically

Use dynamic allocation

- Allocate time to use medium on demand
- Connection-less service

Use **statistical multiplexing** for TDM

Propagation Delay

Finite time for signal to go from one node to another:

delay = distance / speed (where speed=$2 * 10^8$ m/s)

Finite time to send each signal:

time = 1 / baud rate

Nodes will receive signal at different times

- Depends on distance from sender
- Different nodes will perceive medium to be busy/quiet at different times

Need to keep this in mind when managing who transmits when!

Collision Detection: ALOHA

Original contention network

- Developed at U. Hawaii

Send whenever data ready to go

Two stations whose signals overlap get garbled data

- By listening can detect collision
- Wait random time and try again

Not very efficient

- 18% theoretical maximum channel utilisation

Slotted ALOHA

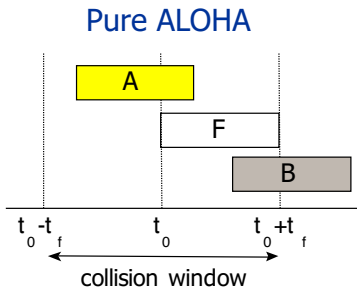
Divide time into slots, each corresponding to one frame

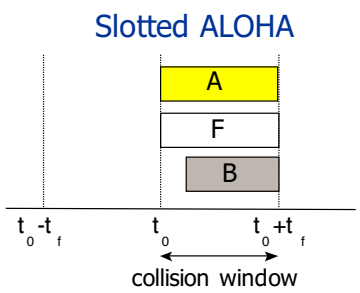
- Start time of frames is synchronised
- Probability of collisions is reduced

Cannot assume synchronised clocks between stations

- Master station sends short signal at start of each time frame

Successful transmission 37% of the time





Assume fixed frames size t_f
- Otherwise even worse performance

Collision window is $2t_f$

Max frame size = slot size t_f

Collision destroys one slot
- Remember propagation delay

Collision window is t_f

CSMA (or being polite)

ALOHA has simple problem:
- No-one listens before they start to send
- Leads to lots of collisions

Carrier Sense Multiple Access (CSMA)
- When ready to send, listen
- If channel busy, wait until idle
- When channel idle, send
- If collision, wait random time and start listening again

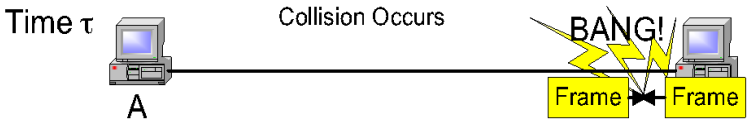
CSMA with Collision Detection

What if two hosts want to transmit?
- Two overlapping signals interfere, needs to be spotted

Collision Detection (CD)
- Listen to channel while sending
- When collision, abort signal with noise burst
- Wait random time and try again

Properties
- Designed for fair access
- Gives unbounded time to access network
- Doesn't waste channel sending broken frames

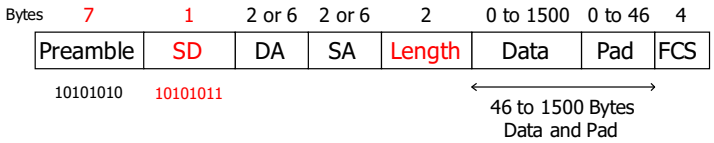
IEEE 802.3: Collision Detection



802.3 allows for 2.5 km max LAN (with repeaters)
- Specifies minimum frame length must be 51.2μs

Ensure sender still sending when collision noise arrives
- Must send for twice the propagation delay
- Assuming 100 ns transitions for sending 1 bit, this means at least 512 bits, hence the pad

Data Frame Format



7+1+2+2+2+46+4 bytes * 8 bits = 512 bits = 51.2 μs

- Time to detect collision over longest network while still transmitting
- When finished sending, cannot guarantee that still listening to hear collision

IEEE 802.3: Retry Timing

Must avoid repeated collisions

- At n^{th} retry, wait between 0 and 2^{n-1} slot times (51.2 msec)
- Do this up to a maximum of 1023 slot times
- Give up on 16^{th} collision

Properties

- Low delay for two hosts' frames colliding
- Reasonable delay for many hosts' frames colliding

IEEE 802.3: Inter-frame Gap

Preamble from another host

9.6 μs interval between successive frames from host

- Allows other hosts to use medium
- Initial frame can be transmitted immediately

CSMA/CD Summary

Fairness

- Equal access to all stations
- No priorities

Probabilistic

- Unbounded access time
- Bad at heavy loads due to exponential back-off

IEEE 802.3 / Ethernet use this

Token Passing ("Collision Free")

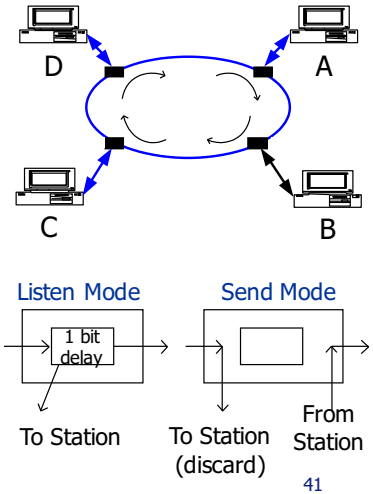
Arrange more orderly sharing of medium
- Uses permission **token**
- Access to medium signalled by passing token around

Avoids collisions through strict control
- But need to handle token control
- Differentiate between tokens and data

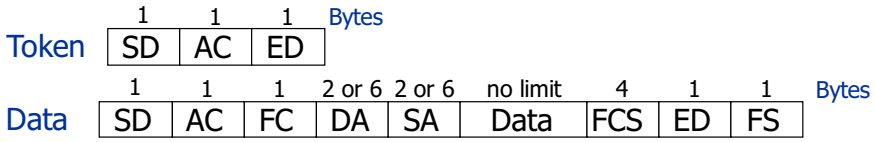
IEEE 802.5 Token Ring

Developed by IBM
- Token Ring and Bus

Token frame inserted by Active Ring Monitor (ARM)
1. Any station takes token and sends data frame
2. Destination copies passing data
3. Sender removes frame on return



Token Ring: Frame Format



Starting Delimiter (SD): **JK0JK000**
- J = high-high, K=low-low transitions
- J & K are invalid in diff. Manchester encoding
  - This is another type of framing

Ending Delimiter (ED): **JK1JK1IE**
- I=1 ⇒ intermediate, I=0 ⇒ last frame
- E=0 from source, set to 1 if error detected (checksum)

Access Control (AC): PPPTMRRR
- Priority P – token priority (different access levels for hosts)
- Token T – flag indicating token or data
- Monitor M – handle failure of source to remove frame
- Reservation R – used to request priority level

Frame Control: FFZZZZZZ
- FF - indicates data / control frame
  - If FF ⇒ data, Z interpreted by destination
  - If FF ⇒ control, all hosts act on control bits, Z

Source/destination addresses
- Similar to IEEE 802.3

Data
- Variable length within token holding time

Frame Check Sequence (FCS)
- 32 bit CRC from SA, DA, Data

Frame Status (FS): ACxxACxx
- A = address recognised, C = frame copied
- Form acknowledgement for each frame
- Repeated for error robustness

Token Passing Summary

Bounded access delay for fair use
- Control passes round all nodes
- But no instant on demand access

Supports giving some stations priority over others
- Not just equal/fair access of contention like CSMA
- Good for real-time control systems

Nice idea but complex in practice ➔ rarely used
- All stations must cooperate
- Must handle token loss

IEEE 802.4/802.5 Token Bus/Ring and FDDI use this

Summary: MAC in Wired LANs

ALOHA
- Contention based service
- Low performance but simple, equal access

CSMA/CD
- Tries to avoid collisions, will detect collisions
- Probabilistic/unbounded access time, equal access

Token Passing
- Avoids collisions
- Bounded access time, access hierarchy but complex

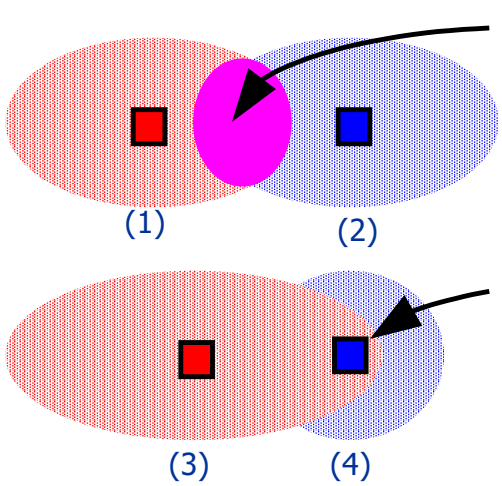
MAC in Wireless LANs

Centralised Medium Access Control
- Good where data is time-sensitive or high priority
- Suffers limits of centralisation

Distributed Medium Access Control
- Good for ad-hoc peers with bursty traffic

No guarantee that all nodes can hear each other
- Makes collision detection harder
- Collision Avoidance (CA) rather than detection (CD)

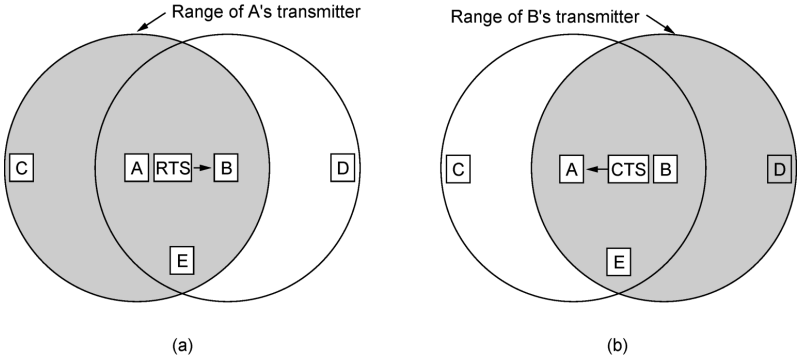
Undetectable Collisions



Signals from (1) and (2) collide
- But (1) cannot hear (2), and vice versa

Hidden receiver problem
- (4) can hear (3)
- But (3) cannot hear (4)

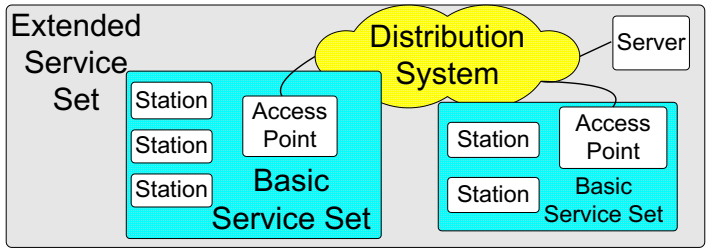
Basic Idea: MACA (Multiple Access with Collision Avoidance)



A sends RTS (Request to Send)
- Anybody who can hear A will hear RTS

B replies with CTS (Clear to Send)
- Anybody who can hear B will hear CTS

IEEE 802.11 - WLAN



Basic Service Set (BSS)
- Smallest building block with stations sharing medium using the same MAC protocol, aka **cell**
- BSSes can overlap

Extended Service Set (ESS)
- Two or more BBSes, connected by distribution system
- Appears as single logical LAN to higher levels

IEEE 802.11 Station Types

Station types based on mobility:

No Transition
- Stationary / only moves within range of one BSS

BSS Transition
- Moves between BSSes in one ESS
- Addressing must recognise new location and deliver via appropriate BSS

ESS Transition
- Moves between BSSes in different ESSes
- Does not guarantee connection to upper layers

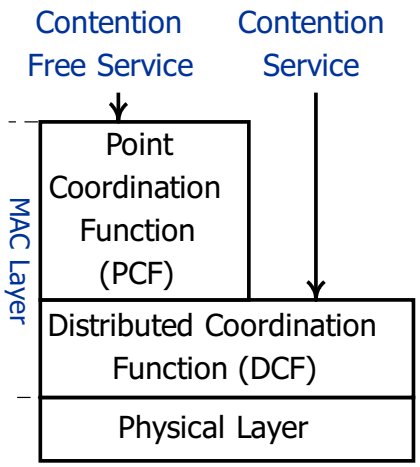
IEEE 802.11 Media Types

Frequency-hopping spread spectrum
- 2.4GHz ISM band with 20 x 1MHz hopping channels
- 1 or 2Mb/s with different FSK encodings
- Low bandwidth but good interference resistance

Direct-sequence spread spectrum (similar to CDMA)
- 2.4GHz ISM band at up to 11Mb/s (802.11b)
- Very good range and variable speed

Orthogonal FDM (similar to ADSL)
- 5Ghz ISM band with 52 narrow bands (802.11a)
- 2.4Ghz ISM band (802.11g)
- Up to 54Mb/s but lower range

Distributed Foundation Wireless MAC



Distributed access control
- With optional centralised control by base station

DCF uses **CSMA/CA**
- Collision Avoidance but no detection (not practical)
- Inter Frame Spaces (**IFS**) give fair access with priorities

Collision Avoidance

RTS (Ready to Send) – request the channel

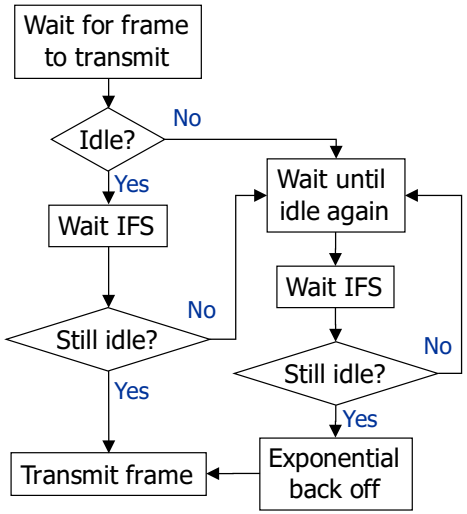
CTS (Clear To Send) – response to RTS frame

ACK (Acknowledgement) – sent on receipt of frame

- MAC-level ACK provides efficient collision recovery

Other stations hear exchange and "sense" channel
- Stations infer how long the channel will be busy
- Repeated failures to transmit ➔ greater back-off time

CSMA with Collision Avoidance



Station with frame to transmit senses medium

If idle and remains idle for IFS period ➔ transmit immediately

If busy ➔ wait until transmission ends + another IFS

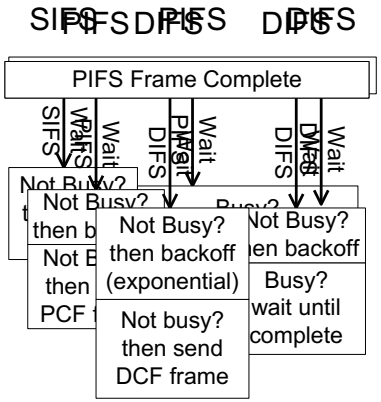
If still idle ➔ back off random amount of time (exponential algorithm) + then transmit, otherwise wait until idle again

