50005 - Networks and Communications - Lecture 2

Oliver Killane

20/01/22

Basic Concepts in Computer Networking

Lecture Recording

Lecture recording is available here

Networking Stack

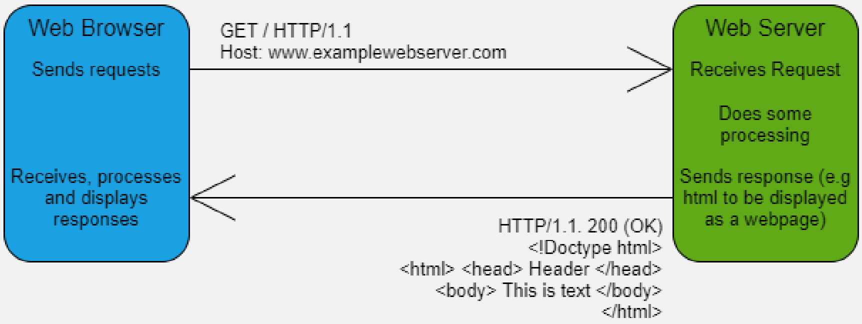
Definition: Application Layer

Applications send and recieve data in a format they specify. Implementations detais of **OS**, **packet types**, **network** setups and hardware models are abstracted away.

Applications use protocols which define structure of data (requests and responses), as well as port numbers and other conventions.

Example: World Wide Web

Part of the internet, invested by Tim Berners-Lee while working at CERN. It uses **HTTP**(HyperText Transfer Protocol). Early versions use plain text (newer & more advanced no longer always true).

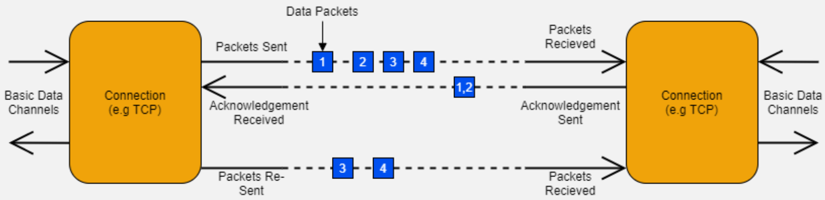


HTTP exists in the application layer of the **TCP/IP** stack. The level of abstraction on which we consider protocols, agreements and transfer of application data.

Definition: Transport Layer

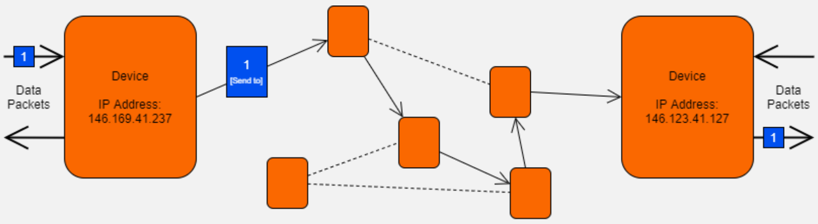
Establishes basic data channels, taking data to be sent or being recieved and converting to/from data packets. Networking can be:

- **Connection-oriented TCP** - Transmission Control Protocol. Packets not acknowledged to have been received are re-sent.
- **Connectionless UDP** - User Datagram Protocol. No checking, packets sent once, more performant.



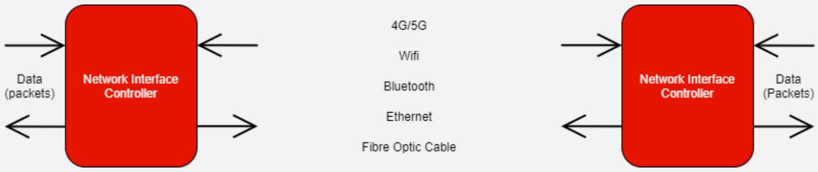
Definition: Network Layer

The internet protocol is used to add **IP Addresses** and other information to packets, and then route them through a mesh network of hosts to reach the destination. The path taken frequently changes and is per-packet.



