University of Glasgow

School of Computing Science

Introduction to Networked Systems

Networked Systems (H)

Lecture 1

Lecture Outline

- Course Administration
  - Aims, Objectives, Intended Learning Outcomes
  - Course Outline
  - Labs and Assessment
  - Reading List
- Introduction to Networks
  - Concepts
  - Protocols
  - Layering
  - Standards

Course Administration

Contact Details and Website

- Lecturers
  - Dr Colin Perkins (Glasgow) and Dr Ian Thng (Singapore)
  - No scheduled office hours – make appointments by email to discuss the course outside scheduled lecture or lab times if necessary
- Lecture notes and other material on online:
  - https://csperkins.org/teaching/2017-2018/networked-systems/
    (or on the School's Moodle site)
  - Paper handouts will not be provided – the act of taking notes helps learning

Aims and Objectives

- To introduce fundamental concepts and theory of communications
- To provide a solid understanding of the technology that supports modern networked computer systems
- To introduce low-level network programming, and give students practice with systems programming in C
- To give students the ability to evaluate and advise industry on the use and deployment of networked systems

Intended Learning Outcomes

- By the end of the course, you should be able to:
  - Describe and compare capabilities of various communication technologies and techniques;
  - Know the differences between networks of different scale, and how these affect their design;
  - Describe the issues in connecting heterogeneous networks;
  - Describe importance of layering, and the OSI reference model;
  - Understand demands of different applications on quality of service requirements for the underlying communication network;
  - Demonstrate an understanding of the design and operation of an IP network, such as the Internet, and explain the purpose and function of its various components; and
  - Write simple low-level communication software, showing awareness of good practice for correct and secure programming

Course Outline

Week	Lecture Slot 1	Lecture Slot 2	Laboratory Session
1	Introduction to Networked Systems		The Berkeley Sockets API
2	Physical and Data Link Layers		TCP/IP Networking in C – A Simple Web Server **(Assessed via exam)**
3	Bridging		
4	Internetworking		
5	Intra-domain Routing		Understanding the Topology of the Internet **(Assessed: 20%)**
6	Inter-domain Routing	The Transport Layer	
7	TCP and Congestion Control		
8	UDP and Network Address Translation		
9	Security Considerations		UDP/IP Networking in C
10	Higher-layer Protocols		

The aims of the labs are to improve your understanding of the network, of network programming using the Sockets API, and to practice systems programming in C.

Assessment

- Assessed exercises: 20%
  - Mixture of formative and summative exercises
  - Don't leave summative exercise to the last minute: they're designed to be completed over several weeks, allowing time for thought and reflection on the material, and are too large to complete in a rush in a couple of days
- Examination: 80%
  - Exam format: answer all three questions

Assessment of Coursework (1)

- The coursework is intended to improve your C programming skills, as well as your understanding of networks
  - The marking scheme will assess the C code you submit, in addition to your understanding of networking
    - Code that is over-complex, convoluted, or difficult to follow will receive fewer marks than code that gives identical results, but that is cleanly structured and easy to understand
    - Assessment explicitly targets C code quality and correctness; marks will be deducted for poor style, bugs, and security problems – even if the submission gives the correct result
  - Note that networked code often exhibits bugs that are difficult to demonstrate in small-scale laboratory tests, but can be found by expert inspection:
    - Race conditions due to use of POSIX threads with incorrect locking
    - Race conditions due to passing socket file descriptors incorrectly
    - Buffer overflows or other security vulnerabilities
    - …

  Just because a program works in your tests doesn't necessarily mean it's bug free – your tests may be insufficient to show the bug

Assessment of Coursework (2)

- Student feedback from a previous year:

  "It seems to me that the assessment was more of an exercise in writing perfect C code than understanding networked systems"

- Today's network is an *extremely* hostile environment
- Writing networked systems that are both secure and safe to deploy *is* "an exercise in writing perfect C code" – if you learn nothing else from this course, that understanding is crucial

Assessment of Coursework (3)

- The University code of assessment specifies penalties for late submission, and for non-adherence to submission instructions
- These penalties will be strictly applied
- If you have special circumstances that will affect your submissions, you must inform the lecturer *before* the deadline

University of Glasgow | School of Computing Science

VIA VERITAS VITA

Assessed Coursework

Course Name					
Coursework Number					
Deadline	**Time:**	**4:30pm**	**Date:**		
% Contribution to final course mark			**This should take this many hours:**		
Solo or Group ✓	**Solo**		**Group**		
Submission Instructions					
Who Will Mark This? ✓	**Lecturer**		**Tutor**		**Other**
Feedback Type? ✓	**Written**		**Oral**		**Both**
Individual or Generic? ✓	**Generic**		**Individual**		**Both**
Other Feedback Notes					
Discussion in Class? ✓	**Yes**		**No**		
Please Note: This Coursework cannot be Re-Done					

Code of Assessment Rules for Coursework Submission

Deadlines for the submission of coursework which is to be formally assessed will be published in course documentation, and work which is submitted later than the deadline will be subject to penalty as set out below. The primary grade and secondary band awarded for coursework which is submitted after the published deadline will be calculated as follows:

(i) in respect of work submitted not more than five working days after the deadline
  a. the work will be assessed in the usual way;
  b. the primary grade and secondary band so determined will then be reduced by two secondary bands for each working day (or part of a working day) the work was submitted late.
(ii) work submitted more than five working days after the deadline will be awarded Grade H.

Penalties for late submission of coursework will not be imposed if good cause is established for the late submission. You should submit documents supporting good cause via MyCampus.

Penalty for non-adherence to Submission Instructions is 2 bands

You must complete an "Own Work" form via http://www.dcs.gla.ac.uk/socs-online for all coursework UNLESS submitted via Moodle

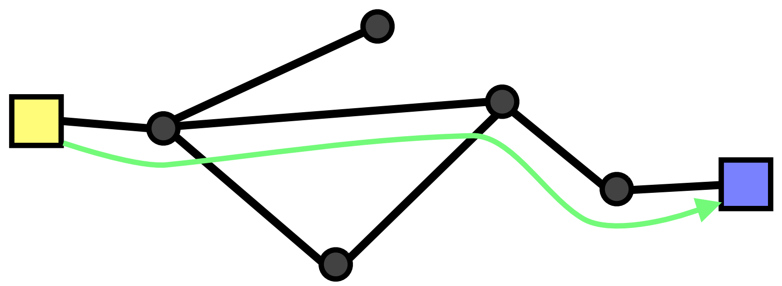
Required Reading

- Any good text on computer networks, for example:
  - Peterson and Davie, Computer Networks: A Systems Approach, 5th Edition, Morgan Kaufman, 2011, ISBN 0123851386
  - Kurose and Ross, Computer Networking: A Top-Down Approach, 6th Edition, Addison-Wesley, 2012, ISBN 0273768964
  - Tanenbaum and Wetherall, Computer Networks, 5th Edition, Prentice Hall, 2010, ISBN 0132553171
  - Bonaventure, Computer Networking: Principles, Protocols and Practice, online textbook (http://cnp3book.info.ucl.ac.be/index.html)
- You are expected to read-along with the lectures – the lectures introduce the concepts, and the books provide detail