Priority and Timing

Short IFS (SIFS)
- Immediate response actions (e.g. ACK, CTS)
- Short IFS gets medium first

PCF IFS (PIFS)
- Medium length
- Polls from central controller

DCF IFS (DIFS)
- Ordinary & management data
- 1st MAC Protocol Data Units (MPDU) of series



Contention example

Multi-Frame Transmissions

Data unit broken into multiple frames
- Individual ACKs good for noisy channels

Once medium acquired send data without interruption
- Following frames sent on receipt of ACK

Use DCF IFS to initiate connection (1st MPDU)

Use Short IFS for later frames of MPDU

IEEE 802.11 Summary

Wireless LAN network

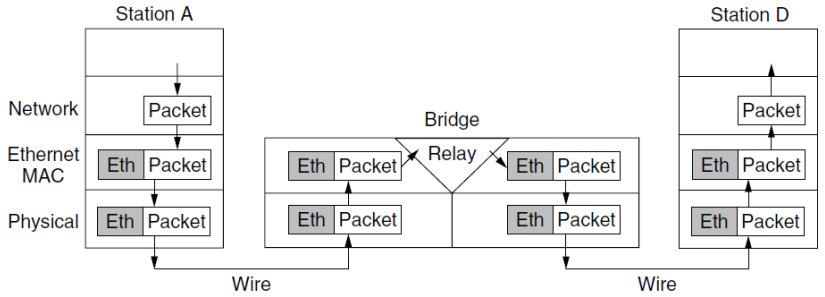
Distributed communication
- Not always via centralised controller

CSMA/CA but no CD

Inter-frame spacing
- Provide priority system using CSMA

Data Link Layer Switching

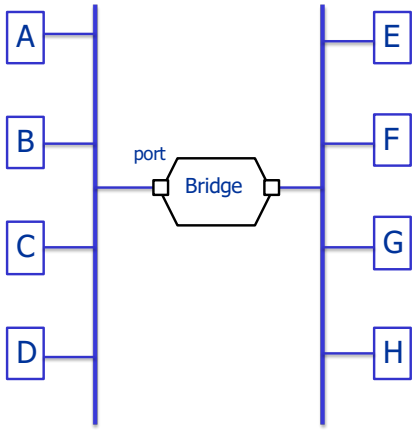
Join LANs together to make a larger LAN
- Layer 2 level **Bridges**
- a bridge for Ethernet it called an **Ethernet Switch**
- the bridge does not interpret higher layer information



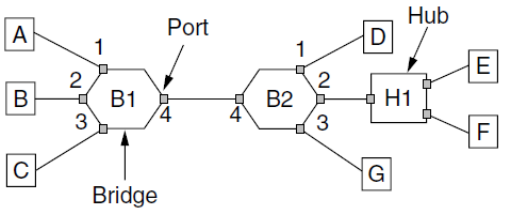
Backward Learning to only relay relevant frames.

- Keep table address/port mappings
- Start by relaying everything
- Learn which addresses are on which ports.

Spanning Tree Algorithm used to eliminate cycles in complex topologies

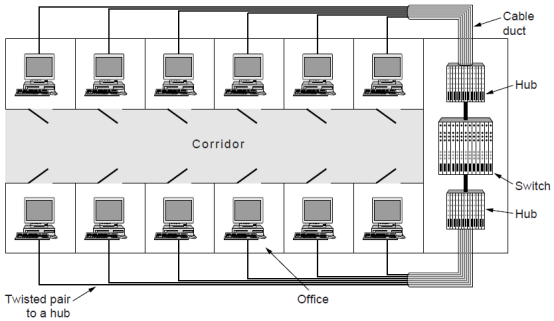


More Complex Topologies



Modern Ethernet switches can connect individual stations

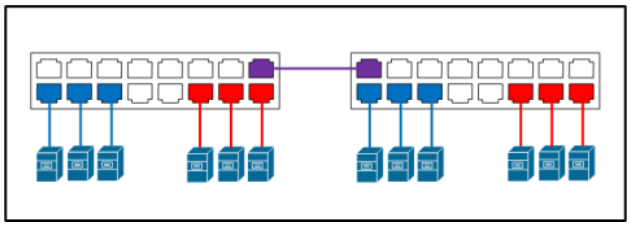
Typical Office Layout



VLANs

Require configuration and changes to the Ethernet headers to use VLAN identifiers.

- Individual stations do not need to be VLAN-aware
- Remember VLANs are layer 2. To connect VLANs need layer 3 routing.



Imperial College London

Computer Networks and Distributed Systems

Part 4 – Network Layer

Course 527 – Spring Term 2015-2016

Emil C Lupu and Daniele Sgandurra
e.c.lupu@imperial.ac.uk, d.sgandurra@imperial.ac.uk

Part 4 – Contents

Interconnecting networks (Layers 1 to 3)
- Repeaters, bridges, routers

Network Layer
- **Routing**
  - Static, distance vector, link state
- **Internet Protocol** (IP)
  - Datagrams (packets)
  - IP addressing
  - Fragmentation
  - Other protocols (ARP, ICMP)

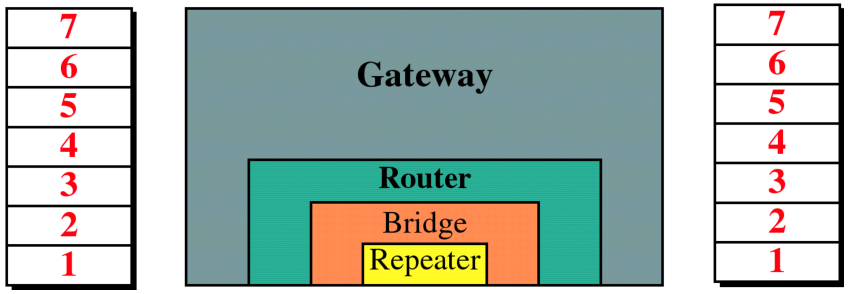
Inter-Networks

Inter-networks formed from smaller networks
- Extending physical limits of networks
- Separating traffic (to spread load or administration)

Different devices interconnect with different low-level protocols
- Cooperation at higher layers to provide uniform service

Connecting Devices and OSI Model

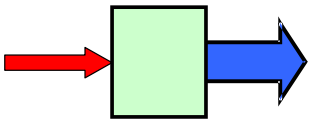
Repeater/Hub
Bridge/Switch



Repeater

Amplifies electrical signal

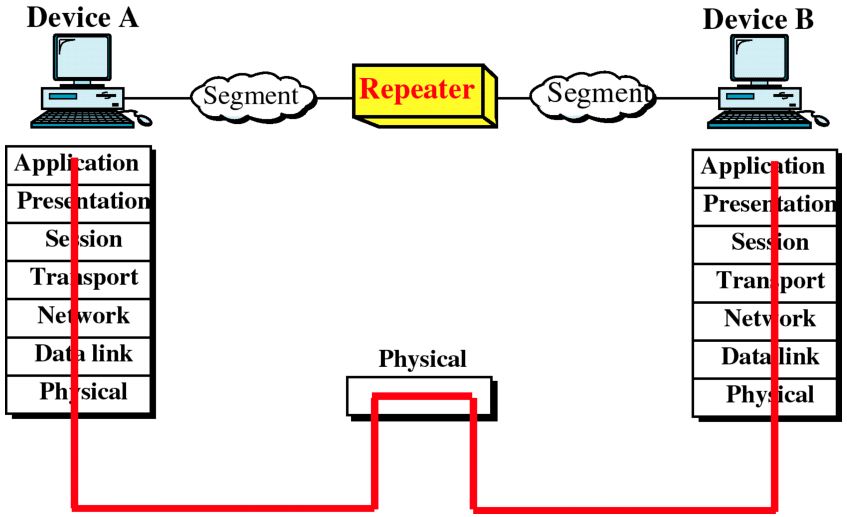
- Makes two wires appear as one
- Improves signal propagation distance



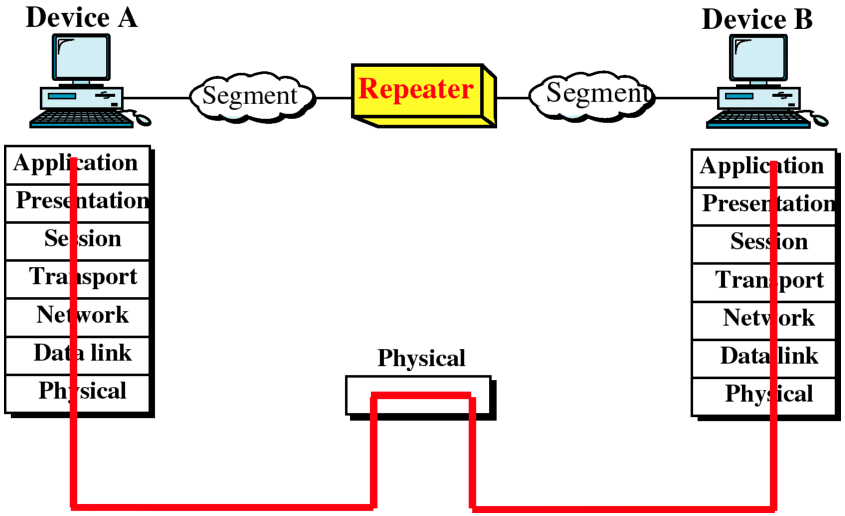
Operates at physical layer

- Transparent to higher layers
- No checking/generating of checksums
- CSMA/CD must cope with longer propagation delays
  - Ethernet (10Mb/s): up to 4 repeaters with 2.5km max length

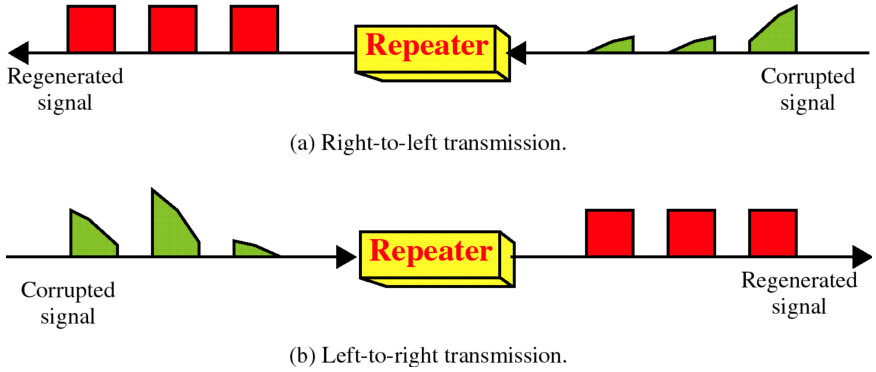
Repeater/Hub and OSI Model



Repeater/Hub and OSI Model



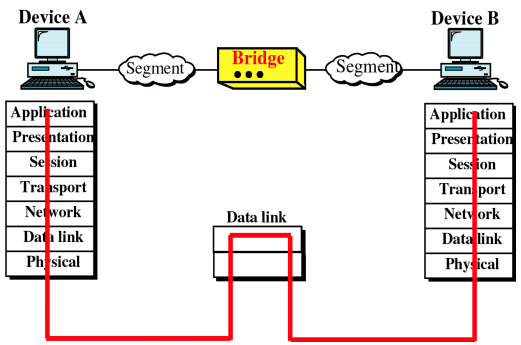
Function of a Repeater



Bridge/Switch

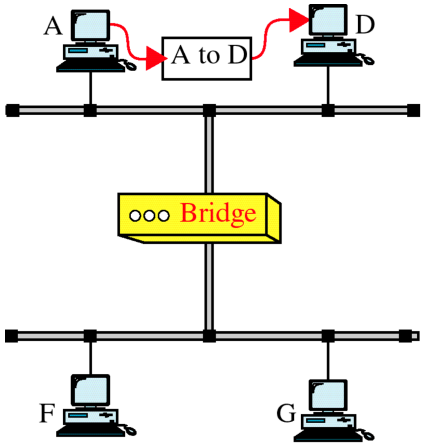
Interconnecting LANs with traffic isolation

Conditional forwarding
- Only forward frames destined for other LAN

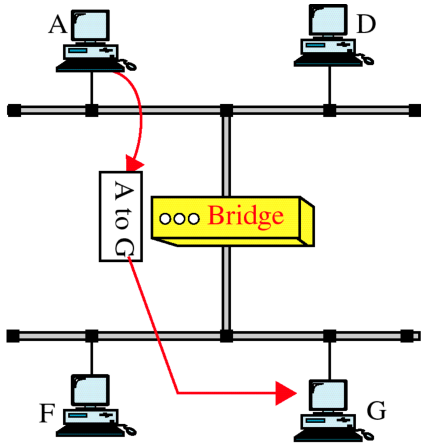


Operates at data link layer
- Reduces load on sub-network
- Store & forward results in higher delay
- Network layers must be same (but not processed)
- Physical layers may be different

Function of a Bridge/Switch



a. A packet from A to D

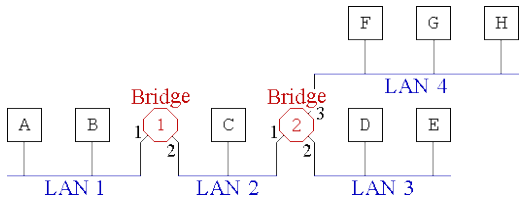


b. A packet from A to G

Transparent (Spanning Tree) Bridge

Bridge records source addrs & links in table

- If destination addr on same link as source ➔ do not forward
- If link for destination addr known ➔ only forward on that link
- Otherwise use flooding
  - Send on all non-source links



Host	A	B	C	D	E	F	G	H									
Link	1	1	2	2	2	2	2										

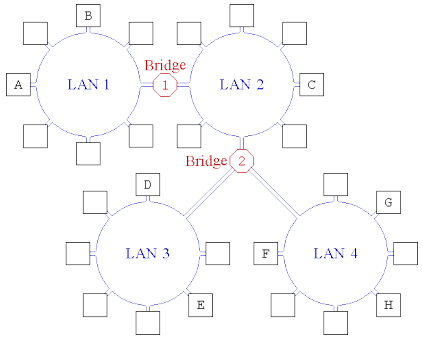
Backwards learning
- Over time all hosts should send frames
- Creates complete host/link tables

Loops in topology
- Make determining location of source impossible
- Causes frames to proliferate

Network layer protocol often handles loops
- Packets may have limited lifetime

Build spanning tree (loop-free subset)

Source Routing Bridge



Bridge issues **discovery frame**
- Copied down every link, recording path list

Destination chooses route based on discovery frames
- Or, sends discovery frames back to sender for decision

Routing path carried in data frames
- Connection-oriented

Keeps bridges simple but end hosts complex
- Hosts must discover routes and putting routes in frames

Route exploration can wipe out benefits
- Bad for networks with high degree of connectivity
- Must cache routes or be very inefficient

