University of Glasgow

School of Computing Science

60 YEARS OF COMPUTING AT GLASGOW

Protocols and Layers

Networked Systems (H)

Lecture 2

Colin Perkins | https://csperkins.org/ | Copyright © 2017 | This work is licensed under the Creative Commons Attribution-NoDerivatives 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nd/4.0/ or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Lecture Outline

- Network protocols
- Protocol layering
- Protocol standards

Colin Perkins | https://csperkins.org/ | Copyright © 2017

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

CC BY ND

Network Protocols: Syntax

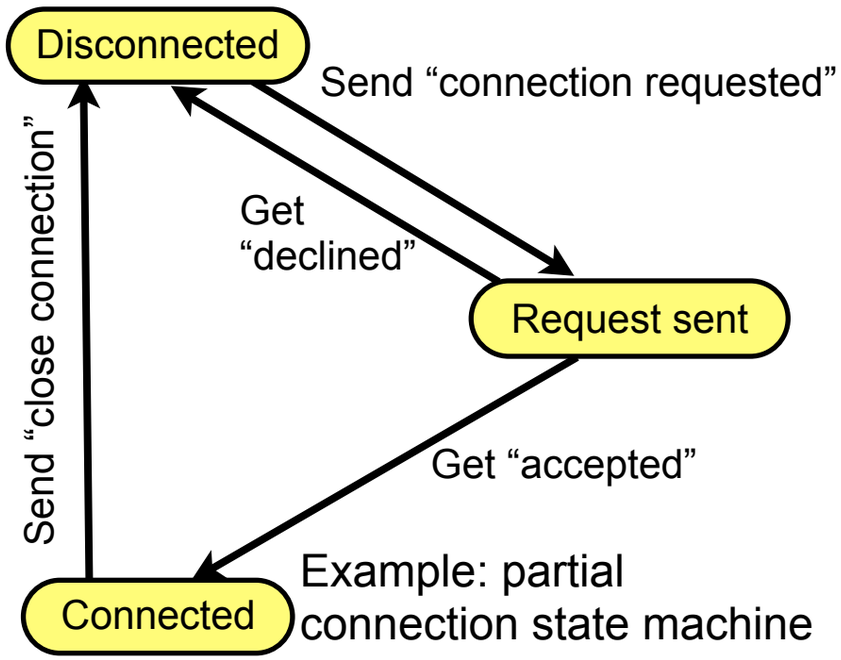
- A protocol will comprise different types of message
  - Known as *protocol data units* (PDUs)
- Each type of PDU will have a particular syntax
  - Describes what information is included in the PDU, and how it's formatted
  - PDUs may be formatted as textual information or as binary data
    - Textual 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

CC BY ND

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

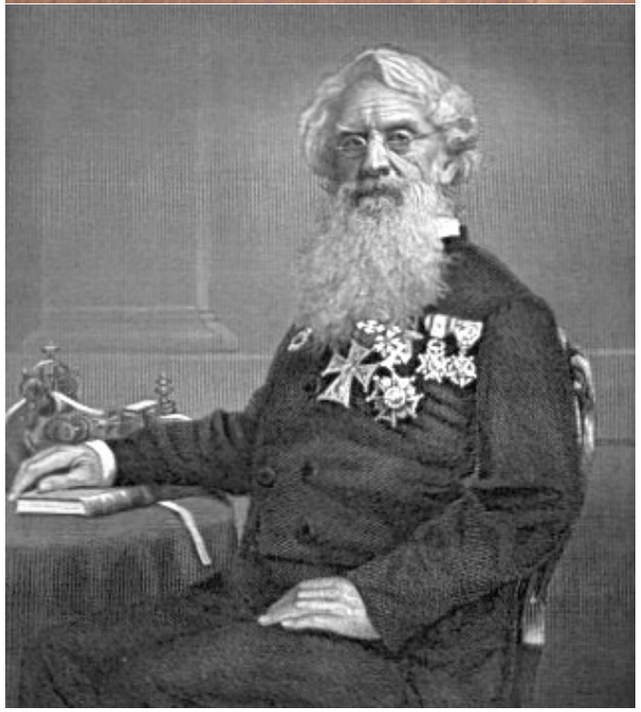
Colin Perkins | https://csperkins.org/ | Copyright © 2017