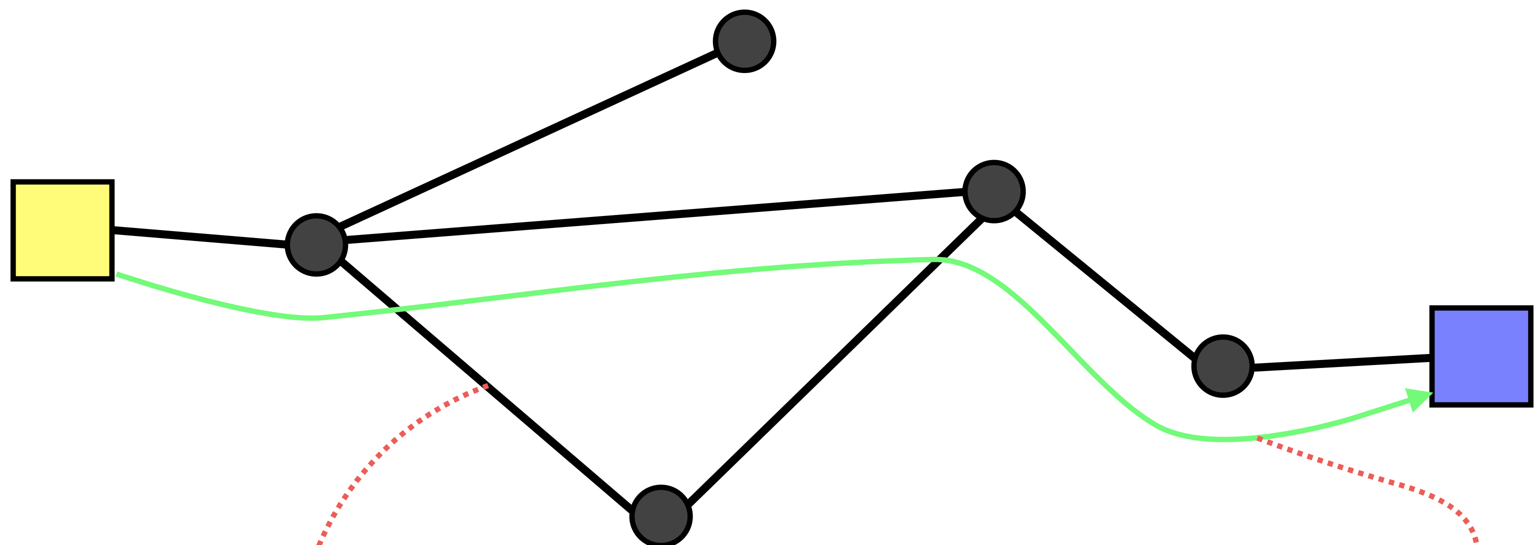
Introduction to Networked Systems

Networked Systems

- Autonomous computing devices that exchange data to perform some application goal
  - The exchange of data is explicitly visible to the application – the system is aware of the network
  - Applications using the Internet is one example, but other networks in widespread use:
    - Digital broadcast TV (e.g., FreeView in the UK)
    - Mobile voice telephony
    - Controller area networks connecting sensors and other components within vehicles or aircraft
    - Sensor networks
    - …



Networked Systems



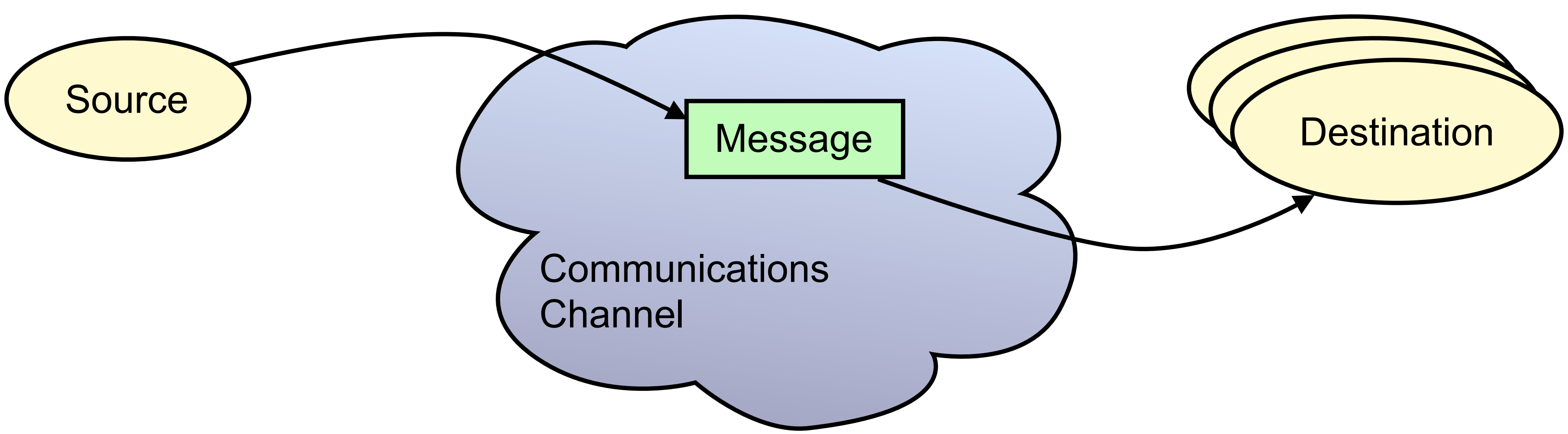
Networked System
– how do systems communicate across the network?

Networking
– how are links interconnected to build a wide-area network?

Communication
– how is information exchanged across a single link?