Have to rerun discovery if bridge/route fails

Token Ring networks use this

Comparison: Types of Bridges

Transparent	Source Routing
Connectionless	Connection-oriented
– Low overhead to send one frame	– Overhead of discovery on first frame
– Failures handled by bridge	– Failures handled by hosts
Transparent at hosts	Not transparent at hosts
– Backwards learning location of hosts	– Discovery frames locate host
Sub-optimal routing	Optimal routing
Complexity in bridge	Complexity in hosts

Combination: Mixed Media Bridge

Interconnect different networks
- e.g. Ethernet and Token Ring
- Can be source-routing/transparent on different sides
- Holds routing tables which differentiate network type

Handles different **maximum frame lengths** (segmentation/fragmentation)
- 1518 Bytes on Ethernet
- 4KB on 4Mb/s token ring
- 17.6KB on 16Mb/s token ring

Fast Bridges/Packet Switches

Packet switching for LANs
- Since 1984 (DEC were first)
- Switches with one port per LAN
- Reduction of collision domains

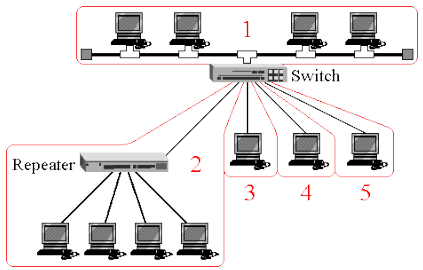
Modern switch technology has improved
- Multi-port bridges forward frames between all ports at wire speed
- Don't use store-and-forward but rather cut-through
  - Forward as soon as dest header field received

Switches are not Hubs

Switched LANs
- Looks bit like shared-bandwidth LAN with hub
- One network cable per computer

But different in terms of:
- frame propagation
- congestion
- cost

Separating Collision Domains



Shared medium requires CSMA/CD to arbitrate
- Contention can be problem on busy network
- Hosts in separate collision domains not competing for media

Switches form ends of **collision domains**
- Reduces to collision domains of 2 (switch + host)

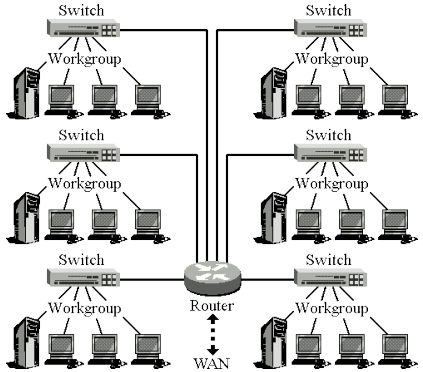
Separating Data Rates

Devices on shared LANs operate at same data rate

Distinct LANs may operate at different rates
- Ports of switch can operate at different rates
- Connection between two hosts limited by slowest hop

Common configuration
- Switch interconnects run much faster than most hosts
- Some hosts have high performance links

Segmentation with Switches



Switches can segment traditional networks

Collapsed backbone

- Backbone in switch rather than shared wire

Hub, Switches vs Routers

Network Switch

- Lives at Datalink layer
  - Knows about MAC addresses and frame formats
- Interconnects network segments

Hub

- Lives at Datalink layer
  - Knows about MAC addresses and frames
- Passively interconnects ports ➔ acts as single network segment

Router

- Lives at Network Layer
  - Knows about IP addresses and IP packets
- Interconnects separate networks
- Carries out more intelligent routing decisions

Network Hubs and Switches

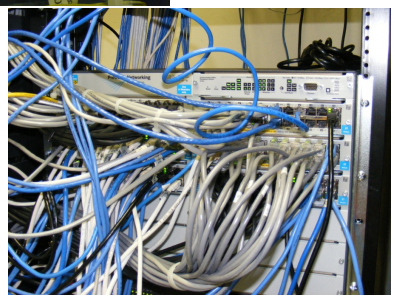
Network Hub

Not on sale anymore?

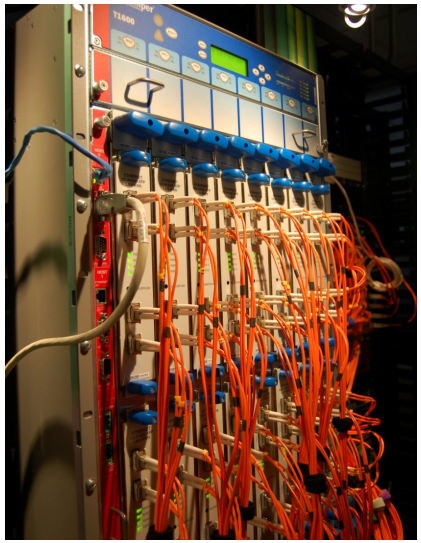
Network Switch

Ethernet Gigabit Switch

- 48 ports
- Cost: ~ £1500



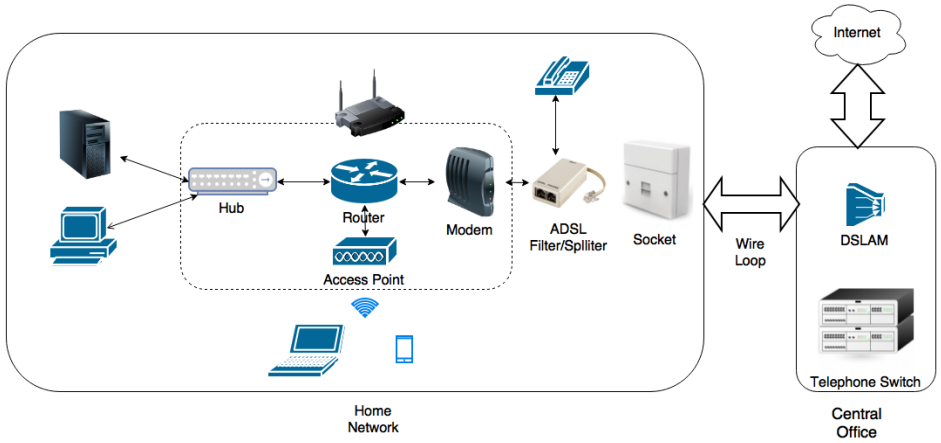
Network Router



Juniper T1600 Core **Router**

- Routes 2 billion packets per second
- 1.6 Tbps capacity
  - 160 ports with 10 Gbps
- Cost: ~ $300,000
- Possible to interconnect up to 16 of them ➔ 30 billion packets per second

Home Network



Routing

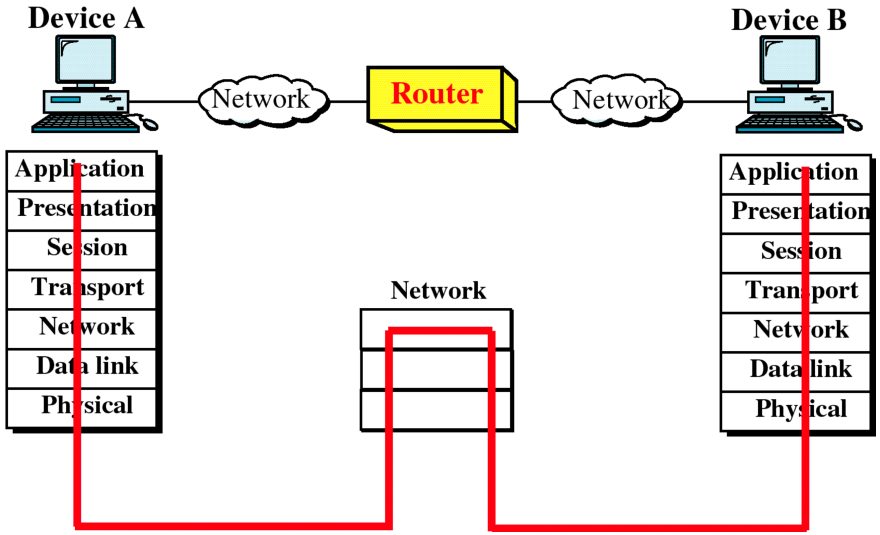
Problem: No single network can serve all users

- Network too long, too much traffic, too complex for lower layers, can't maintain complete network plan
- Think Internet scale!

Solution:

- LANs (subnets) interconnected using **routers**
- **Routing** refers to selecting path from source to destination across multiple subnets
- Network layer must cope with differing underlying LANs

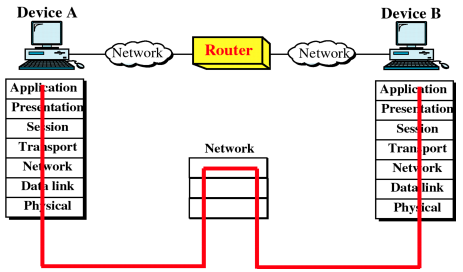
Router and the OSI Model



Router/Gateway

Router

- Determines next hop for packet, depending on dest addr
- Lookup in routing table



Operates at network layer

- Router forwards packets based on dest networks
  - Unlike bridges, which use hosts
- Verifies/modifies packets
  - Updates fields affected by routing
  - Checks/recalculates checksum

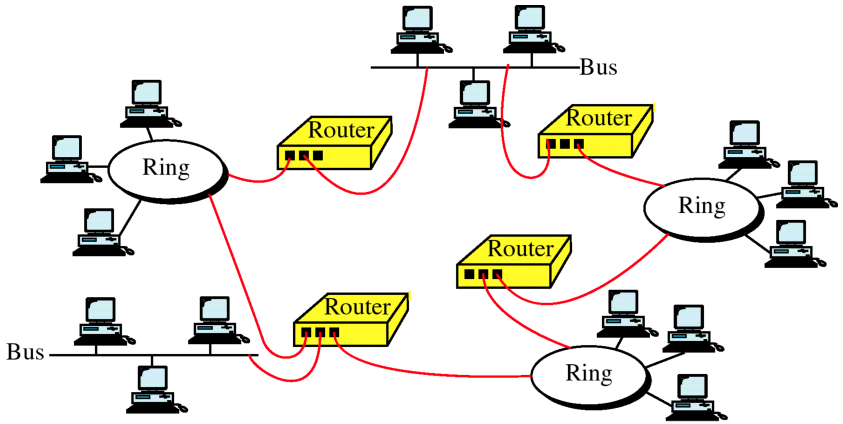
Typically used for connecting sites

- Overcome physical and administrative boundaries
- Greater management and traffic isolation

Not transparent to end nodes

- Frames addressed to router's data link address
- Host needs to know whether/which router to send to

Routers and an Internet



Routing: Objectives

Correctness: Find a route (if it exists)

Efficiency: Routes should provide good performance

- Routes should use minimal resources

Robustness: Return route even when links/nodes fail

Fairness: Hosts should have equal access to network

- Respect priority markings for Quality of Service (QoS)

Adaptability: Routes should reflect network conditions

- But no overreacting to problems

Simplicity: Cheap, predictable and verifiable

Routing: Metrics

Efficiency: Find routes with good properties in terms of

- available bandwidth
- delay
  - Link latencies
  - Hop count
- price
- priority for traffic types

Routing: Properties

No centralised control
- No knowledge of topology or underlying protocols

Interconnection on global (Internet) scale
- May use intermediate networks to get to destination
- Hide underlying interconnection of networks from users
- Networks may be not completely inter-connected

Routing Strategies

Static (non-adaptive) routing
- Compute routes once and load into router
- Worked for early ARPANET

Dynamic (adaptive) routing
- Change routes to reflect changes in topology/load (as seen through congestion)
- Usually used in packet-switched networks
- **Distance Vector Routing** and **Link State Routing**

Non-Adaptive Routing: Static Routes

Routing using fixed directory
- Full address maps to route to host
- Default link for unknown hosts

All packets for host pair always take same route

Often used with list of known hosts/links
- May be set up by pathfinder algorithm (similar to source-routing bridge)

Static routing tables for workstations use this
- Most traffic sent to default gateway/router

Adaptive Routing: Flooding + Random

Flooding
- Send packet to all neighbours except source
  - Unless packet seen before (remove loops)
- Shortest path and fast discovery
- Good for pathfinders and essential/low latency data
- But inefficient and leads to high load on network

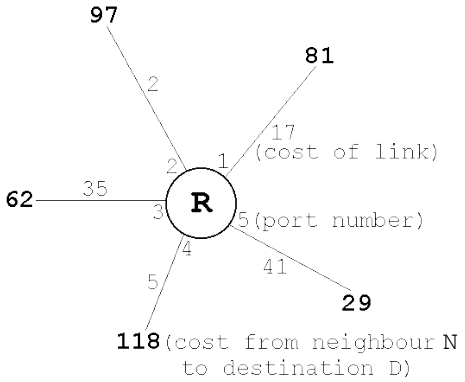
Random
- Forward packet to random link
- Highly robust but slow convergence and inefficient

Adaptive Routing: Distance Vector

Used in ARPANET and Internet until 1979
- By Bellman-Ford, Ford-Fulkerson
- Implemented as Routing Information Protocol (**RIP**)

Router maintains table (vector) of distances
- Usually delay/queue length to each neighbour
- Periodically exchanges this information with neighbours
- Re-computes distance and updates its tables

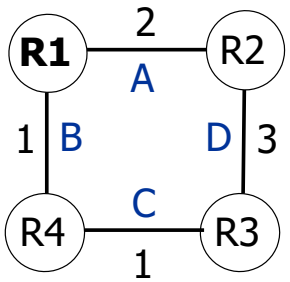
Example: Distance Vector Routing



cost (R → D) =
cost (R → N) + cost (N → D)

Port 1 → 17 + 81 = 98
Port 2 → 2 + 97 = 99
Port 3 → 35 + 62 = 97
Port 4 → 5 + 118 = 123
Port 5 → 41 + 29 = 70

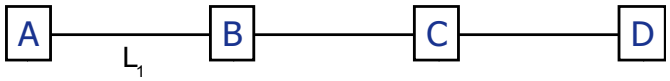
Best choice here is port 5, with distance vector of 70



Step 3

	To R1	R2	R3	R4
Vector R2	2	0		4
Vector R4	1	3		0
Routing Table R1	0 -	2 A	∞ -	1 B

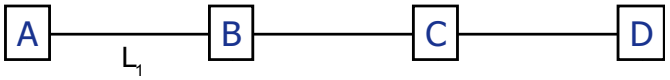
Tutorial Question: Distance Vector Routing



Link 1 is down

	A	B	C	D
A	0	∞	∞	∞
B	∞	0	1	2
C	∞	1	0	1
D	∞	2	1	0

Tutorial Question: Distance Vector Routing



Link 1 comes up

Time	B	C	D
0	∞	∞	∞
T	1	∞	∞
2T	1	2	∞
3T	1	2	3

Link 1 goes down

Time	B	C	D
0	1	2	3
T	3	2	3
2T	3	4	3
3T	5	4	5
4T	5	6	5

Good news travels fast. Bad news travels slowly

- "Counting to infinity" problem. Because e.g., B has no means of knowing it is on the path that C advertises.