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	• - •		





Photos: public domain (source Wikipedia)

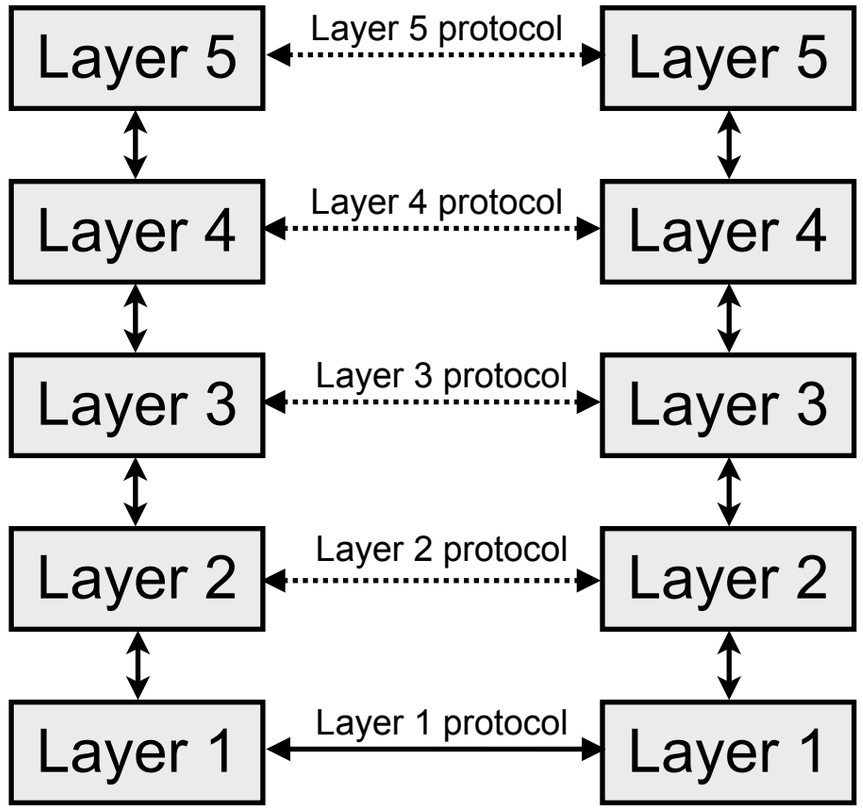
Samuel Morse

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

CC BY ND

Protocol Layering

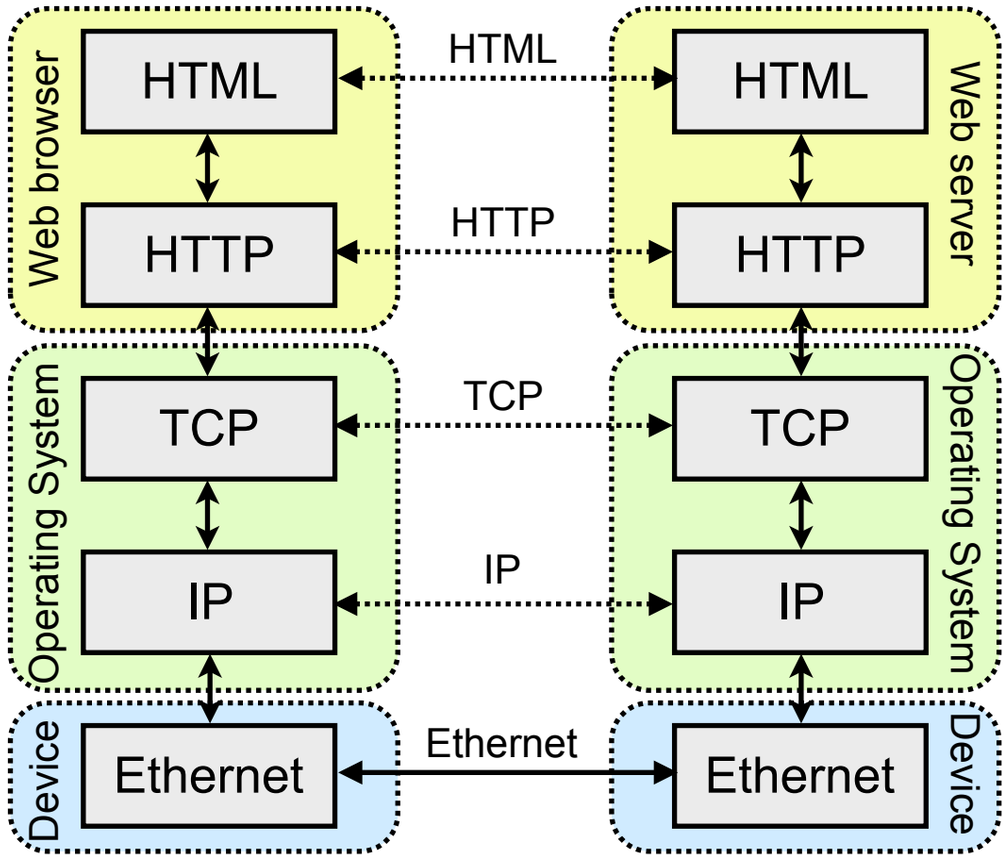
- Communications systems are typically organised as a series of *protocol layers*
  - Structured design to reduce complexity
  - Each layer offers *services* to the next higher layer, which it implements using the services of the lower layer – well defined *interfaces*
    - Highest layer is the communicating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