Communication

- Messages transferred from source to destination(s) via some communications channel
  - Size of messages might be bounded
  - Communication might be simplex, half- or full-duplex



Information

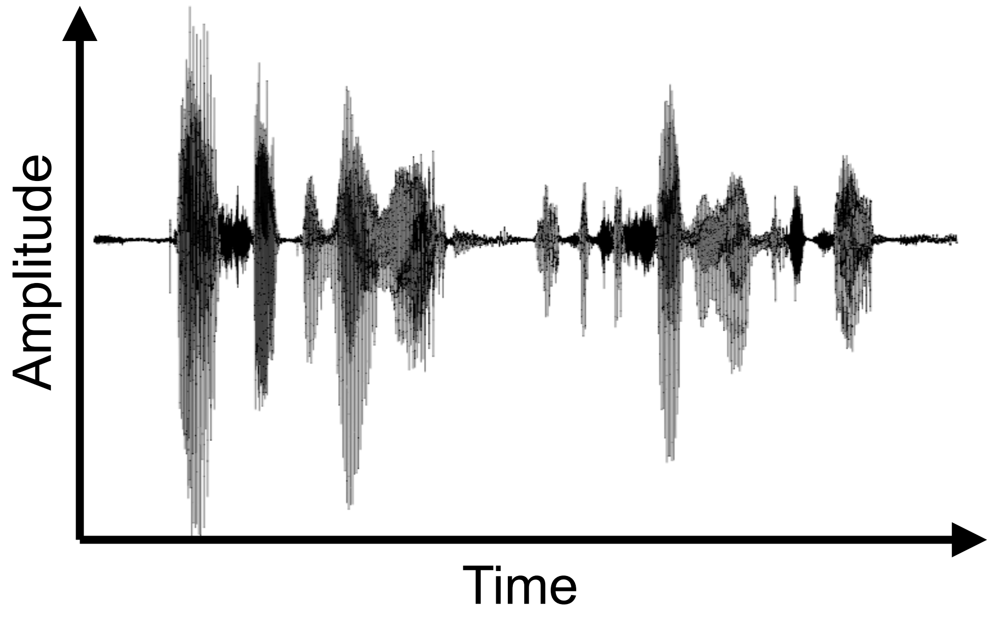
- Messages convey *information*
  - The amount of information in a message can be characterised mathematically – *Information Theory*
- Capacity of channels to convey information can also be modelled
  - How much? How fast? How much power used?
  - Physical limits exist on the capacity of a channel

Signals

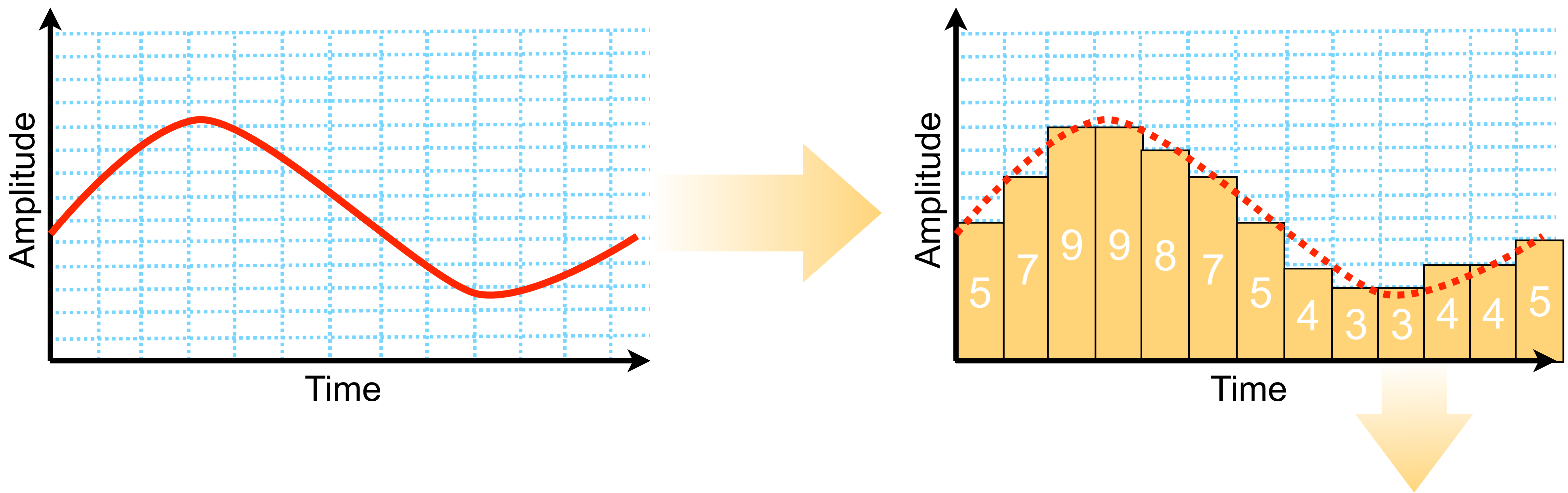
- Physical form of a message is a *signal*
  - May be a material object (carrier pigeon, CD, …)
  - Usually a wave (sound, electrical signal, light, radio, …)
- Signal may be analogue or digital
  - Analogue: a smooth continuum of values
  - Digital: a sequence of discrete *symbols*
    - Mapping information to symbols is known as *coding*

Analogue Signals

- Simplest analogue signal: amplitude directly codes value of interest
    - AM Radio, analogue telephones
- Can be arbitrarily accurate
- Susceptible to noise and interference on channel
- Difficult to process with digital electronics



Analogue Signals



Any analogue signal can be represented digitally: *sample* the signal at a suitable rate, *quantise* to nearest allowable discrete value, and convert to digital representation

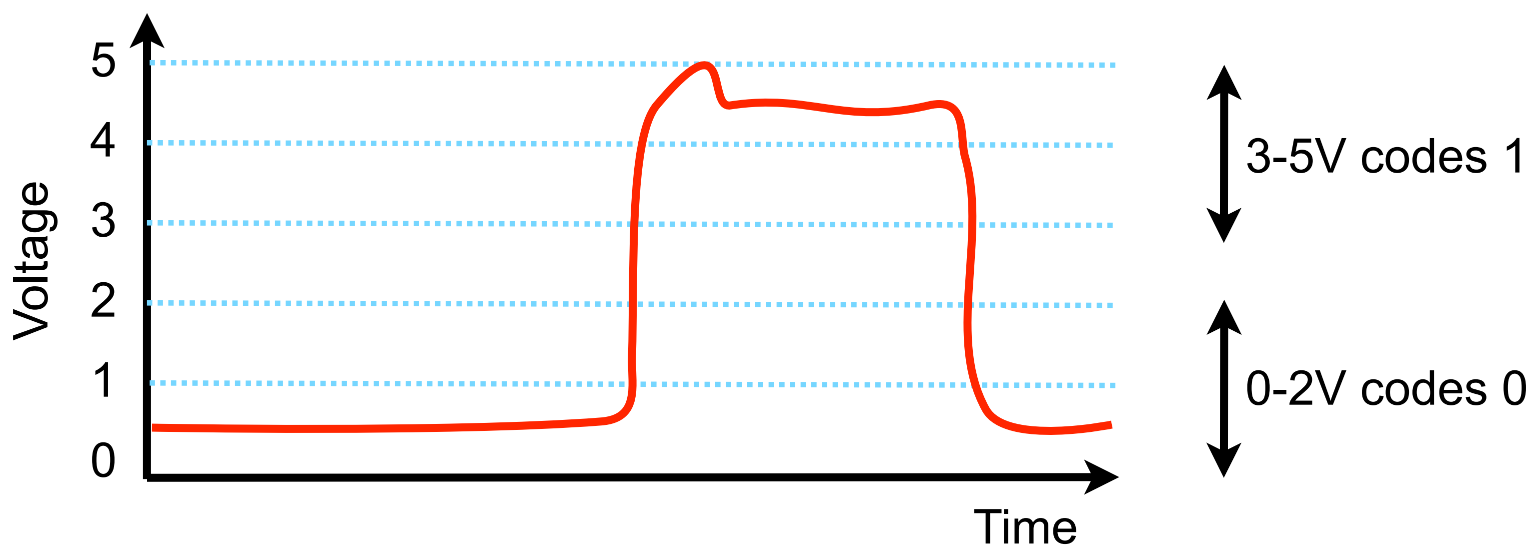
- The *sampling theorem* determines the rate at which the signal must be sampled for accurate reconstruction (→ lecture 2)

0101
0111
1001
1001
1000
...