Distance Vector Problems

Poor efficiency

- Slow to converge after changes
- Distance vectors increase linearly with network size
  - May not fit inside packet

Route finding suboptimal

- Only considers delay not bandwidth of links
- Prone to oscillations in cost
  - Routing tables do not include paths

Adaptive Routing: Link State Routing

Properties

- Faster convergence and more reliable
- Less bandwidth intensive than DVR
- But more complex and memory/CPU intensive

Each router maintains (partial) map of network

- Consists of more than just neighbours
- May include bandwidth and other metrics

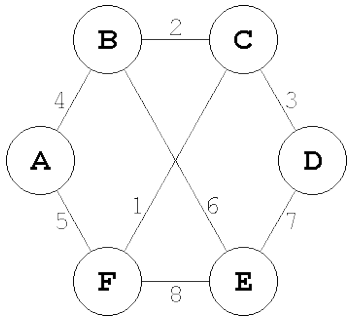
Each router does the following:

1. Discover identities of all neighbours
2. Measure delay (or cost) to neighbours (ECHO packet)
3. Construct and send Link State packet to all routers
4. Compute shortest path to every other router
   - Use Dijkstra's algorithm

When link state changes

- Notification packet flooded throughout network
- All routers re-compute routes

Link State Packets



A	
Seq No	
TTL	
B	4
F	5

B	
Seq No	
TTL	
A	4
C	2
E	6

C	
Seq No	
TTL	
B	2
D	3
F	1

D	
Seq No	
TTL	
C	3
E	7

E	
Seq No	
TTL	
B	6
D	7
F	8

F	
Seq No	
TTL	
A	5
C	1
E	8

Link state packet

- ID of source, sequence number (to handle order & loss)
- Time-to-live (decremented each second until discarded)
- List of neighbours with costs

Link State Distribution

Based on flooding algorithm

- Don't send on incoming link

SeqNo to ensure only newer state packets forwarded

- Drop old & duplicate packets
- Some delay in forwarding to wait for newer packets

Different routers have different views of topology

- Inconsistencies, loops, unreachable nodes

Hierarchical Routing

Complete Internet map in every router infeasible

Instead exploit hierarchy and use regions

- Router knows local topology in detail
- Router knows route to other regions
  - But not their internal arrangements

Regions may map to:

- Geographical area (e.g. London academic network routes between universities)
- Organisation's network (e.g. Imperial has routers in core network, routing between departments and to external links)

Internet Routing

Autonomous systems (AS) are regions on the Internet

Within ASs: Open Shortest Path First (**OSPF**)

- Variant of Link State Routing
- Supports load balancing over multiple lines
- Routing includes type of service (but not used)

Between ASs: Border Gateway Protocol (**BGP**)

- Variant of Distance Vector Protocol
- Records exact path used
- Supports custom routing policies

Summary: Network Interconnection

Repeater (Physical layer)
- Extends range of signals

Bridge (Data link layer)
- Segments collision domains, transparent or source routing

Switch (Data link layer)
- Separates networks, wire speed bridge with multiple ports

Router (Network layer)
- Interconnects LANs, LAN not host addressing, visible to end nodes, needs routing protocol

Internet Protocol (IP)

Basic protocol for the Internet
- Defined in RFC 791

Datagram oriented
- Treats packets independently
- Packets contain complete addressing information
- Unreliable delivery (no notification)
- Variable sized data payload
- No checksum on data payload, just on header

IP Services

Addressing

Packet timeouts
- Avoid congestion and routing problems

Fragmentation
- May split packets if underlying network requires it

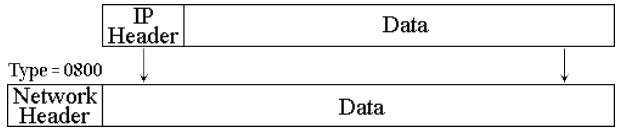
Type of Service through priorities
- Requires routers on path to read and treat differently

Other options
- Source routing requirements, route recording, security labels

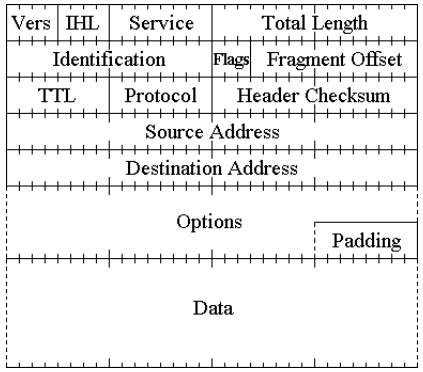
IP Datagrams

IP datagrams are "virtual" or "universal" packets
- IP dest addr is always final destination address
- Physical dest addr in frame is changed at each hop

Along the path each router:
- Removes packet from LAN frame
- Determines next router/local link
- Re-encapsulates in appropriate LAN frame for next hop



IP Datagram Format



Version: IP version (usually 4)

Internet Header Length
- In 4 byte multiples (5<=IHL<=15)
- Options increase this
- Gives data offset

Type of Service
- Trade-off between delay, reliability and throughput

Total Length
- max 64KB with IPv4

IP Header (Cont.)

Time to Live (TTL): Handles routing loops
- Decremented each routing hop
- Datagram dropped when = 0

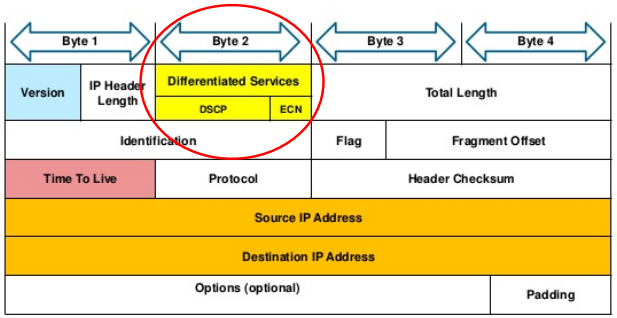
Protocol
- 0 = reserved, 1 = ICMP, 6 = TCP, 17 = UDP
- Similar to Ethernet protocol type field

Header checksum: 1's complement sum of header, not data
- Sum of header and checksum should = 0

Source and destination addresses

Options
- Security, loose/strict source routing, record route, stream ID, timestamp, …
- Padded to multiples of 32bits

IPv4 Packet Header (RFC 2474 & 3168)



Differentiated Services Code Point (DSCP):
- Technologies that require real-time data streaming.

Explicit Congestion Notification (ECN)
- End-to-end notification of network congestion without dropping packets.

IP Addressing

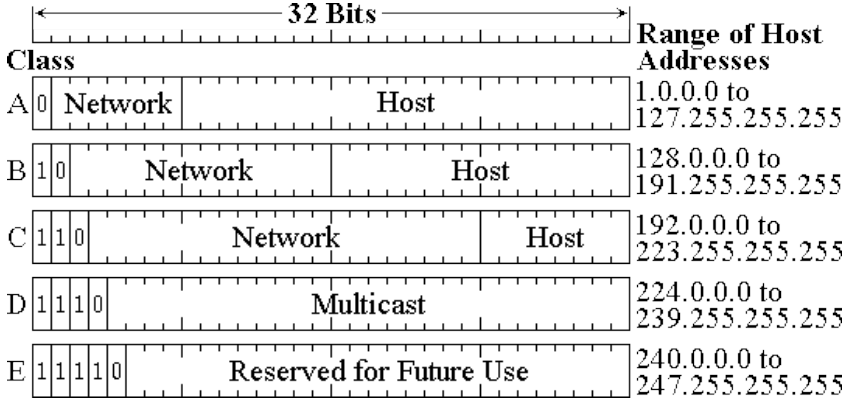
Ethernet addr 48 bits, written as hex pairs

IP addr 32 bits, written as dotted decimal
- e.g. `146.169.7.41`
- No direct mapping of IP addrs to Ethernet addrs

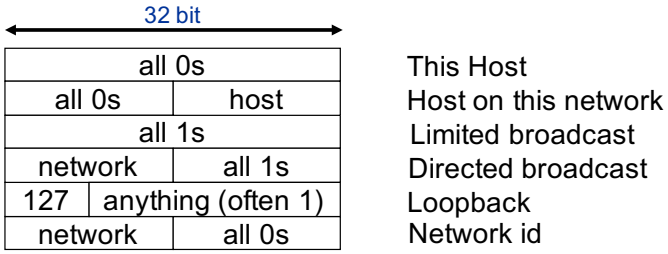
IP addr identifies **network** and **host** on that network
- Not machine but connection to network
- Device on n networks has n IP addrs – one for each

Address space administered by ICANN
- Assigned addrs don't have to be connected

IP Address Classes



Special IP Addresses



Addrs with all bits 0 or 1 are not assigned to hosts
- Useful at start-up if host/network not known

Broadcast is never valid source address

Loopback is for local inter-process communication (IPC)
- Should never exist on the network wire

Special IPv4 Addresses

Address Block	Present Use	Reference
0.0.0.0/8	"This" Network	RFC 1122, Section 3.2.1.3
10.0.0.0/8	Private-Use Networks	RFC 1918
127.0.0.0/8	Loopback	RFC 1122, Section 3.2.1.3
169.254.0.0/16	Link Local	RFC 3927
172.16.0.0/12	Private-Use Networks	RFC 1918
192.0.0.0/24	IETF Protocol Assignments	RFC 5736
192.0.2.0/24	TEST-NET-1	RFC 5737
192.88.99.0/24	6to4 Relay Anycast	RFC 3068
192.168.0.0/16	Private-Use Networks	RFC 1918
198.18.0.0/15	Network Interconnect Device Benchmark Testing	RFC 2544
198.51.100.0/24	TEST-NET-2	RFC 5737
203.0.113.0/24	TEST-NET-3	RFC 5737
224.0.0.0/4	Multicast	RFC 3171
240.0.0.0/4	Reserved for Future Use	RFC 1112, Section 4
255.255.255.255/32	Limited Broadcast	RFC 919, Section 7 RFC 922, Section 7

Private Internet Address Ranges

Address ranges for internal use

10.0.0.0 - 10.255.255.255 (10/8 bit prefix)

172.16.0.0 - 172.31.255.255 (172.16/12 bit prefix)

192.168.0.0 - 192.168.255.255 (192.168/16 bit prefix)

Addresses never routed on public Internet
- Not all devices need to be globally visible
- Used for testing and NAT (see later slides)

Subnets

As organisations grow, need finer control over network sizes

- Single class A/B/C network not good enough

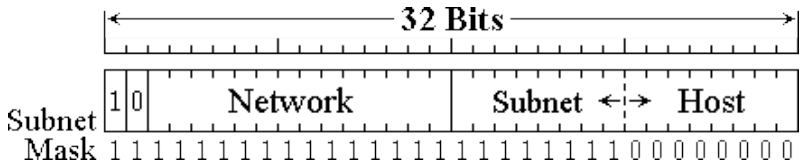
Subnet is sub-network within assigned IP network

- To global Internet there is no distinction
- Internally subnet addrs may be used for routing, admin

Trade division into subnets for num of hosts in subnet

- Subnets can be any size within host field

Subnets



Use high-order bits from host field to create subnets within network class:

subnet mask & address = network portion

Number of hosts and subnets

2^{subnet_bits} = number of subnets per network

- Although usage of all 0s and 1s is not RFC-compliant

$2^{(32 - network\_bits - subnet\_bits)} - 2$ = num of hosts per subnet

- All 0s and all 1s are not valid addresses

Subnet Example

In DoC, we have a class B network

- 8 bits for subnets and 8 bits for hosts
- Subnet mask 255.255.255.0 with class B net: 256 subnets, each with 254 hosts

Example:

- 146.169.7.41 is global IP address of host 41 (columbia) on subnet 7 (DSE group) on IP network 146.169.0.0
- Full DNS name: columbia.doc.ic.ac.uk
- Broadcast to subnet on 146.169.7.255
- 7-net subnet mask of 255.255.255.0, DoC network mask of 255.255.0.0

Issues with IP Addressing

Support for mobility (laptops, phone, …)

- Connect to different points in different networks
- Routing depends on address used

Expansion of networks

- Renumbering/adding new number ranges hard
- Hosts with multiple IP addresses

Total size of address space limited

Address Space Problem

Shortage of unallocated addresses
- Practical address space in IPv4 is 100 million hosts
- IP is more popular than its designers expected

Some addr classes unnecessarily large
- Some organisations have more than they need
- Class B bigger than needs of most people
  - 64516 host addrs with 256 subnets of 254 hosts
  - Never mind class A!

Address Space Solutions

Stricter access to allocation
- Class A "virtually impossible" to obtain now
- Blocks of class C now allocated in preference to class B

Make address allocation more flexible
- Classless Inter-Domain Routing (**CIDR**)

Reuse addresses in different parts of network
- Network Address Translation (**NAT**)

Add more address bits: **IPv6**

Classless Inter-Domain Routing (CIDR)

Partition world into four zones

Allocate networks with variable subnet masks
- Size according to need, not just fixed classes A/B/C
- "Subnetting for global Internet"

Advantages
- More efficient allocation then previous classfull approach
- Works alongside previous allocations

Disadvantages
- Makes routing harder
- Not fundamentally larger address space

Guidelines for Management of IP Address Space (RFC 1366)

Europe
194.0.0.0 - 195.255.255.255

North America
198.0.0.0 – 199.255.255.255

Central/South America
200.0.0.0 – 201.255.255.255

Asia & Pacific
202.0.0.0 – 203.255.255.255

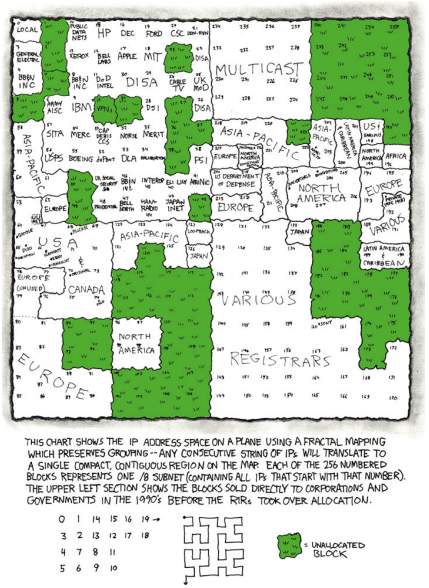
Future use
204.0.0.0 – 223.255.255.255

IPv4 CIDR Chart — **RIPE NCC**

IP Addresses	Bits	Prefix	Subnet Mask
1	0	/32	255.255.255.255
2	1	/31	255.255.255.254
4	2	/30	255.255.255.252
8	3	/29	255.255.255.248
16	4	/28	255.255.255.240
32	5	/27	255.255.255.224
64	6	/26	255.255.255.192
128	7	/25	255.255.255.128
256	8	/24	255.255.255.0
512	9	/23	255.255.254.0
1 K	10	/22	255.255.252.0
2 K	11	/21	255.255.248.0
4 K	12	/20	255.255.240.0
8 K	13	/19	255.255.224.0
16 K	14	/18	255.255.192.0
32 K	15	/17	255.255.128.0
64 K	16	/16	255.255.0.0
128 K	17	/15	255.254.0.0
256 K	18	/14	255.252.0.0
512 K	19	/13	255.248.0.0
1 M	20	/12	255.240.0.0
2 M	21	/11	255.224.0.0
4 M	22	/10	255.192.0.0
8 M	23	/9	255.128.0.0
16 M	24	/8	255.0.0.0
32 M	25	/7	254.0.0.0
64 M	26	/6	252.0.0.0
128 M	27	/5	248.0.0.0
256 M	28	/4	240.0.0.0
512 M	29	/3	224.0.0.0
1024 M	30	/2	192.0.0.0
2048 M	31	/1	128.0.0.0
4096 M	32	/0	0.0.0.0

K = 1,024 • M = 1,048,576

The Internet Map



courtesy of xkcd (2006)

Layout with a space-filling curve

"Old Internet" in top left corner

Go to caida.org for lots of pretty pictures

Network Address Translation (NAT)

Often only fraction of hosts require external access

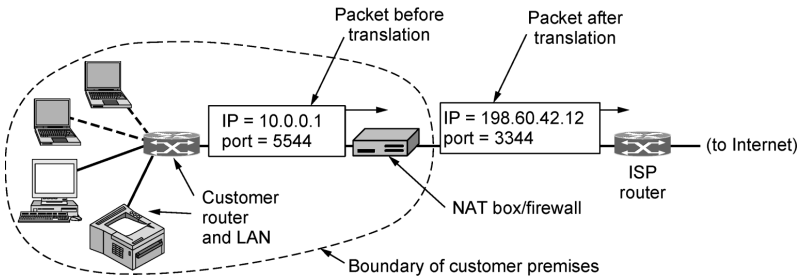
Hide large network in small Internet address range

- External addr: real, allocated IP address
- Internal addrs: from private addresses range
- Gateway box translates internal addrs to dynamically allocated external addrs for traffic leaving LAN

Full address becomes IP addr + port *(next part)*

- External addr may be shared by multiple hosts over time
- Can lead to problems if changes aren't anticipated…

Placement of a NAT Box



Placement and operation of a NAT box.