Colin Perkins | https://csperkins.org/ | Copyright © 2017

Protocol Layering: Example

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

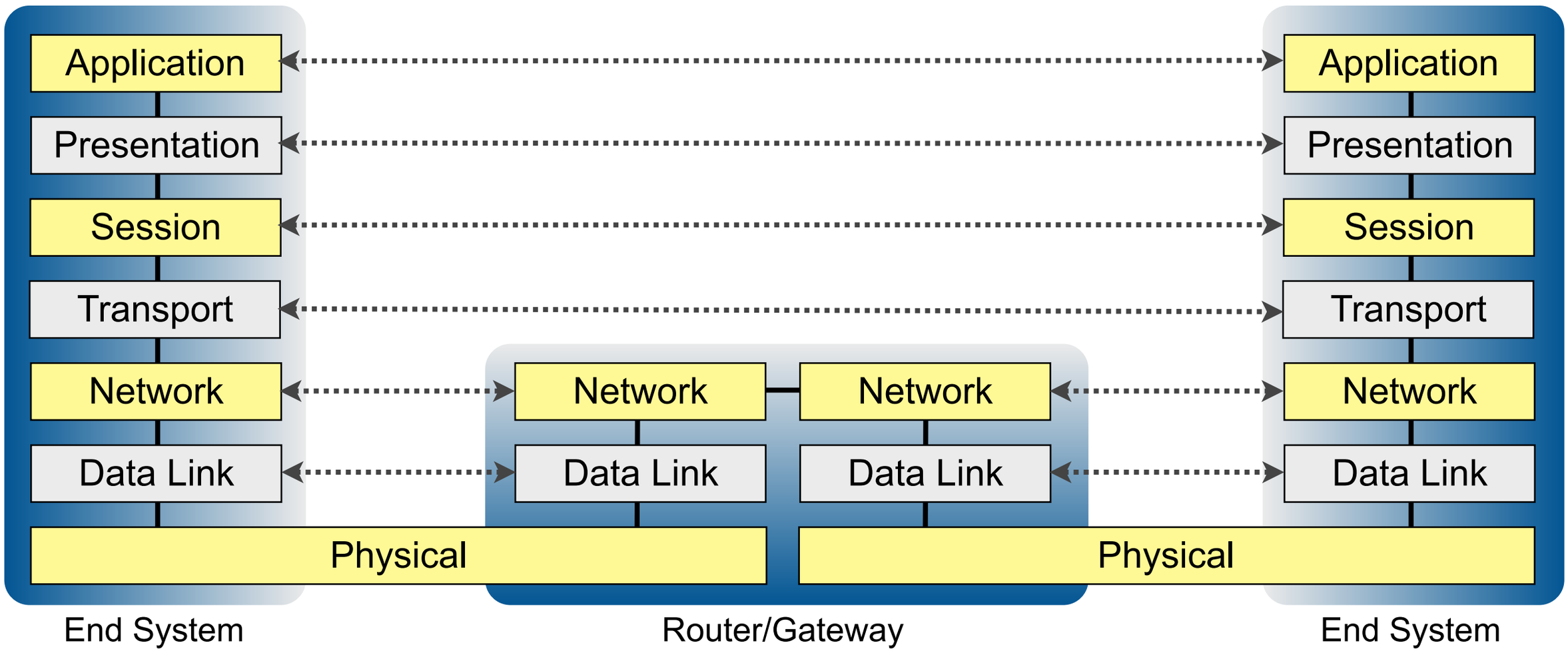


Colin Perkins | https://csperkins.org/ | Copyright © 2017

OSI Reference Model

A standard way of thinking about layered protocol design

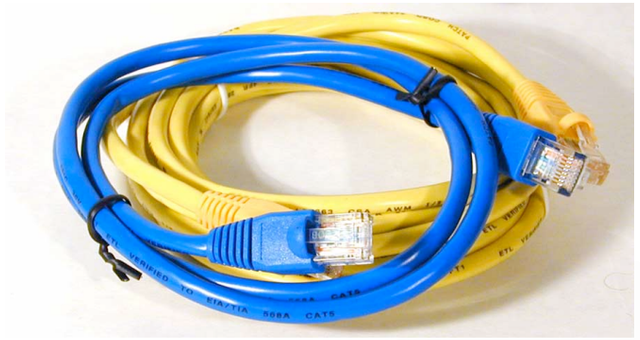
A design tool; real implementations are more complex