Digital Signals

- Digital signals comprise a sequence of discrete symbols – fixed alphabet, not arbitrary values
  - But underlying channel is almost always *analogue*
  - Modulation used to map a digital signal onto the channel (→ lecture 2)

  - Example: non-return to zero modulation:



Digital Signals: Baud Rate

- Computing systems use *binary* encoding
  - The digital signal comprises two symbols: 0 and 1
- Networked systems often use non-binary encoding
  - Example: wireless links frequently use complex modulation schemes with either 16, 64, or 256 possible symbols (→ lecture 2)
  - Number of symbols transmitted per second is the *baud rate* – and can differ from the bit rate

Émile Baudot (1845-1903)

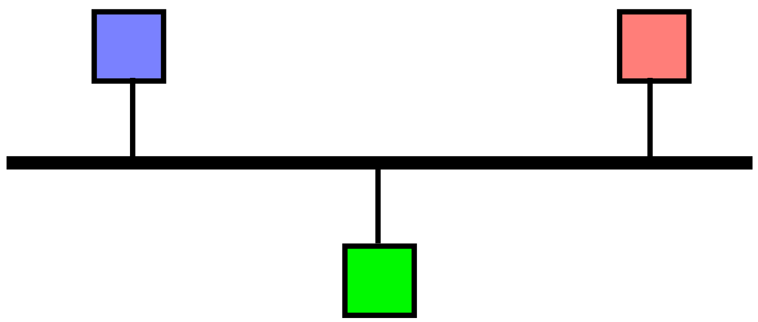
Channels and Network Links

- A signal is conveyed via a channel
  - May be directly conveyed – electrical signals in a cable
  - May be modulated onto an underlying carrier – radio

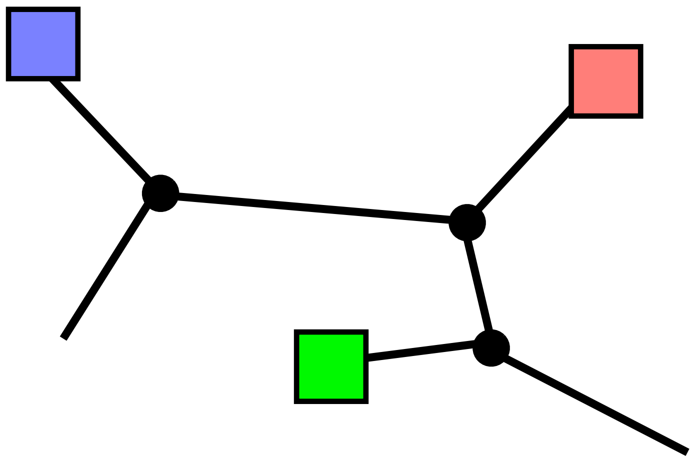
- The combination of signal and channel forms a link

From Links to Networks

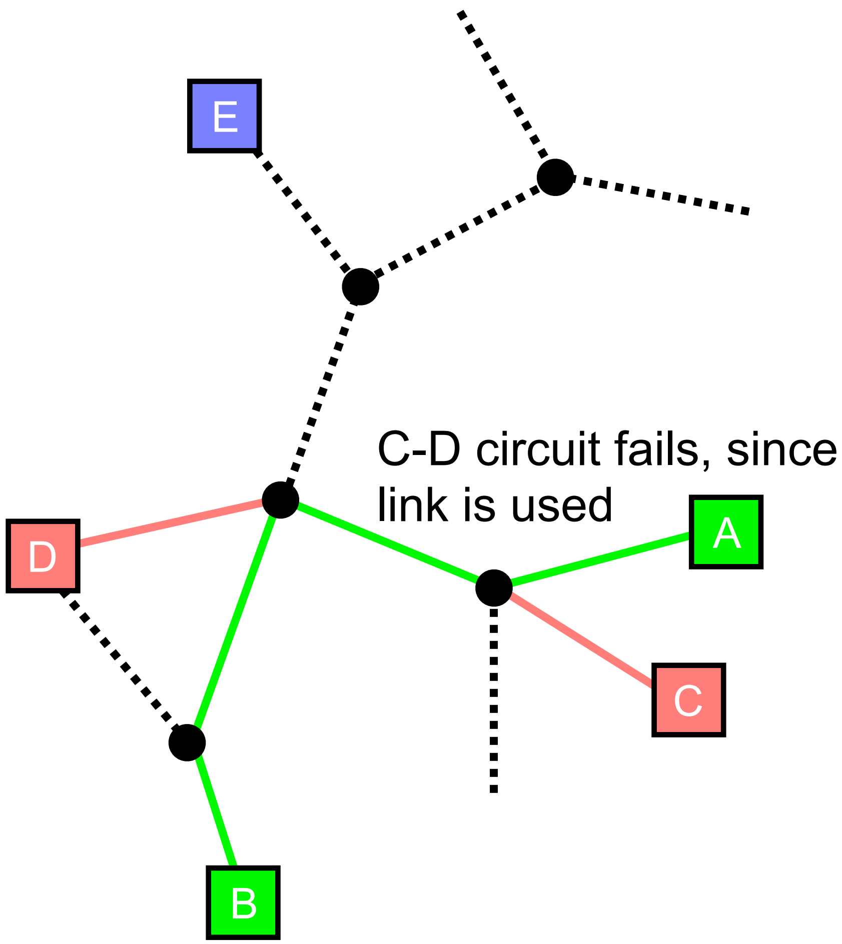
- A link directly connects one or more hosts



- A network comprises several links connected together
  - The devices connecting the links are called either *switches* or *routers* depending on the type of network