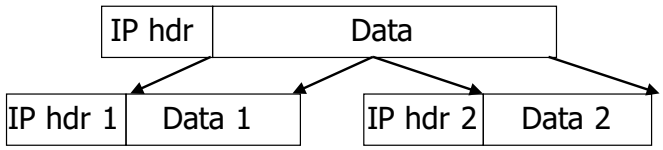
Fragmentation

Networks have maximum transfer unit (**MTU**)

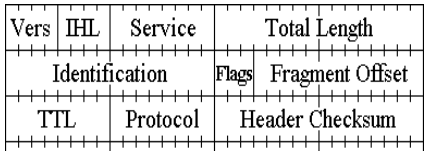
- LAN frames have limit on data carried
- e.g. Ethernet frame <1500 bytes
- Sender only knows datagram size for local network

Fragmentation of IP packets

- Large IP datagrams broken up when moving through network with smaller MTU
- Smaller datagrams need their own complete IP header



IP Datagram Fragmentation I



Fields to aid reassembling fragmented datagram

- Flags:
  - Bit 0: reserved, always 0
  - Bit 1: DF, 0 = may fragment, 1 = don't fragment
  - Bit 2: MF, 0 = last fragment, 1 = more fragments
- Fragment offset: position of fragment
  - In 8byte multiples, 1st is 0

IP Datagram Fragmentation II

All stations must accept fragments of <= 576 bytes

Final destination can reassemble original datagram

- Missing fragments are waited for
- Whole datagram discarded if any are lost
  - Best-effort connectionless delivery
  - Transport layer deals with missing datagrams

Fragmentation adds much complexity to routers

- e.g. multiple levels of fragmentation possible
- Often easier to return ICMP error message *(next slide)*

Internet Control Protocols

Address Resolution Protocol (ARP) - Mapping IP Addresses to Devices

DHCP

Internet Control Message Protocol (ICMP)

Mapping IP Addresses to Devices

Need to translate between addresses

- Data link layer: frames between devices use data link addrs, e.g. Ethernet MAC addrs
- Network layer: hosts send packets using IP addrs

Static mapping

- May be sufficient for small isolated network
- But Ethernet addr space is larger than IP addr space

But IP addresses are virtual

- No relation to hardware, maintained in software

IP supports interconnections of different networks

- Not all devices have Ethernet addresses

Dynamic Address Resolution

Need to bind protocol address dynamically

- Only possible for two devices on same network

Table lookup

- IP addr/data link addr in sequential/hash table

Closed-form computation

- Make physical addr simple function of IP addr

Message exchange

- Dedicated protocol for dynamic lookup, e.g. ARP
- Usual method on TCP/IP networks with static addresses, e.g. Ethernet

Address Resolution Protocol (ARP)

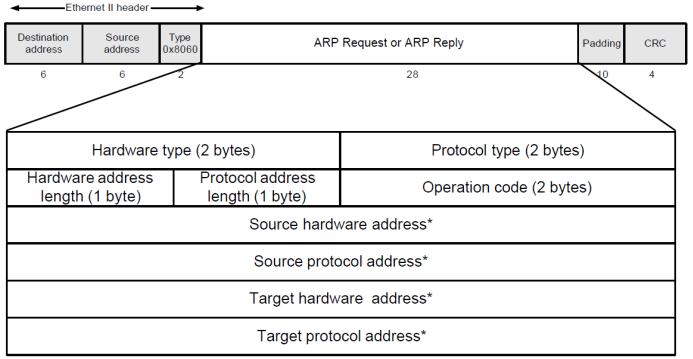
Hosts maintain caches of IP/data link address mappings for LAN

If host A has no entry for host B:

- A broadcasts **ARP request**
  - Requesting data link addr for B's IP address
- B recognises its IP address
  - Returns **ARP response** with its data link address
- B also caches A's data link/IP address mapping
  - Likely to need it in future exchanges

ARP is network layer protocol, not visible to the user

ARP Message Format



* Note: The length of the address fields is determined by the corresponding address length fields

HW Addr Type: 1 = Ethernet. Proto Addr Type: 0800h = IP

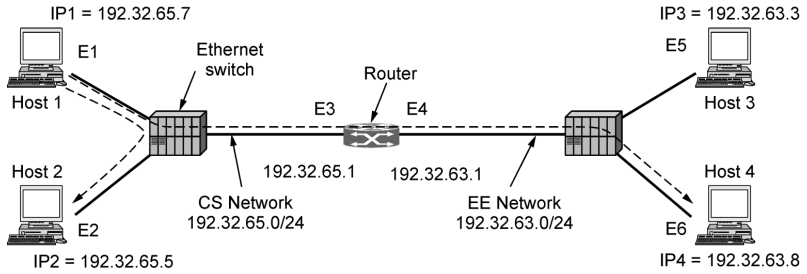
HW Addr Length: 6 bytes. Proto Addr Length: 4 bytes

Operation: 1 = request, 2 = response

Target HW Addr: undefined on request

Target machine swaps target and sender in response

Example of ARP



Frame	Source IP	Source Eth.	Destination IP	Destination Eth.
Host 1 to 2, on CS net	IP1	E1	IP2	E2
Host 1 to 4, on CS net	IP1	E1	IP4	E3
Host 1 to 4, on EE net	IP1	E4	IP4	E6

Two switched Ethernet LANs joined by a router.

Reverse Address Resolution

Determine one's own IP address from Ethernet addr

- e.g. after booting machine

Use **RARP**, giving itself as both target and sender

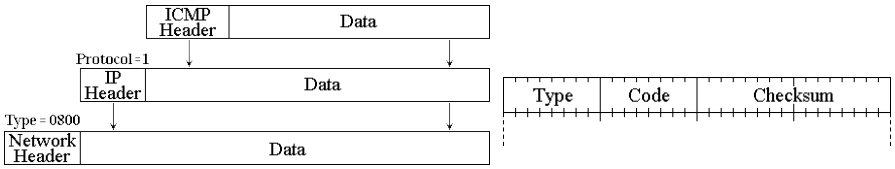
- Need one RARP server per network
- Same format as ARP
- Limited broadcast of requests

Dynamic Host Configuration Protocol (DHCP)

Initially a host will only know its Ethernet address. DHCP used to assign dynamic IP addresses.

- Broadcast request for IP address (including source Eth addr) `DHCP Discover`
- DHCP server allocates address and issues `DHCP offer`
- Client (choses offer if necessary and) replies with `DHCP request`
- Address allocation confirmed with `DHCP ACK`
- Allocation for fixed period of time using leases. Host can request renewal or can release an address
- Also used to give client other configuration parameters, network mask, routers, host name, etc.
- Widely used for cable modems, WLANs, …

Internet Control Message Protocol (ICMP)



Allows routers to send control/error msgs to other routers/hosts

- Behaves as if higher level protocol, but integral to IP

ICMP provides for feedback about comms problems

- IP unreliable ➔ no guarantees of delivery, loss notification, control msg return

ICMP Message Format

Type (8bit) + code (8bit) gives kind of message

Type 3 codes:

0 = Net unreachable
1 = Host unreachable
2 = Protocol unreachable
3 = Port unreachable
4 = Fragmentation needed and DF set
5 = Source route failed

Checksum of type & code, 1s compliment

Some Types:

0 = Echo reply
3 = Destination Unreachable
5 = Redirect
8 = Echo request (ping)
11 = Time exceeded
12 = Parameter problem
13 = Timestamp
14 = Timestamp reply
15 = Information request
16 = Information reply
17 = Address mask request
18 = Address mask reply

Popular client uses of ICMP

Ping
- Used to verify that path works and end host present.
- Collects round trip time and failure
- Send echos, display echo reply.

Traceroute
- How to find out info about intermediate hosts?
- Send packets and increment TTL at each packet.
- When TTL=0 router discards packet but sends ICMP error message.

TRY Them!!

IPv6

IETF addresses many problems of IPv4 with **IPv6**

128 bit addresses (vs. 32 bit in IPv4)
- $3.4*10^{38}$ unique host addresses (vs. $4.2*10^{9}$ in IPv4)
  - $\sim 10^{80}$: number of atoms in the visible universe

Simplified 7 field header (vs. 13 fields in IPv4)
- Faster processing in routers possible
- More options through extension headers
- Support for authentication, privacy, service types, mobility, …

Compatible with IPv4 for transition
- Some gateways and tricks to hide IPv6's greater capabilities

Issues with IPv6

Difficult to implement properly

Transition from IPv4 to IPv6 hard and slow
- Not widely deployed over backbone
  - Router/switch manufacturers not pushing it
  - ISPs and network providers not demanding it
- Many of the benefits lost in gateways to IPv4

Currently useful within organisation
- But not many people to talk IPv6 with
- Mobile phones may push adoption…

Imperial College London

Computer Networks and Distributed Systems

Part 5 – Transport Layer

Course 527 – Spring Term 2015-2016

Emil C Lupu & Daniele Sgandurra

e.c.lupu@imperial.ac.uk, d.sgandurra@imperial.ac.uk

Part 5 – Contents

Host-to-Host Communications

Datagrams transferred between hosts
- Network / Internet level in stack

Routed through networks based on IP addresses
- Source address of sending machine
- Destination address of recipient machine

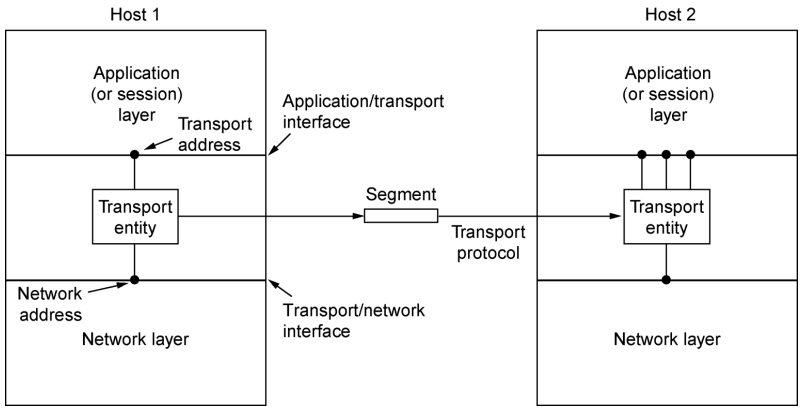
But:
- No identification of which applications send or receive
- Only unreliable connection-less datagram service

App-to-App Communications

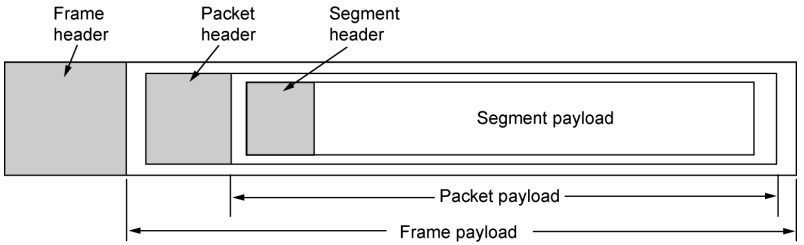
Identify application instances (processes)?
- Need well-understood identification
- Need to handle process restart
- Need to handle pools of processes serving clients
  - e.g. multi-threaded web server

Destination given by function or service
- One process may handle multiple services

The network, transport, and application layers



Nesting of segments, packets, and frames



Protocol Ports

Use **ports** to define end points
- Abstract addressing
- 16 bit unsigned integer: 0 - 65535

Processes specify ports to send to or receive from
- Use operating system calls to **bind** to port
- Packets have source and destination ports

Well Known Service Ports

Other systems know which port to address to
- Standard services usually run on well known ports

List of well known services (RFC 1060)

Ports 0 – 255: Internet Assigned Numbers Authority (IANA)

Ports 0 – 1023: Privileged UNIX standard services

Static file with service/port mappings

Unix: **`/etc/services`**

Windows: **`\windows\system32\drivers\etc\services`**

Some Assigned Ports

Port	Protocol	Use
20, 21	FTP	File transfer
22	SSH	Remote login, replacement for Telnet
25	SMTP	Email
80	HTTP	World Wide Web
110	POP-3	Remote email access
143	IMAP	Remote email access
443	HTTPS	Secure Web (HTTP over SSL/TLS)
543	RTSP	Media player control
631	IPP	Printer sharing

Less Well Known Addresses

Ports need not all be "well known"

Source port can be generated by application
- Useful when just needed for interaction

Destination port need not be well known
- Pass information to sender for novel application

Portmapper service (e.g., for Remote Procedure Calls)
- Process register (name, port) binding
- Client can query portmapper service by name.
- Portmapper (port 111)
- See remote procedure calls later on in the course

UDP/TCP Services

Two main transport layer protocols: **UDP** and **TCP**

Some services use UDP:
- **`bootp`** (on Windows), **`tftp`**, **`snmp`**, …

Some services use TCP:
- **`smtp`**, **`ftp`**, **`telnet`**, **`finger`**, **`http`**, …