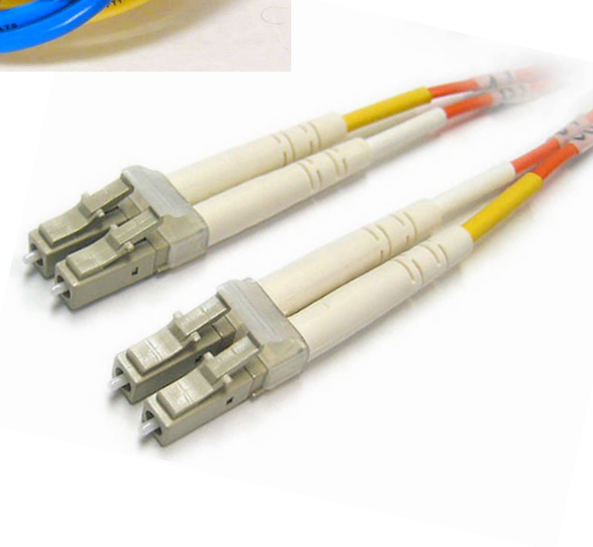


Colin Perkins | https://csperkins.org/ | Copyright © 2017

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



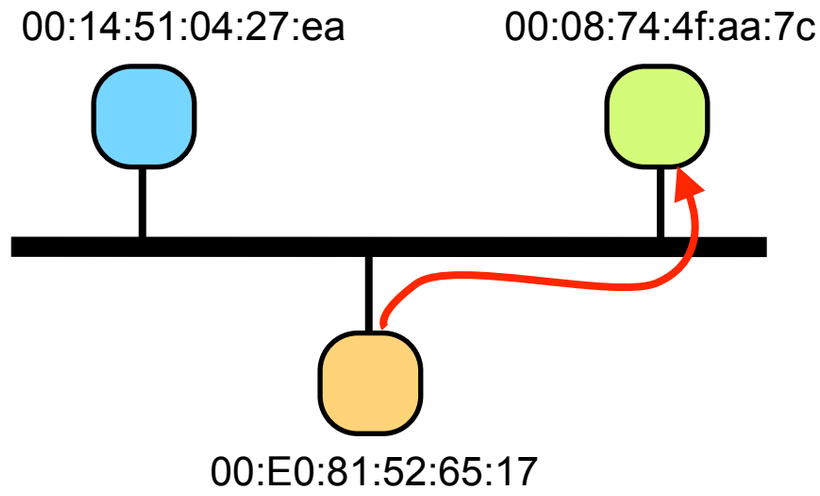


- For wireless links:
  - Radio frequency, transmission power, modulation scheme, type of antenna, etc.

Colin Perkins | https://csperkins.org/ | Copyright © 2017

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



CC BY ND

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

CC BY ND

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

BY ND

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

CC BY ND

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

Colin Perkins | https://csperkins.org/ | Copyright © 2017

Application Layer

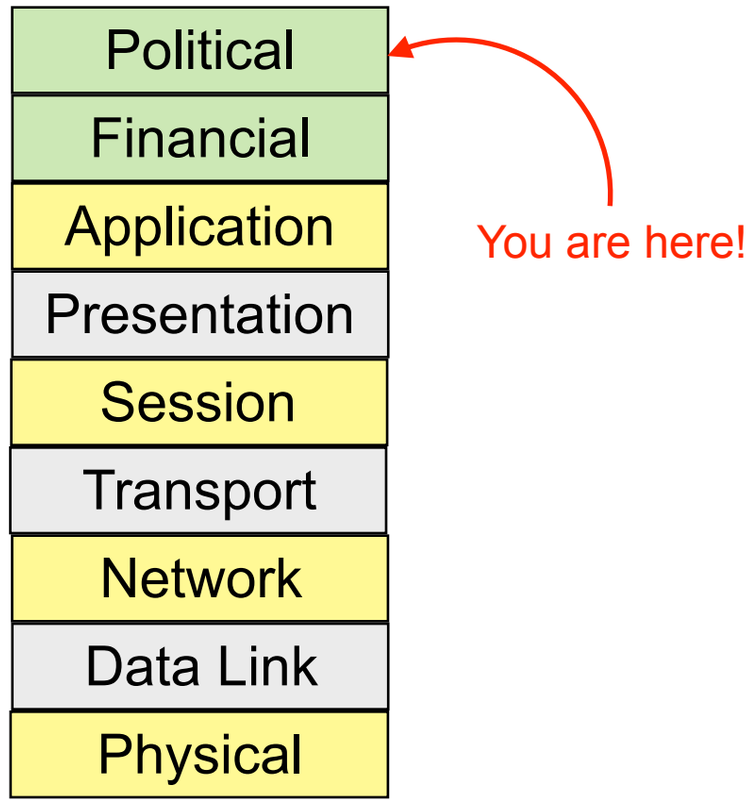
- User application protocols
  - *Not* the application programs themselves

- Examples:
  - Flickr API, Facebook API, Google Maps API, etc.
  - Web services
  - Grid computing

CC BY ND

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
  - …

BY ND

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/



Colin Perkins | https://csperkins.org/ | Copyright © 2017

Summary

- Protocols: syntax and semantics
- Layered network architectures
- Importance of standards

“Networks are like onions…”



Colin Perkins | https://csperkins.org/ | Copyright © 2017