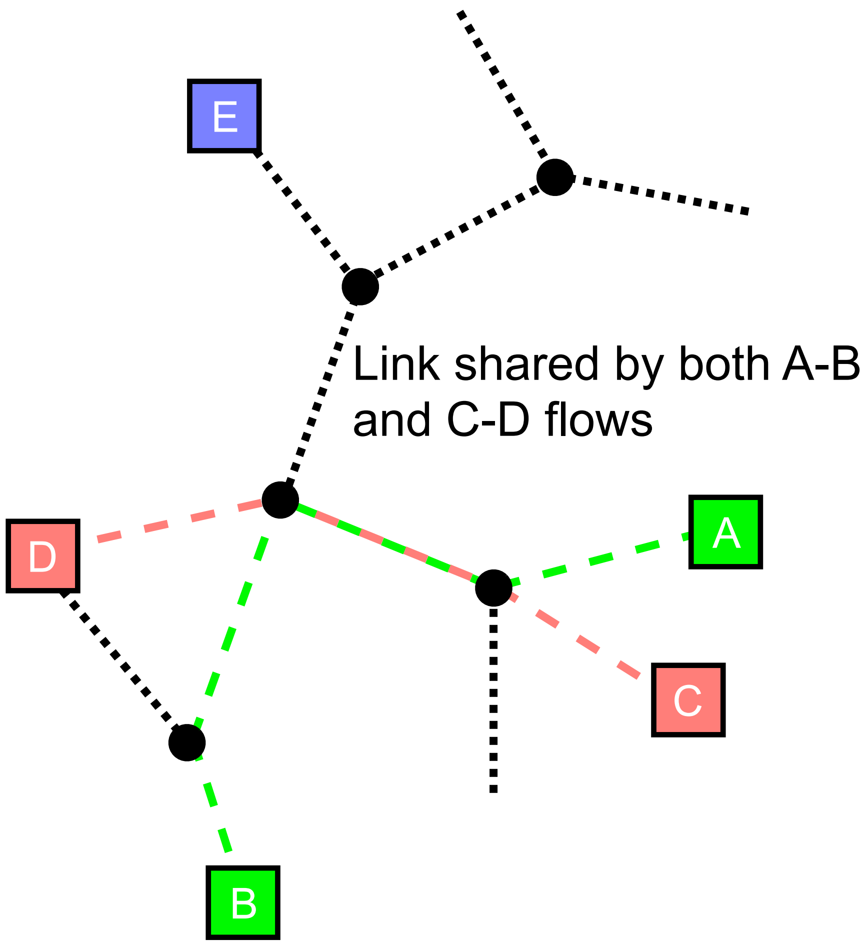


Circuit Switched Networks



- A dedicated *circuit* can be set up for A and B to communicate
  - A and B exchange arbitrary length messages
  - Guaranteed capacity once circuit is created
  - But – the dedicated circuit can block other communications (e.g. the C to D path); the capacity of the network gives the blocking probability
  - Example: traditional telephone network

Packet Switched Networks



- Alternatively, messages can be split into small *packets* before transmission
  - Allows A-B and C-D to communicate at the same time, sharing the bottleneck link
  - Connectivity guaranteed, but the available capacity varies depending how many other people are using the network
  - Packets are small, and have a size constraint; a message can consist of many packets
  - Example: the Internet

Networked Systems

- All networked systems built using these basic components:
  - Hosts – the source and destination(s)
  - Links – physical realisation of the channel, conveying messages
  - Switches/routers – connect multiple links
- Layered on top are *network protocols* which give meaning to the messages that are exchanged

Protocols and Layers

Network Protocols

- Communication occurs when two (or more) hosts exchange messages across a network
- To be meaningful, the messages need follow some well known *syntax*, and have agreed *semantics*
  - A *network protocol* is an agreed language for encoding messages, along with the rules defining what messages mean and when they can be sent
    - c.f. a programming language, where syntax and semantics define the legal programs
  - Numerous network protocols exist; some operate between hosts, some between routers, and some between hosts and routers
  - The protocols define the behaviour of the network

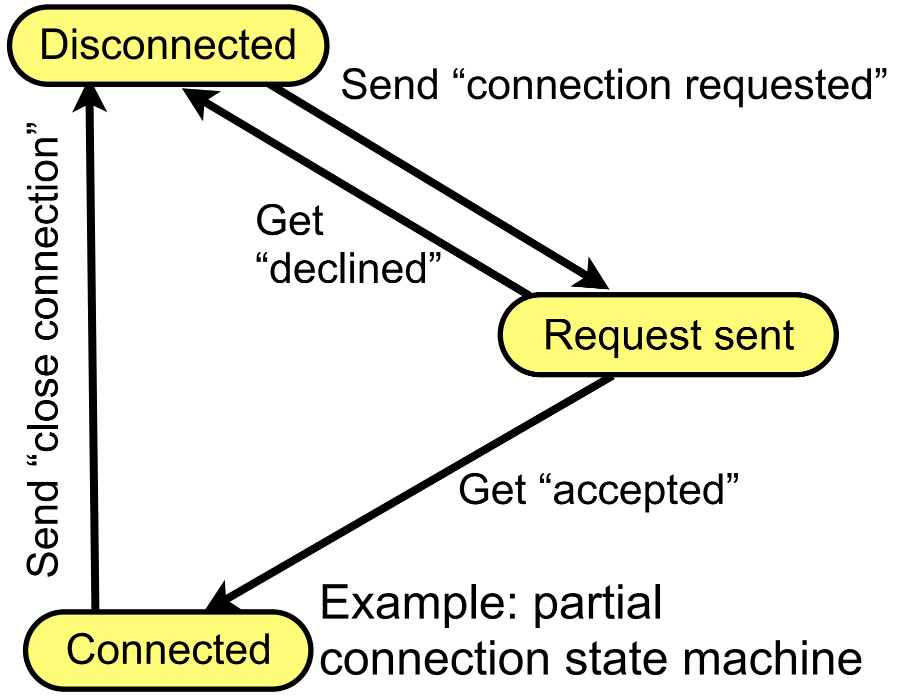
Network Protocols: Syntax

- A protocol will comprise different types of message
  - Known as *protocol data units* (PDUs)
- Each type of PDU will have a particular syntax
  - Describes what information is included in the PDU, and how it's formatted
  - PDUs may be formatted as textual information or as binary data
    - Textual PDUs have a syntax and grammar that describes their format
      - Much like a programming language has a grammar
      - Examples: HTTP/1.1, SMTP, SIP, Jabber
    - Binary PDUs similarly have rules describing their format
      - Is data big or little endian? 32 or 64 bit? Fixed or variable length? What are the alignment requirements?
      - Examples: TCP/IP, RTP
- PDUs define what messages are legal to send

Network Protocol: Semantics

- Protocol semantics define when PDUs can be sent, and what response is needed
  - Define who can send PDUs, and when they can be sent
  - Define roles for the hosts (e.g., client and server, peer-to-peer)
  - Define what are the entities that communicate and how they are named
  - Define how errors are handled

- Commonly described using state-transition diagram
  - States indicate stages of protocol operation
  - Transitions occur in response to PDUs, and may result in other PDUs being sent