Some services use either:
- **`echo`**, **`dns`**, **`ssh`**, **`bootp`** (on Linux), **`irc`**, …

Complete Addresses

The complete addressing for a datagram (UDP/TCP) describes the sender and receiver in such a way that the services which are communicating are fully defined:

Source IP address and source port
- Source port = return addr (may be omitted in UDP)
- e.g. **`columbia.doc.ic.ac.uk:57992`**

Destination IP address and destination port
- e.g. **`www.amazon.co.uk:80`**

Sockets

Socket: Abstract communications endpoint
- **Stream socket** (SOCK_STREAM) which uses TCP
- **Datagram socket** (SOCK_DGRAM) which uses UDP

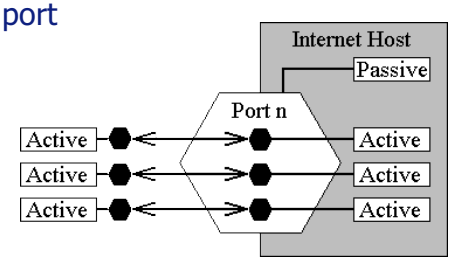
Active socket
- Connected to remote active socket via open connection
- Closing connection closes sockets at both ends

Passive socket
- Unconnected socket awaiting incoming connection
- Active socket spawned to handle connection
  - Passive socket goes back to waiting for new connections

Sockets and Ports

A socket is not a port (but they are related)

1+ sockets associated with 1 port
- At most one passive socket
  - Awaiting new connections
- Multiple active sockets
  - Handling open connections



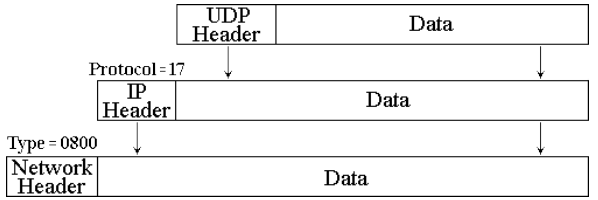
Hence 4-part addressing

i.e. (src IP, src port, dst IP, dst port)
- Many connections use same port
- Need to know where connection is coming from

The socket primitives for TCP

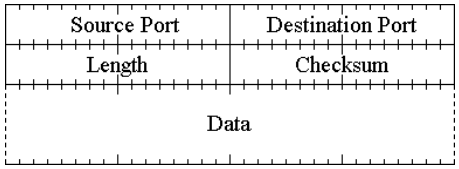
Primitive	Meaning
SOCKET	Create a new communication endpoint
BIND	Associate a local address with a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Passively establish an incoming connection
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEIVE	Receive some data from the connection
CLOSE	Release the connection

User Datagram Protocol (UDP)



UDP provides plain, IP-like service
- Connection-less datagrams, unreliable delivery, no sequence control, possible duplication
- Good for fast transfer with resilience to packet loss

UDP Header Format



Source port
- Optional, 0 if not used, reply-to port if used

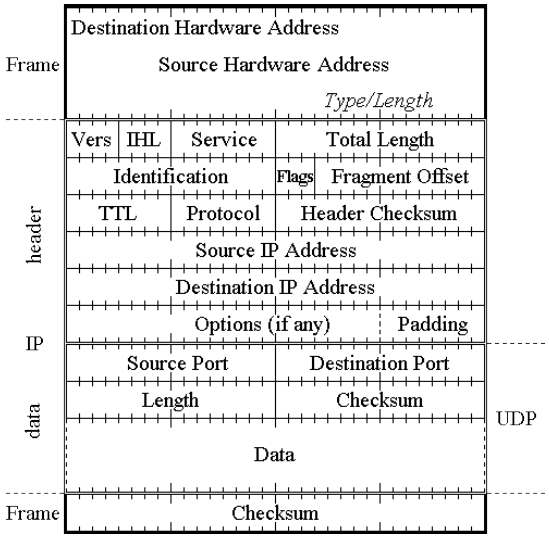
Destination port
- Host addressing still provided by IP headers

Length (in bytes)
- Includes header and data (min. 8 for no data)

UDP Checksum

Checksum (16 bit using 1s complement sum)
- Calculated over pseudo header + UDP header + data
- **Pseudo header** mimics IP header
  - Source IP address
  - Destination IP address
  - Protocol (17)
  - UDP length
    - Zeros to pad to multiple of two octets
- Allows detection of changed IP headers by gateways without actually duplicating data

UDP/IP Packet in Ethernet Frame



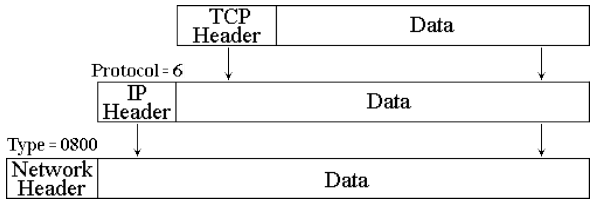
Reliable Service?

UDP is unreliable

Features of reliable a service
- Unstructured stream abstraction
- Virtual circuit connection
- Full-duplex connection

Could build reliable service over UDP
- Needs to keep track of successfully transmitted packets
- Add error-correcting & retransmission mechanisms

Transmission Control Protocol (TCP)



TCP adds a lot to IP:

- Streams with reliable delivery
- Full-duplex operation
- Flow control
- Network adaptation for congestion control
- Complexity & overheads

Connections

TCP uses **connection** as its basic abstraction

- Rather than just protocol port (receiving datagrams)

Connection-oriented service on top of connectionless IP

- Connection identified by pair of endpoints

Endpoint is (host, port) tuple: (IP addr:port)

e.g. src: `146.169.7.41:1069`
dst: `140.247.60.24:25`
Mail application on `146.169.7.41` connects to SMTP port on `140.247.60.24`

TCP Features

Streams

- TCP data is stream of bytes
- Underlying datagrams concealed

Sequence numbers for reliable delivery

- Used to maintain byte order in stream
- TCP detects lost data and arranges retransmission
- Stream delayed during retransmission to maintain byte sequence

TCP Features

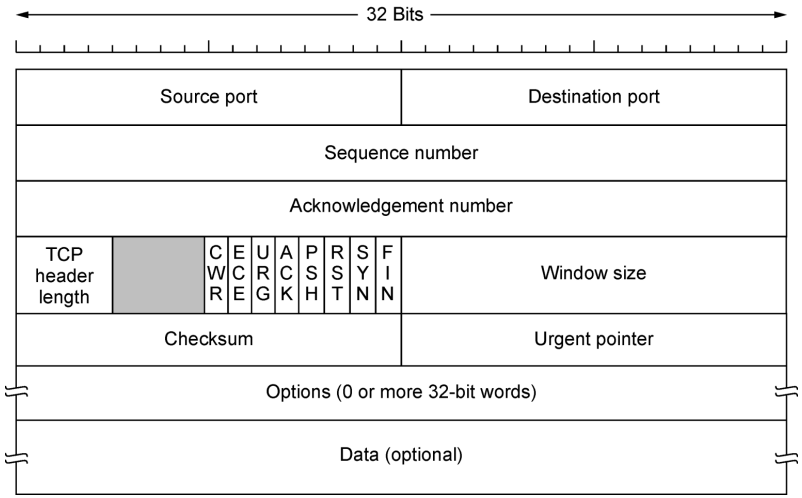
Flow control

- Manages data buffers and coordinates traffic to prevent overflows
- Fast senders have to pause for slow receivers to keep up

Congestion control

- Monitors and learns delay characteristics of network
- Adjusts operation to maximise throughput without overloading network

TCP Segment Format



Source and Destination Port

Local end points of the connection.

- TCP port + IP address = **48-bit unique end point.**

Source and destination end points together identify the connections.

- **5 tuple**: connection identifier (protocol, source IP and source port, destination IP and destination port).

TCP Fields: Sequence Numbers

Sequence number (seq num)

- Indicates position in stream of 1st data byte in segment

Acknowledgement number (ack num)

- Exploit full-duplex connection and use segments to piggyback acknowledgments
- Ack num = next seq num sender expects to receive
- Cumulative acknowledgments

TCP Fields: Data Sizes

Data offset (TCP Header Length):

- Number of 32-bit words in TCP header
- Needed because of variable length options

Window size (used for flow control)

- Num of data bytes which may be sent, starting with byte indicated by ack num
- Recipient should not send more than (window size – bytes sent) without acks (in transit)
- 0 means "no more data now, please"

TCP Fields: Control + Checksum

Control bits

URG: urgent pointer valid

ACK: ack num valid

PSH: "push" this segment (transmit promptly)

RST: reset connection

SYN: synchronise sequence numbers

FIN: sender has reached end of byte stream

Urgent pointer

- Pointer to high priority data in stream (e.g. error conditions)

Options

- Negotiate max segment size, scaling factor for window size, …

Checksum (16 bit using 1s complement sum)

- Same as for UDP with pseudo header

Passive and Active Opens

Both endpoints cooperate to open TCP connection

Passive open

- One end waits for incoming requests (server)

Active open

- Other end initiates communication (client)

Connection Control

Two hosts must synchronise seq nums

- Controls packet order and detects loss + duplicates

Use **SYN** segments to establish connection

- Establish initial sequence num (**ISN**)
- Stream positions are offsets from ISN
  - 1st data byte in segment = ISN + 1

ISN chosen randomly

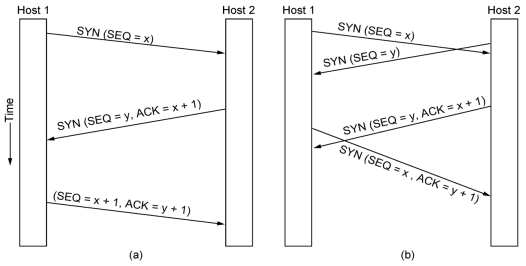
- Need to be unique over life-time of connection
- Starting at 0 bad idea because of old packets…

Connection Establishment

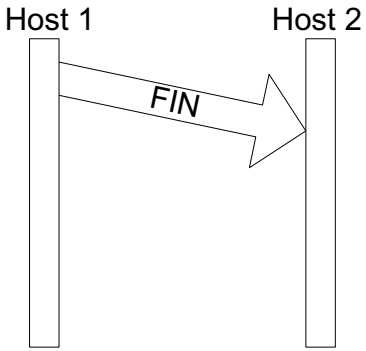
Three way handshake

- Establishes connection
- Sender and receiver agree on seq nums
- Works when two hosts establish connection concurrently

What happens when an old duplicate of the first message arrives?



Asymmetric Connection Release



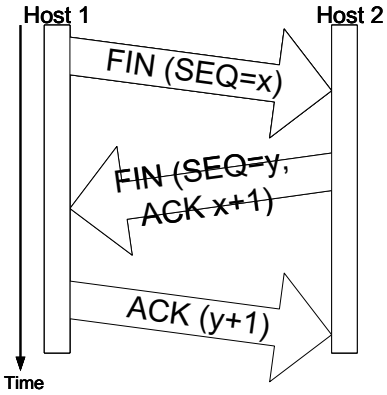
Unilateral close of connection

May loose data being transmitted

Symmetric Connection Release

Treat connection as two unidirectional connections to be released

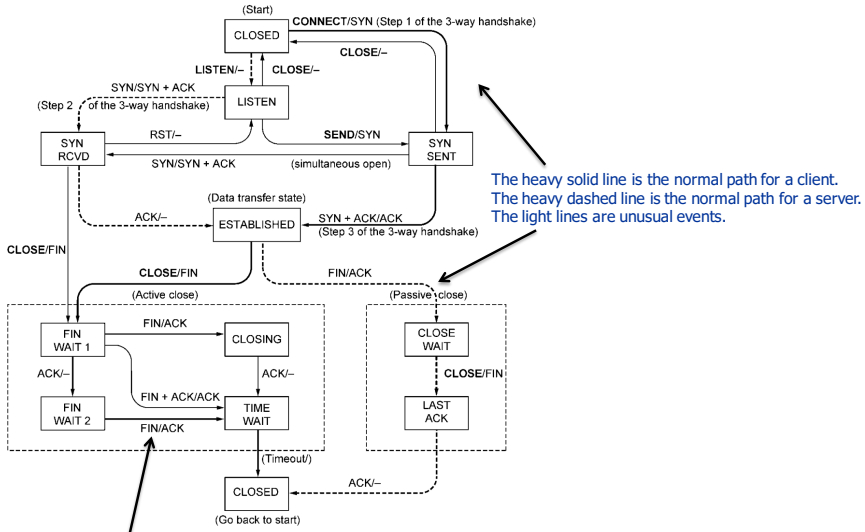
- Hosts agree on end seq nums
- Timeouts to handle lost messages



The states used in the TCP connection management finite state machine

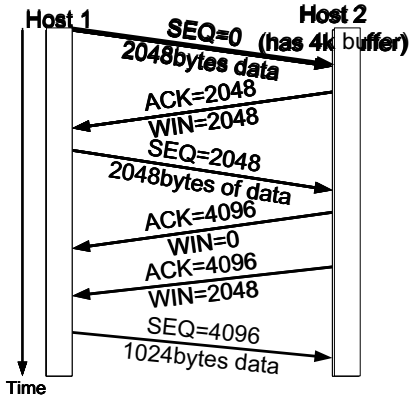
State	Description
CLOSED	No connection is active or pending
LISTEN	The server is waiting for an incoming call
SYN RCVD	A connection request has arrived; wait for ACK
SYN SENT	The application has started to open a connection
ESTABLISHED	The normal data transfer state
FIN WAIT 1	The application has said it is finished
FIN WAIT 2	The other side has agreed to release
TIME WAIT	Wait for all packets to die off
CLOSING	Both sides have tried to close simultaneously
CLOSE WAIT	The other side has initiated a release
LAST ACK	Wait for all packets to die off

TCP Connection Management State Machine



TCP Window Management

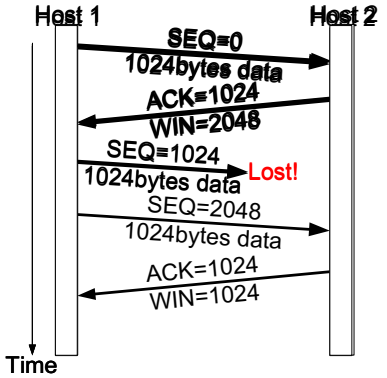
- Host 2 has 4k buffer
- Sent data controlled by WINdow field
- Sender does not have to fill receiver's buffer with each segment
- SEQuence and ACKnowledgement indicate what has been sent/received



Loss & Duplicate ACKs

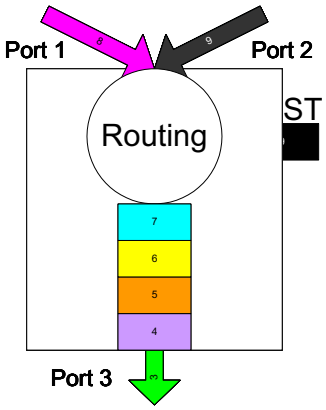
Duplicate ACK

- Generated when out-of-order segment received
- Notifies sender of duplicate and expected seq num
- Sender resends data if ACK takes longer than timeout or several duplicates received



