CO212 - Networks and Communications

14th January 2020 — Week 1, Lecture 1

Evolution of the Internet

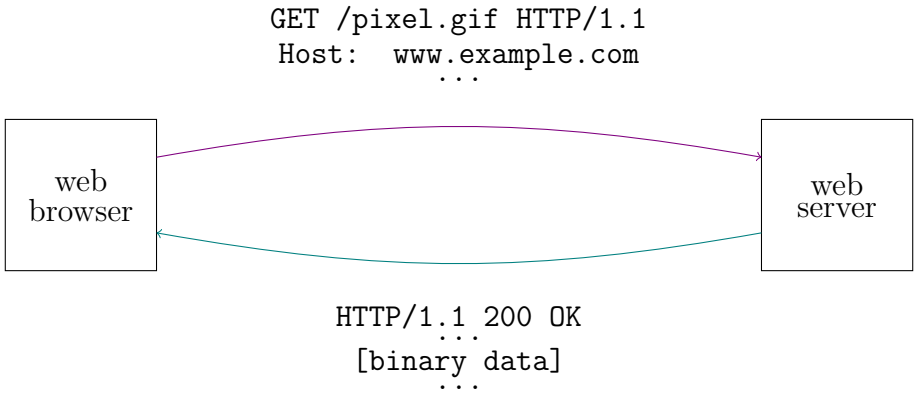
Literally only writing this part so I have something for the first lecture.

- (1969 - October) first message sent on ARPANET; "login", crashed after "l" and "o" were sent
- (1971) universities in West and East coast of USA connected
- (1980) London connected

14th January 2020 — Week 2, Lecture 1

World Wide Web (WWW)

This is an example of an **application** on the internet, based on HTTP (HyperText Transfer Protocol). A web browser (the client) sends a request to the web server over a pipe, which can be any form of connection between the two devices (can also be the same device), which in turn sends back a response.



Layers

- **application layer**

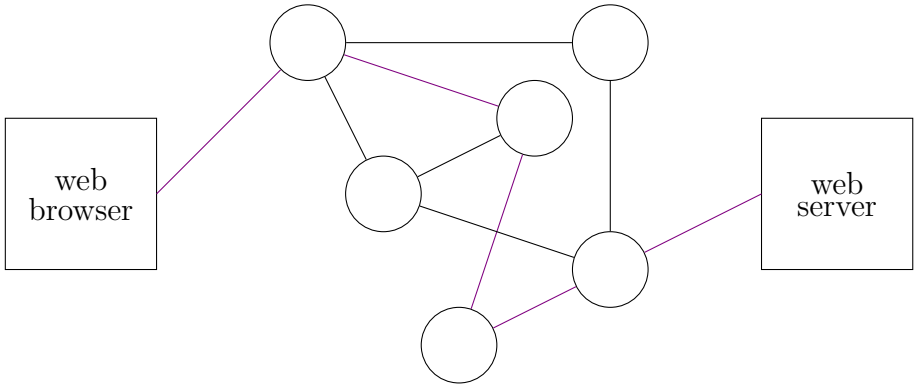
  Any software written for the internet is on the application layer.

- **transport layer**

  In the **transport layer**, packets leave your (client) machine to the server, and the server sends back packets to your client. This layer divides a (big) message into smaller chunks, and sends them to the other side (re-ordered) to be presented to the recipient.

- **network layer**

  The **route / path** (sequences of switches a packet goes through) each packet takes can be different from the others, and is often the most optimal route available. This is done on the **network layer**, which routers are a part of.



- **data link layer**

  Our devices are linked to the network on the **data link layer**, via network interface controllers (NICs). Examples of this include Ethernet, fiber optic network cards, as well as wireless devices such as WiFi access points, and USB dongles for 4G. A communication link is any connection between packet switches and / or end systems.

- **physical layer**

  Finally, on the **physical layer**, there are various forms of communication media, including fiber-optic cables, twisted-pair copper wire, coaxial cables, and wireless local-area links (802.11, Bluetooth, etc).

16th January 2020 Week 2, Lecture 2

Circuit Switching

Old phones used circuit switching, which creates a connection between the two points, which is used for the entire communication. This isn't used for the internet as the failure of one node in the circuit would lead the the entire communication dropping - whereas a different route would be calculated in packet switching.

Compared to packet switching, it has an expensive setup phase, but will need very little processing once the connection is established. However, it is inefficient for sharing resources - if a node is used as part of a circuit, it cannot be used by another connection for a different circuit. The resources are blocked once a connection is established (hence it is an inefficient way to use the network). On the other hand, packet switching has no setup cost, but has a processing cost, as well as space overhead, for every packet. It has a processing cost for forwarding the packets, as well as space overhead as there must be redundant data for each packet, such that it is self contained. It is specifically designed to share links, hence it allows for a better utilisation of network resources.