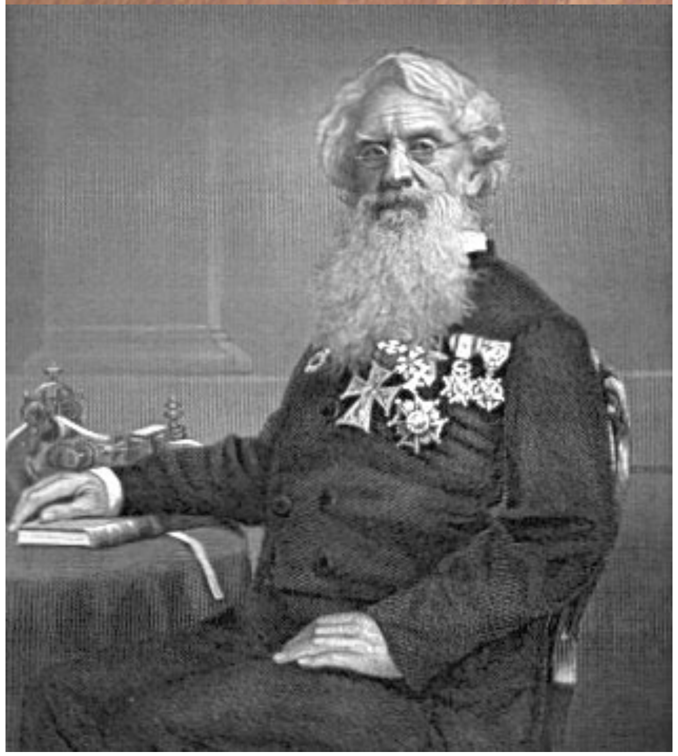
Example: partial connection state machine

Network Protocol Example: Morse Code

- A simple network protocol: Morse code and the telegraph
  - Signals on electrical cable form the channel
  - Syntax: pattern of dots and dashes signals letters

A	• -	J	• - - -	S	• • •
B	- • • • •	K	- • -	T	-
C	- • - •	L	• - • •	U	• • -
D	- • •	M	- -	V	• • • -
E	•	N	- •	W	• - -
F	• • - •	O	- - -	X	- • • -
G	- - •	P	• - - •	Y	- • - -
H	• • • •	Q	- - • -	Z	- - • •
I	• •	R	• - •		

  - Semantics:
    - Different gap lengths to signal end of word, end of latter
    - Use of STOP for end of message

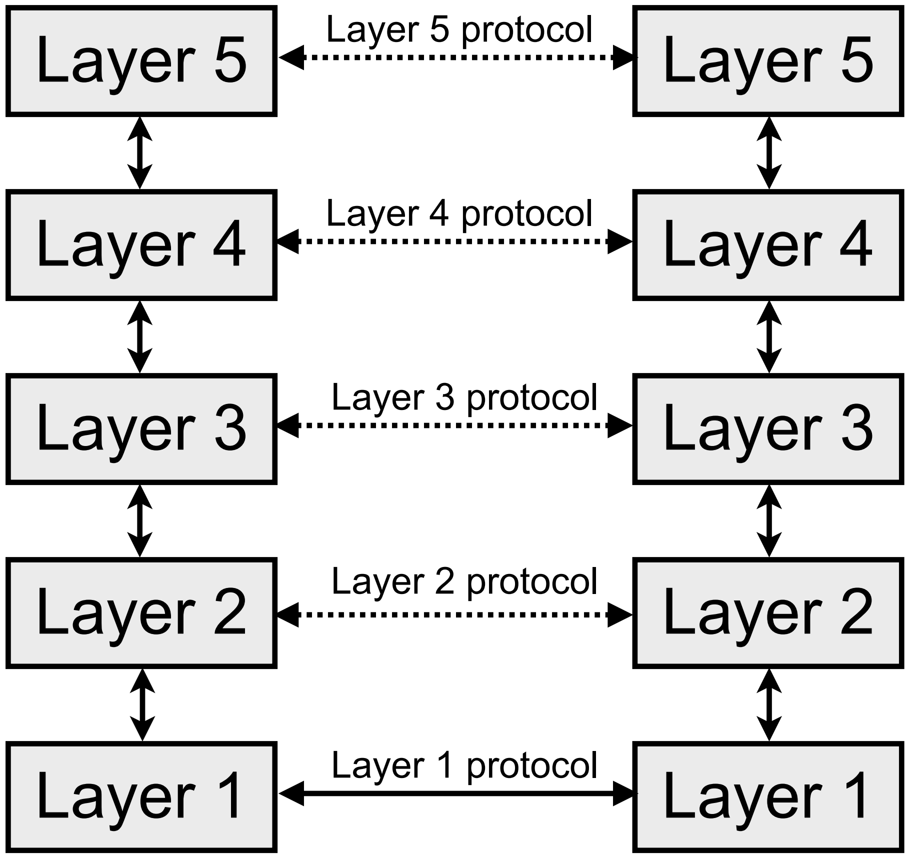


Samuel Morse

Photos: public domain (source Wikipedia)