Congestion

Congestion occurs in routers and in medium access



- Multiple incoming links can saturate single outgoing link
- Slower outgoing link can be saturated by one incoming link

Routers use store-forward

- Process each packet before sending
- If buffer becomes full, drops packets

TCP Congestion Control

Packet loss mostly due to congestion and not error

- Detect congestion by considering packet loss
- Change transmission rate to adapt to congestion

TCP sender maintains 2 windows

- **Receiver window** (flow control) and **congestion window**
- Uses whichever currently smaller

TCP Congestion Window

Congestion window based on network conditions

- Windows grows and shrinks based on packet loss
- Different algorithms for finding optimal size
  e.g. additive increase, multiplicative decrease

Requires efficient timeouts to detect loss

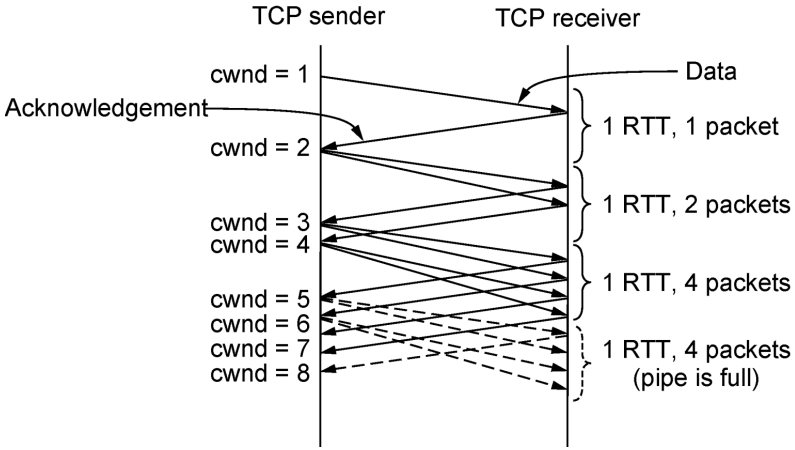
- TCP measures RTT and adjusts timeouts

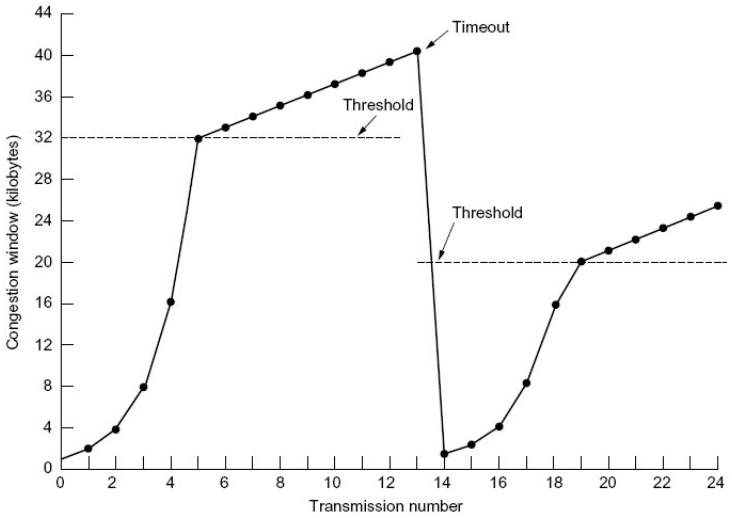
Lots of complexity to ensure throughput, fairness, …

Rough Behaviour

- Start with small window size.
- For each data segment acknowledged before time out increase window size by another segment until threshold.
- Increase linearly afterwards.
- For each packet lost reduce threshold by half.
- AIMD (additive-increase, multiplicative-decrease).

Slow start from an initial congestion window of one segment





Summary: UDP or TCP?

UDP

No need for reliability and error detection

- Message exchanges without transactional behaviour, e.g. DNS, DHCP
- Real-time apps, e.g. sensor monitoring, video streaming

Good for short communications

Efficient for fast networks

TCP

Need for reliability, error correction, flow and congestion control, or security

- Terminal sessions, e.g. SSH, Telnet
- Large data transfer, e.g. web, FTP, email

Efficient for long-lived connections

Requires more CPU time and bandwidth than UDP

Imperial College London

Computer Networks and Distributed Systems

Part 6 – Application Layer

Course 527 – Spring Term 2015-2016

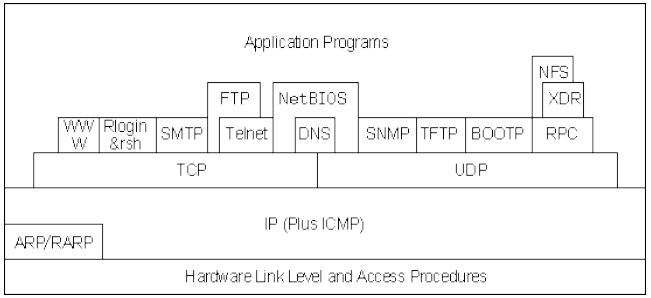
Emil C Lupu & Daniele Sgandurra

e.c.lupu@imperial.ac.uk, d.sgandurra@imperial.ac.uk

Part 6 – Contents

Application Layer

- Application level protocols
  - Domain name resolution (DNS)
  - Email (SMTP, POP, IMAP)
  - World Wide Web (HTTP)



Client

Initiates connection

Often user-invoked app running on local machine
- e.g. web browser, email clients

Uses service/resource from server

Resource use usually temporary

Server

Waits for connections
- Handles multiple clients

Special purpose app providing service/resource
- e.g. web server, chat server

Provides controlled access to resources/services

Often started at boot time
- e.g. "daemon" in UNIX; "service" under WIN

Peer-to-Peer

Not all communication client/server based

Peer-to-peer model
- Hosts operating together rather than one sided
- Offering of resources/service to one another
  - e.g. chat, file trading, Internet telephony

Often implemented as client/server on same host

Centralised server may coordinate

Application Layer Protocols

Clients, servers, peers need communication protocols

Web:	**HTTP**
Email protocols:	**SMTP, MIME, POP, IMAP**
File transfer:	**FTP**
Name resolution:	**DNS**

- Not user application but service

Names vs IP Addresses

IP addrs good for identification of network interfaces, but:

- Addr **`146.169.14.6`** not very memorable
- Machines, services may move between networks

Use meaningful, user-friendly names in addition to IP addrs

- e.g. **`columbia.doc.ic.ac.uk`** instead of **`146.169.7.41`**

Names assigned independently of IP addresses

- Names fixed when IP addrs change for technical reasons

Alias names can identify services

- Independent of machine providing service
- e.g. **`www.doc.ic.ac.uk`** for **`linnet.doc.ic.ac.uk`**

Mapping Names to IP Addrs

Need mapping between names and numbers

Local file
- Large and hard to maintain
- Was used in early ARPANET

Centralised server
- Clients query database to perform name resolution
- Bottleneck, single point of failure, admin hard

Distributed look-up system
- Domain Name System (DNS)

Domains Name System (DNS)

Internet is inter-network of autonomous networks

- Must avoid conflicts between names
- Must support independent administration of names

DNS names form hierarchies

- e.g. **`columbia.doc.ic.ac.uk`**

Requires uniqueness of complete name only

- Like postal address
- e.g. **`fred.foo.com`** different to **`fred.bar.com`**

Names and Domains: Externally Managed

Top-level structure conveys meaning

- `.uk` = United Kingdom
  - Country ID from global naming standard
- `.ac` = Academic network
  - Standard sub-domain within UK
- `.ic` = Imperial College
  - Name assigned by UK academic net administration
  - Owned by Imperial College

Within each domain naming managed independently

- e.g. `co.uk` is independent of `co.fr`

Top Level Domains (TLDs)

Domain	Specification	Example
`com`	Commercial	`ibm.com`
`net`	Network providers	`internic.net`, `demon.net`
`edu`	Educational institution	`mit.edu`
`gov`	Government organisation	`whitehouse.gov`
`mil`	US Military organisation	`navy.mil`
`org`	Other organisation	`linux.org`, `redcross.org`
<country code>	Domain administered by country (ISO 3166)	`fr`, `uk`, `zw`
2nd level country domains	Sub-domains to TLD administered by organisation for that country	`ac.uk`, `co.uk`

Name Assignment

ICANN authorises registrars for `.uk`, `.com`, `.org`

- Non-profit corp. with various Internet responsibilities

In UK, **Nominet** assigns names

- 2nd level names (co, ltd etc.) used by Nominet only
- TLD & 2nd level names not allowed as 3rd level names
  - `nhs.co.uk` not allowed as `nhs.uk` is 2nd level domain
  - `net.org.uk` not allowed as `.net` is TLD
- One character 3rd level names reserved
- `ac.uk`, `gov.uk`, `nhs.uk`, `police.uk`, `mod.uk` not controlled by Nominet

UK 2nd Level Domains

`.co.uk`	Commercial enterprises
`.me.uk`	Personal domains
`.org.uk`	Non-commercial organisations
`.plc.uk` `.ltd.uk`	Registered company names only
`.net.uk`	Internet Service Providers
`.sch.uk`	Schools
`.ac.uk`	Academic Establishments
`.gov.uk`	Government Bodies
`.nhs.uk`	NHS Organisations
`.police.uk`	UK Police Forces
`.mod.uk`	Ministry of Defence

Names and Domains: Internally Managed

Local domains follow organisational structure

- **`.doc`** = Dept. of Computing
  - Assigned by College admin
- **`.columbia`** = Machine in DoC
  - Assigned by Dept admin

Type of name may reflect type of machine, naming conventions.

- e.g. in DoC birds are servers, colours are printers

Names may reflect services machine provides

- e.g. **`www`**, **`mail`**

DNS Name Resolution

Host knows local DNS server to ask for names

- Local database of names maintained manually

Authoritative results

- Returned from name server managing that name

Local DNS servers do not move often

- May be hard-coded or given by DHCP
- DNS server known by its IP address
- Domain will typically have 2-3 DNS servers e.g., ns0, ns1 in doc.ic.ac.uk or ic.ac.uk.

Recursive DNS Lookups

If name cannot be resolved locally:

- Make new request to server up the hierarchy
- DNS server now becomes DNS client

When the top level is reached:

- Go down to required domain

Repeated until someone knows name,

- or decided that name not resolvable

DNS Servers are not required to support recursive queries.

Example: Recursive DNS Lookup

vm-shell1.doc.ic.uk → **www.csail.mit.edu**

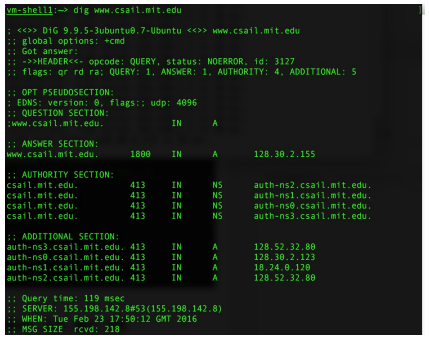
Vm-shell1.doc.ic.ac.uk -> **dns0.doc.ic.ac.uk**

dns0.doc.ic.ac.uk queries **ns.ic.ac.uk**

ns.ic.ac.uk queries **edu-server.net**

edu-server.net queries **mit.edu**

mit.edu queries **csail.mit.edu**



TLD servers replicated around the world

- Make queries faster that reach top of tree
- Load shared and gives redundancy

Iterative (non-recursive) DNS Lookups

```
vm-shell1:~ > dig +trace www.csail.mit.edu;
<<>> DiG 9.8.3-P1 <<>> +trace www.csail.mit.edu
;; global options: +cmd
.                       339347  IN      NS      j.root-servers.net.
…
;; Received 228 bytes from 192.168.1.1#53(192.168.1.1) in 25 ms

edu.                    172800  IN      NS      a.edu-servers.net.
…
;; Received 270 bytes from 199.7.83.42#53(199.7.83.42) in 112 ms

mit.edu.                172800  IN      NS      usw2.akam.net.
…
;; Received 414 bytes from 192.31.80.30#53(192.31.80.30) in 104 ms
csail.mit.edu.          1800    IN      NS      auth-ns2.csail.mit.edu.
...
;; Received 191 bytes from 193.108.91.37#53(193.108.91.37) in 11 ms
www.csail.mit.edu.      1800    IN      A       128.30.2.155
;; Received 51 bytes from 18.24.0.120#53(18.24.0.120) in 85 ms
```


DNS Caching

Servers cache responses as names often needed again

- Exploits locality of reference
  - If I look at a web site, my colleagues may look too
  - If **`ns.ic.ac.uk`** resolves **`www.doc.ic.ac.uk`** once, likely to get more requests in future

Cached answers non-authoritative

- May be wrong, out-of-date

DNS entries give TTL based on volatility

- Caches expire records after their TTL has expired
- Stable names can safely be cached for much longer

Types of DNS Records

A records: name ➔ IP address mapping

MX records: mail server address for domain

- Used when delivering email to, e.g., **`@doc.ic.ac.uk`**

NS records: name server address for domain

CNAME records: alias names

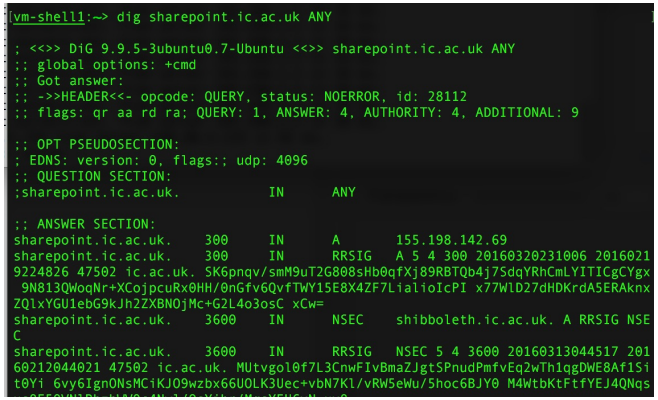
- e.g. **`www.doc.ic.ac.uk`** alias for **`linnet.doc.ic.ac.uk`**

PTR records: name ➔ IP addr (for reverse lookups)