Protocols

A protocol is a set of rules (an agreement between communicating parties on how communication is to proceed), run by end systems as well as packet switches. It must be unambiguous, complete (includes actions and / or responses for all possible situations), and also define all necessary message formats. The phases are as follows;

- **handshake** establishes identities and / or context
- **conversation** free-form exchange
- **closing** terminating the conversation

The internet protocol stack is based on the 5 layers briefly covered in the previous lecture. Some examples of design issues that can be encountered are as follows;

- **addressing** how to denote the intended recipient
- **error control** how to detect (and possibly fix) transmission errors, e.g. checksums
- **flow control** ensure data travels through communication media without issues
- **multiplexing / demultiplexing** conversion of data into binary, and parallel communications
- **routing** which route is chosen

Most network layers are either connection-oriented, where a connection is first established, data is exchanged, and the connection is finally released, or connectionless, where data is marked with its destination.

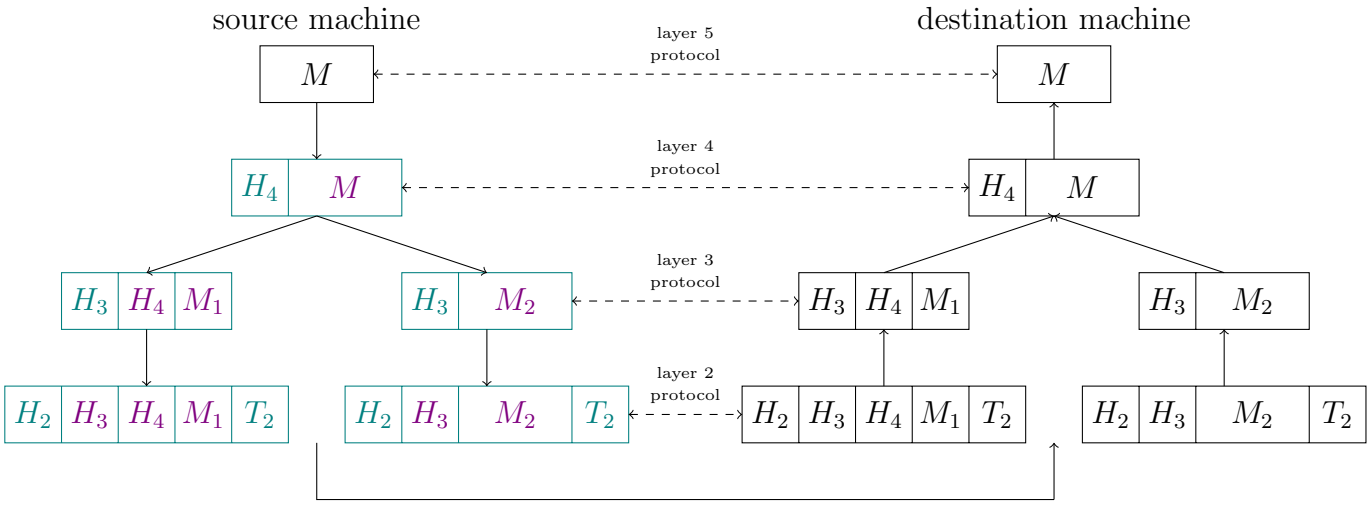
The TCP/IP (4 layer) stack consists of application, transport, internet, and network access (which combines data link and physical). On the other hand, the OSI (7 layer) model consists of the application layer, presentation, session, transport, network, data link, and physical.

application
presentation
session
transport
network
data link
physical

application
transport
network / internet
data link
physical

application
transport
internet
network access

A **service** is a set of primitives that a layer provides to the layer above it, whereas a **protocol** is a set of rules that prescribe the layout and meaning of packets. In a protocol stack, layer k puts its entire packet as data into a layer $k-1$ packet, the latter may add a header and / or a trailer. This may have to be split across several lower level packets (**fragmentation**). An example of protocol layering is as follows;



Note that the connection between the two machines can have multiple nodes in between, that can read up to a different physical layer. For example, if there was a link-layer switch after the source machine, it can read up to the link layer (layer 2), remove and add headers / trailers, and then send it on to the next device. The next device may be a router for example, which can read up to the network layer (layer 3), and do the same.

The data from the layer above is known as the SDU (service data unit), and the SDU combined with a header, added by the current layer, is known as the PDU (protocol data unit).