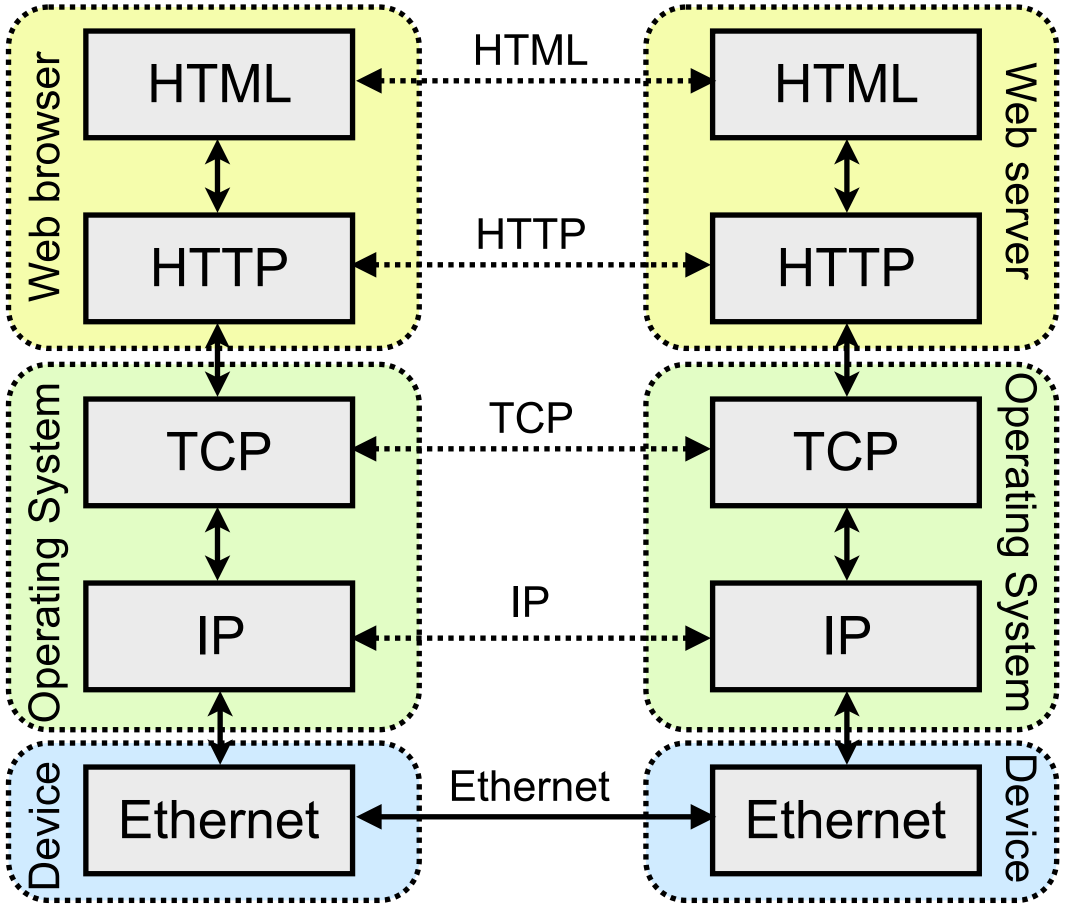
Protocol Layering

- Communications systems are typically organised as a series of *protocol layers*
  - Structured design to reduce complexity
  - Each layer offers *services* to the next higher layer, which it implements using the services of the lower layer – well defined *interfaces*
    - Highest layer is the communicating application
    - Lowest layer is the physical communications channel
  - Peers at some layer, *i*, communicate via a layer *i* protocol, using lower layer services



Protocol Layering: Example

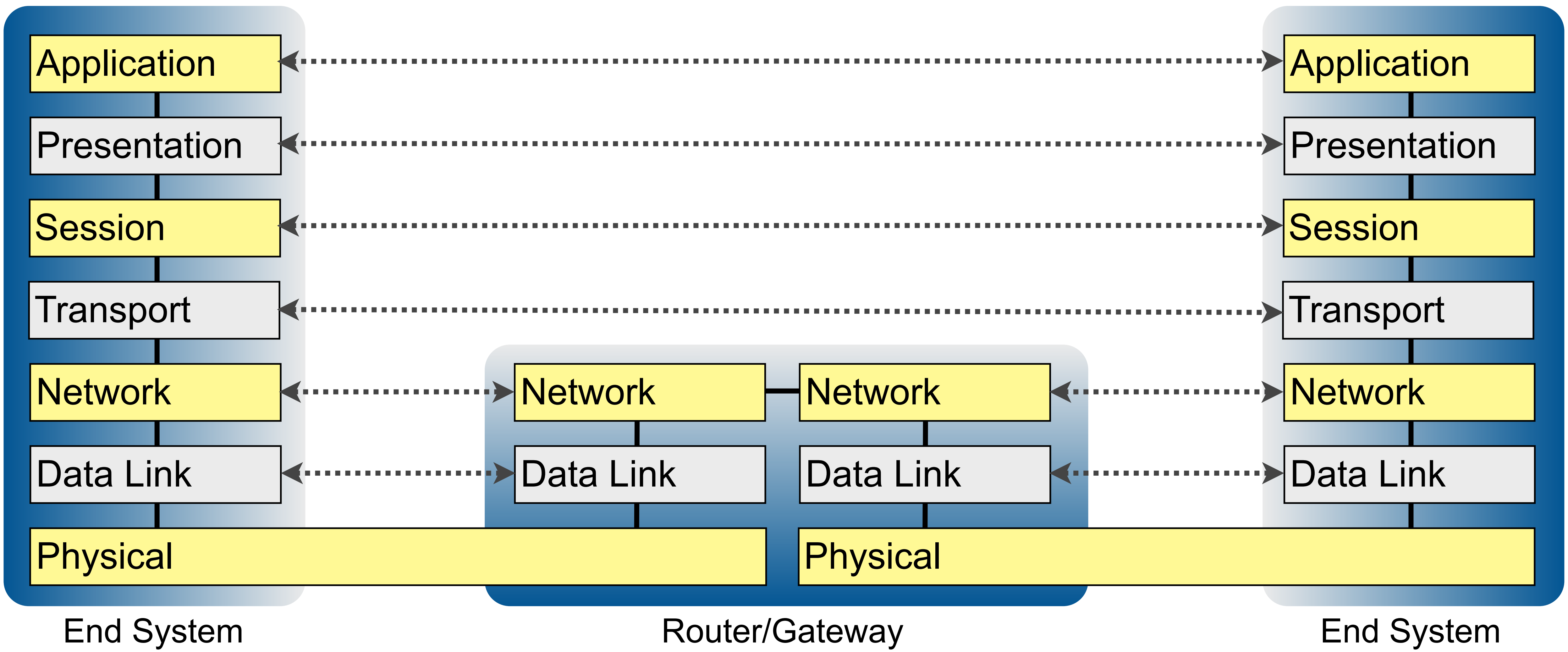
- Web browser talking to a web server
- Simplified view with five protocol layers:
    - HTML
    - HTTP
    - TCP
    - IP
    - Ethernet



OSI Reference Model

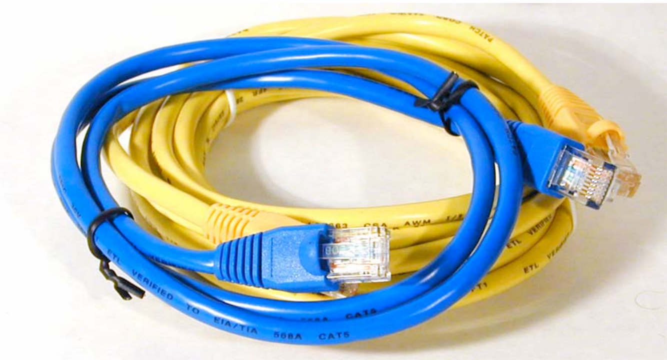
A standard way of thinking about layered protocol design

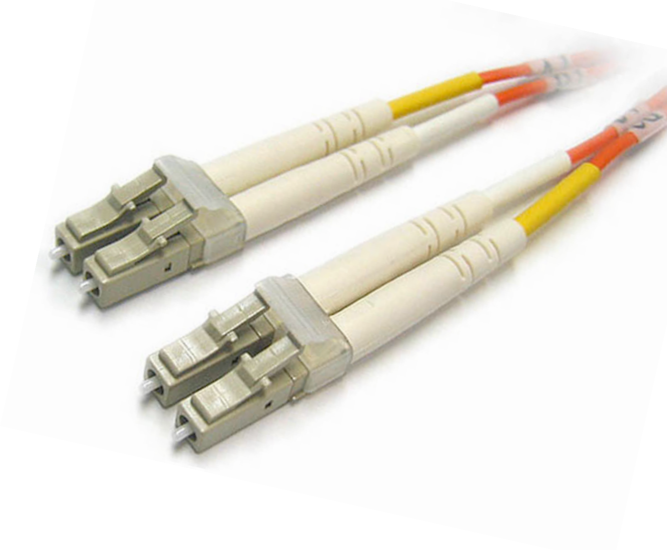
A design tool; real implementations are more complex