- e.g. **`146.169.7.41`** ➔ **`columbia.doc.ic.ac.uk`**

RRSIG records: DNSSEC Signature

Example: DNS Host Lookup



```
vm-shell1:~> dig sharepoint.ic.ac.uk ANY

; <<>> DiG 9.9.5-3ubuntu0.7-Ubuntu <<>> sharepoint.ic.ac.uk ANY
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 28112
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 4, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;sharepoint.ic.ac.uk.           IN      ANY

;; ANSWER SECTION:
sharepoint.ic.ac.uk.    300     IN      A       155.198.142.69
sharepoint.ic.ac.uk.    300     IN      RRSIG   A 5 4 300 20160320231006 2016021
9224826 47502 ic.ac.uk. SK6pnqv/smM9uT2G808sHb0qfXj89RBTQb4j7SdqYRhCmLYITICgCYgx
 9N813QWoqNr+XCojpcuRx0HH/0nGfv6QvfTWY15E8X4ZF7LialioIcPI x77WlD27dHDKrdA5ERAknx
ZQlxYGU1ebG9kJh2ZXBNOjMc+G2L4o3osC xCw=
sharepoint.ic.ac.uk.    3600    IN      NSEC    shibboleth.ic.ac.uk. A RRSIG NSE
C
sharepoint.ic.ac.uk.    3600    IN      RRSIG   NSEC 5 4 3600 20160313044517 201
60212044021 47502 ic.ac.uk. MUtvgol0f7L3CnwFIvBmaZJgtSPnudPmfvEq2wTh1qgDWE8Af1Si
t0Yi 6vy6IgnONsMCiKJO9wzbx66UOLK3Uec+vbN7Kl/vRW5eWu/5hoc6BJY0 M4WtbKtFtfYEJ4QNqs
```

Answer of form:

`<domain name> <TTL> <class> <type> <value>`

TTL: Time to expiry if cached (in sec)

class: IN = Internet (almost always)

type: Type of record

Example: DNS Domain Lookup

```
vm-shell1:~> dig google.com ANY

; <<>> DiG 9.9.5-3ubuntu0.7-Ubuntu <<>> google.com ANY
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46860
;; flags: qr rd ra; QUERY: 1, ANSWER: 14, AUTHORITY: 4, ADDITIONAL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;google.com.                    IN      ANY

;; ANSWER SECTION:
google.com.             86400   IN      TYPE257 \# 19 0005697373756573796D616E74
65632E636F6D
google.com.             60      IN      SOA     ns2.google.com. dns-admin.google
.com. 115345670 900 900 1800 60
google.com.             3600    IN      TXT     "v=spf1 include:_spf.google.com
~all"
google.com.             600     IN      MX      50 alt4.aspmx.l.google.com.
google.com.             600     IN      MX      20 alt1.aspmx.l.google.com.
google.com.             600     IN      MX      30 alt2.aspmx.l.google.com.
google.com.             600     IN      MX      40 alt3.aspmx.l.google.com.
google.com.             600     IN      MX      10 aspmx.l.google.com.
google.com.             198     IN      AAAA    2a00:1450:4009:80f::200e
google.com.             150     IN      A       216.58.213.110
google.com.             87838   IN      NS      ns2.google.com.
google.com.             87838   IN      NS      ns1.google.com.
google.com.             87838   IN      NS      ns4.google.com.
google.com.             87838   IN      NS      ns3.google.com.

;; AUTHORITY SECTION:
google.com.             87838   IN      NS      ns1.google.com.
google.com.             87838   IN      NS      ns4.google.com.
google.com.             87838   IN      NS      ns2.google.com.
google.com.             87838   IN      NS      ns3.google.com.

;; ADDITIONAL SECTION:
ns2.google.com.         88517   IN      A       216.239.34.10
ns3.google.com.         88517   IN      A       216.239.36.10
ns1.google.com.         88517   IN      A       216.239.32.10
ns4.google.com.         88517   IN      A       216.239.38.10

;; Query time: 12 msec
;; SERVER: 155.198.142.8#53(155.198.142.8)
;; WHEN: Tue Feb 23 17:54:38 GMT 2016
;; MSG SIZE  rcvd: 508
```

Electronic Mail

Movement of structured text msg between systems

Email message fields:

To:, Cc:, Sender:, Reply-to:	Email addresses
From:	Who sent the message
Received:	List of mail servers that processed mail
Subject:	Summary of message
Message-Id:	Unique id for message

- Detect duplicate messages, responses

Email address: **`<user name>@<mail domain name>`**

- e.g. **`ecl1@doc.ic.ac.uk`**
- Mail server for domain found through DNS MX records

Email Subsystems

Mail User Agents (**MUA**)

- e.g. Thunderbird, Outlook, mutt, …
- Email client for composition, display, filing, …

Message Transfer Agents (**MTA**)

- e.g. sendmail, exim, Exchange, …
- Mail server that handles message transfer

Email often sent from MUA to **relay** (or **gateway**) MTA

- Usually in same organisation as sender
- Delivers to MTA in recipient's domain
  - Possibly via other relays
- Holds (spools) mail if destination unreachable
  - Supposed never to lose mail

Simple Mail Transfer Protocol (SMTP)

Protocol used by MTAs (and MUAs)

- Delivers mail from one system to another

Two parties communicate using TCP and port 25

- Sender (client) & Receiver (server)

Three basic steps

1. Start session
2. Exchange data
3. Complete session

SMTP: Protocol Steps

Open TCP connection to port 25

Server responds with `220` message

Client sends `HELO` command identifying its domain

Client sends 1+ mail messages

- First giving identity information (`MAIL-FROM`, `RCPT-TO`)
- Then sending `DATA` part
- Server responds to each message with 3-digit code

`QUIT` message tells server to close TCP connection

Example: SMTP Session

Often direct connection hosts no longer allowed

```
220 finch.doc.ic.ac.uk ESMTP Exim 4.63 Wed, 03 Jan 2006
  16:18:37 +0000
helo doc.ic.ac.uk
250 finch.doc.ic.ac.uk Hello columbia.doc.ic.ac.uk
  [146.169.7.41]
mail from: prp@doc.ic.ac.uk
250 <prp@doc.ic.ac.uk> is syntactically correct
rcpt to: prp@doc.ic.ac.uk
250 <prp@doc.ic.ac.uk> is syntactically correct
data
354 Enter message, ending with "." on a line by itself
subject: test
1 2 3
.
250 OK id=16l78a-0001cK-00
quit
221 finch.doc.ic.ac.uk closing connection
```


SMTP: Basic Sender Commands

Command	Argument	Meaning
HELLO/HELO	Sender's domain	I'm in this domain (must be first command)
MAIL FROM:	User id	Identify sender
RCPT TO: (1 or more)	User id	Identify recipient(s) (first must follow MAIL (or other starting command))
DATA		Email data follows (must follow RCPT)
<CRLF>.<CRLF>		End of email text (line with just full-stop)
RESET/RSET		Abort current mail
VERIFY/VRFY	User id	Is user ID valid?
QUIT		Sender signing off (last command)

SMTP: Basic Receiver Replies

Reply Code	Meaning
220	Service ready
221	OK, I too am closing connection
250	OK, requested mail action completed
354	Start sending me email text, end with <CRLF>.<CRLF>
500	Syntax error – command unrecognised
501	Syntax error in parameters or arguments
503	Bad sequence of commands
550	Requested action not taken: mailbox unavailable
552	Requested mail action aborted: exceeded storage allocation

Example: Delivered Message

```
Return-path: <prp@doc.ic.ac.uk>
Envelope-to: prp@doc.ic.ac.uk
Delivery-date: Wed, 02 Jan 2006 11:42:00 +0000
Received: from [146.169.7.46] (helo=finch.doc.ic.ac.uk
 ident=exim) by falcon.doc.ic.ac.uk with esmtp (Exim
 3.13 #8) id 16l78q-0004M7-00 for prp@doc.ic.ac.uk; Wed,
 02 Jan 2006 11:42:00 +0000
Received: from columbia.doc.ic.ac.uk ([146.169.7.41]
 helo=doc.ic.ac.uk) by finch.doc.ic.ac.uk with smtp
 (Exim 3.16 #7) id 16l78a-0001cK-00 for
 prp@doc.ic.ac.uk; Wed, 02 Jan 2006 11:41:59 +0000
Subject: test
Message-ID: <E16l78a-0001cK-00@finch.doc.ic.ac.uk>
From: prp@doc.ic.ac.uk
BCC:
Date: Wed, 02 Jan 2006 11:41:59 +0000

1 2 3
```


Post Office Protocol (POP)

SMTP designed for permanently available hosts

- e.g. Internet mail servers
- SMTP delivers mail to mailbox at ISP
- Not usually used to deliver mail to user's desktop

Post Office Protocol allows user access to mailbox

- POP client (MUA) connects to mailbox (POP) server
- Connection to port 110 when "get mail" is requested
- POP authenticates user (with password)
- Mailbox downloaded for processing

Internet Message Access Protocol (IMAP)

POP only supports downloads of Inbox

IMAP handles multiple folders

- Possible to leave emails on mail server
- Useful when using multiple MUAs or machines

Other features

- Partial downloads of emails/attachments
- Offline mode when unconnected
- Server-side searches

But IMAP complex and adds server load

File Transfer Protocol (FTP)

Used to exchange files between hosts

- Optimised for efficient transfer of large files
- Separate control and data connections
  - Uses port 21 to initiate control connection
- Data can be
  - fetched by client (GET)
  - sent to server (PUT)

Typical FTP client exposes messages exchanged

- Some command capabilities but not designed to be terminal program

```
vm-shell1:~> sftp ecl1@shell2.doc.ic.ac.uk
WARNING : Unauthorized access to this system is forbidden and will be
prosecuted by law. By accessing this system, you agree that your actions
may be monitored if unauthorized usage is suspected.

   ***DO NOT RUN RESOURCE-INTENSIVE PROCESSES ON THE SHELL SERVERS***
Connected to shell2.doc.ic.ac.uk.
sftp> !ls
Desktop      IMAP_backup  Microsoft    Templates
desktop.ini  KDesktop     Pictures     testftp.txt
Documents    lib          Public       WINDOWS
GNUstep      Macromedia   public_html  WindowsDocuments
sftp> get testftp.txt
Fetching /homes/ecl1/testftp.txt to testftp.txt
/homes/ecl1/testftp.txt                       0%    0     0.0KB/s   --:-- ETA
sftp> put testftp.txt
Uploading testftp.txt to /homes/ecl1/testftp.txt
testftp.txt                                 100%    0     0.0KB/s   00:00
sftp> bye
vm-shell1:~>
```


SMTP vs. FTP

Information transfer from client to server

Information transfer in either direction

Single connection for control/data

Separate control and data connections

Message transfer protocol

- Server file system invisible

File transfer protocol

- File system (partly) visible

ASCII/MIME encoded data

- Data typed with MIME

ASCII/binary data

- Data un-typed

Multiple "hops" to perform end-to-end transaction

- Spooling when server unavailable

Direct client/server interaction

- Interactive protocol - fails if server unavailable

World Wide Web (WWW)

Developed by Tim Berners-Lee (CERN in 1989)

Killer application for the Internet

- Supports transfer and display of documents
- Documents include multimedia content and hyperlinks
- Important lesson for HCI…

Client/server based

- Client (browser): Firefox, Explorer, Opera, lynx, …
- Web servers: Apache, IIS, …

Web Standards

World Wide Web Consortium (W3C) manages standards:

HyperText Transfer Protocol (**HTTP**)

- Used by browsers to get resources from servers

HyperText Mark-up Language (**HTML**)

- Encoding of data for web pages
- Supports text with images, formatting and hyperlinks

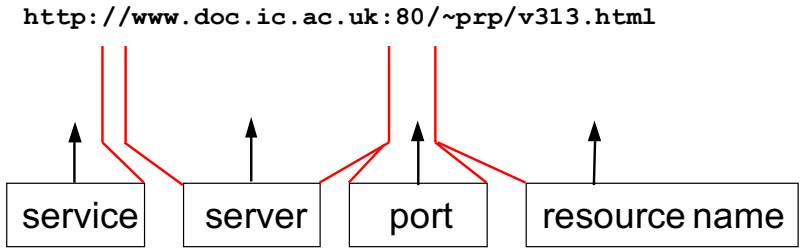
Uniform Resource Locator (**URL**)

- Used to identify links to resources on servers

Uniform Resource Identifier (**URI**)

- More general, newer version of URLs
- URIs can also be location independent (pure names)

Uniform Resource Locators (URLs)



URLs can indicate:

- other protocols, e.g. ftp, gopher, telnet, mail, news, file, …
- other resource types, e.g. image, program, service …

URLs encode location of resource

Hypertext Transfer Protocol (HTTP)

HTTP server usually listens at TCP port 80

Stateless, transaction-oriented protocol

1. Client opens connection to server
2. Request sent from client to server
3. Server responds
4. Connection closed

HTTP Request Message

Request format:

Request line (method, identifier, version)
Header (additional info)
Body (data)

Retrieve web page from server:

```
GET /index.html HTTP/1.0
User-Agent: Mozilla/2.01 (X11; I; IRIX 5.2 IP7)
Accept: image/gif, image/x-bitmap, image/jpeg
/* a blank line */
```


HTTP/1.0 Methods

HTTP 1.0 has **methods** rather than commands

GET: Client requests resource from server

- No permanent action on server is implied
- Most common method

HEAD: Requests only header of web page

- Useful when deciding if changed

POST: Append/send data to named resource

- Used to submit client data from web forms

Others often not implemented:

PUT, DELETE, LINK, UNLINK

HTTP Reply Messages

Reply format:

Status line (version, code, optional message)

Header (additional info)

Body (data, MIME compatible)

Sample response:

```
HTTP/1.0 200 OK
MIME-Version: 1.0
Server: CERN/3.0
Date: Wednesday 10-Apr-96 03:59:47 GMT
Content-type: text/html
Content-length: 2168
Last-Modified: Friday 06-Oct-95 07:16:52 GMT
/* a blank line */
/* HTML text of the Web page */
```


HTTP Reply Codes

200	OK
204	No response
301	Moved (permanently)
400	Bad request (e.g. syntax error)
401	Unauthorised
402	Payment required
404	Not found
503	Service unavailable
505	HTTP version unsupported

Stateless and Non-Persistent

HTTP protocol is stateless

- Server doesn't keep track of client requests
  - Simplifies server design
- But websites would like to keep track of customers

HTTP/1.0 uses non-persistent connections

- New TCP connection opened for every request
  - Page with 10 images will open 11 TCP connections
- Adds server load and setup costs and delays
- TCP congestion control inefficient for short transfers

Maintaining State: Cookies

Websites want to identify users

- IP addrs not practical ids for users due to NAT, sharing

Persistent information through **cookies**

- Allows server to maintain state between HTTP requests
- Sent by web server and stored by browser
  - Name/value pair with expiry time
  - Set-Cookie header in HTTP response
  - Javascript (setCookie/getCookie)

Persistent Connections: HTTP/1.1

Deals with some of HTTP/1.0's performance issues

- **Persistent connections**
  - Client opens TCP connection
  - HTTP requests pipelined through this connection
  - Multiple requests with one TCP open/close overhead
- Other features:
  Better proxy handling; more methods; improved encoding and authentication

Now in widespread use

- Easy transition because backwards compatible

Further Information On The Web

World Wide Web Consortium (W3C)

- `www.w3.org` website
- Much information, tutorials, standards definitions for HTTP, HTML, XML, DOM, CSS, RDF etc.

Practical guides in the O'Reilley books