Physical Layer

- Defines characteristics of the cable or optical fibre used:
  - Size and shape of the plugs
  - Maximum cable/fibre length
  - Type of cable: electrical voltage, current, modulation
  - Type of fibre: single- or multi-mode, optical clarity, colour, power output, and modulation of the laser

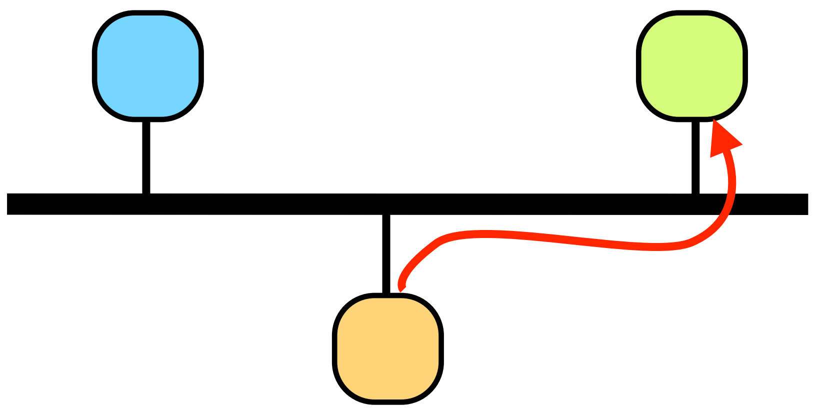




- For wireless links:
  - Radio frequency, transmission power, modulation scheme, type of antenna, etc.

Data Link Layer

- Structure and frame physical layer bit stream
  - Split the bit stream into messages
  - Detect/correct errors in messages
    - Parity and error correcting codes
    - (Negative) acknowledgements + retransmission
- Perform media access control
  - Assign addresses to hosts on the link
  - Arbitrate access to link, and determine when hosts are allowed to send message
  - Ensure fair access to the link and provide flow control to avoid overwhelming hosts
- Examples: Ethernet, 802.11



Network Layer

- Interconnects multiple links to form a wide area network from source host to destination host
  - Data delivery
  - Naming and addressing
  - Routing
  - Admission/Flow control
- Example: IP

Transport Layer

- End-to-end transfer of data from the source to the destination(s)
  - Transfers data between a session level service at the source, and corresponding service at the destination
  - May provide reliability, ordering, framing, congestion control, etc.
    - Depends on guarantees provided by the network layer
- Example: TCP

Session Layer

- Manages (multiple) transport layer connections
- Example session layer functions:
  - Open several TCP/IP connections to download a web page using HTTP
  - Use SMTP to transfer several email messages over a single TCP/IP connection
  - Coordinate control, audio and video flows making up a video conference

Presentation Layer

- Manages the presentation, representation, and conversion of data:
  - Character set, language, etc.
  - Data markup languages (e.g., XML, HTML, JSON)
  - Data format conversion (e.g., big or little endian)
  - Content negotiation (e.g., MIME, SDP)
- Common services used by many applications

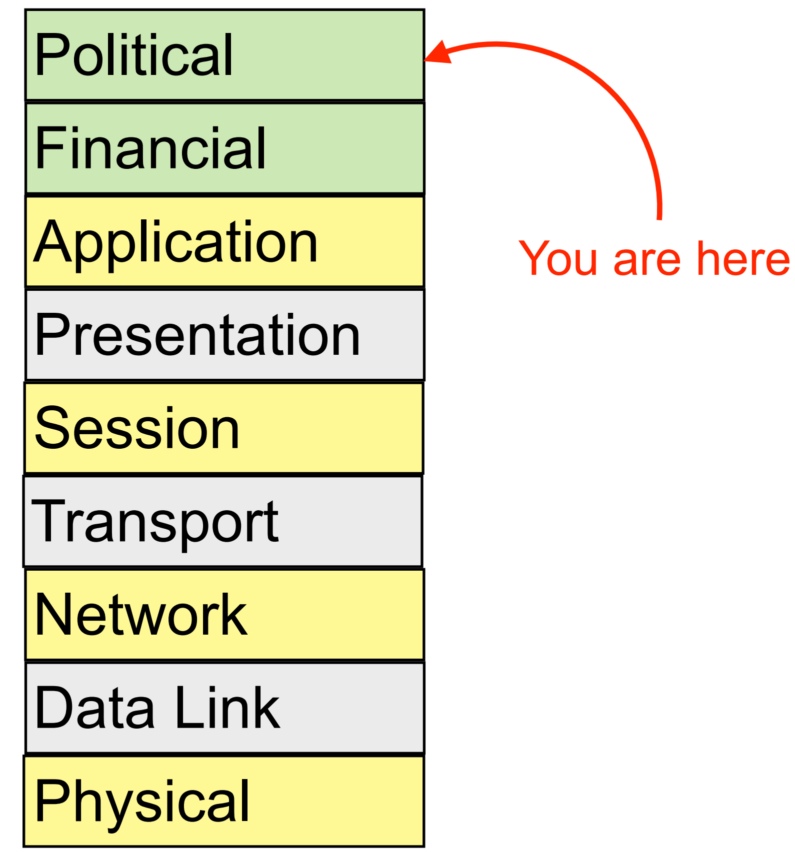
Application Layer

- User application protocols
  - *Not* the application programs themselves

- Examples:
  - REST APIs
  - WebRTC

Protocol Standards

- A formal description of a network protocol
- Ensure interoperability of diverse implementations
- Variety of standards setting procedures:
  - Open or closed standards development process
  - Free or restricted standards availability
  - Rules around disclosure of intellectual property rights, use of encumbered technologies
  - Individual *vs.* corporate *vs.* national membership
  - Lead technical development *vs.* describe existing practices
  - Collaborative *vs.* combative process



- Not all players in the standards process have the same goals
  - Delaying a standard to allow a proprietary solution to gain market share
  - Incorporating intellectual property, patented technologies, etc.
  - Enhancing, *or subverting*, the security of a protocol
  - …

Key Standards Organisations

- Internet Engineering Task Force
  - http://www.ietf.org/ and http://www.rfc-editor.org/
- International Telecommunications Union
  - http://www.itu.int/ (part of the United Nations)
- 3rd Generation Partnership Project
  - http://www.3gpp.org/
- World Wide Web Consortium
  - http://www.org/

Summary

- Communication → networking → networked systems
- Protocols: syntax and semantics
- Layered network architectures
- Importance of standards

"Networks are like onions…"