Definition: Data Link Layer

NIC (network Interface Controller) hardware controlling communication over standards to allow physical communication of data (e.g ethernet, wifi, bluetooth, coaxial cable) to transfer data (packets) between devices.



Definition: Physical Layer

The actual hardware transfering data.

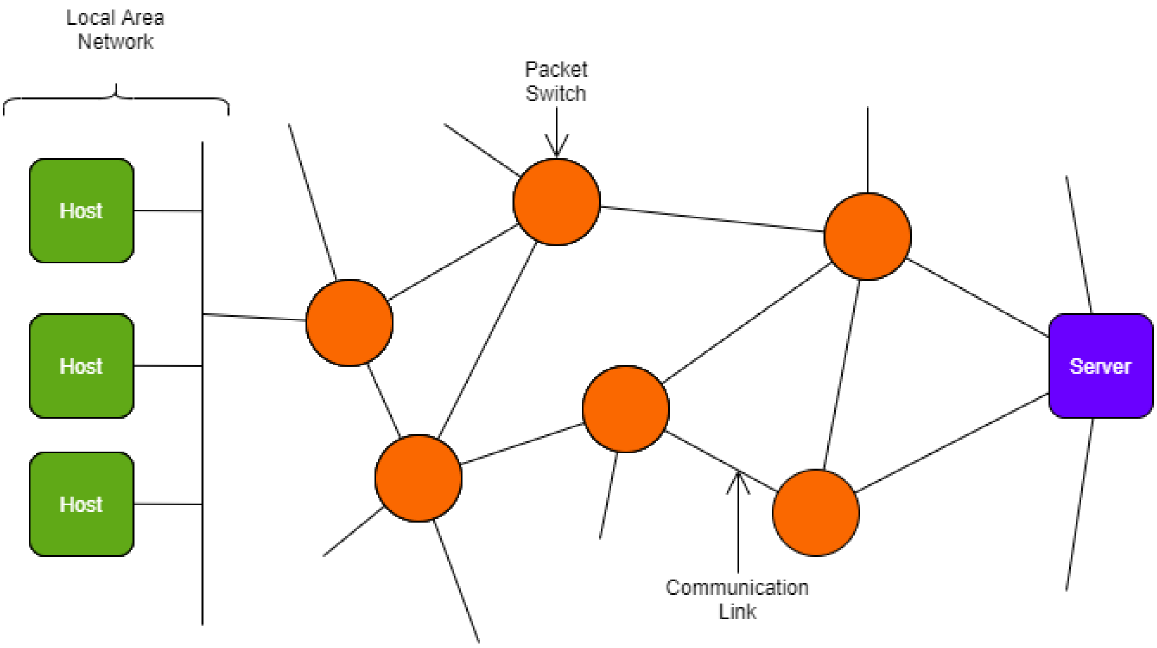
- Fibre-Optic cable
- Twisted-pair copper cable
- coaxial cable
- wireless links (wifi: 802.11, bluetooth)

Internet Structure

Definition: Host/End System

A computer system that is the source, or destination of communication.

- **Smartphone** Send and recieves data (e.g to browse internet)
- **Home Security System** To send and recieve security footage



Some simple terms (for now):

Packet Switch	A **Data Link Layer** switch or router (routes data through a network)
Communication Link	A connection between packet switches and/or end systems (hosts).
Route	A sequence of switches a packet traverses to go from source to destination.
Protocol	A standard concerning the control and format of sending and receiving data to and from end systems.

Packet Switching

Lecture Recording

Lecture recording is available here

Packet Switching

- Data is split into packets which are independently routed through the network.
- Switches & routers use packet information, and network status to determine which next router/end system to forward a packet on to.
- If any links in the network become slow or disconnected, packets are rerouted.
- No setup cost.
- Processing cost associated with forwarding each packet.
- Space cost associated with containing independent information in each and every packet.
- Quality of service is difficult to guarantee (no connection, processing and switching overhead, others can start using links as no reservation).
- High network resource utilisation (can send packets different routes in parallel, two connections can work on shared communication links)
- **Example** The internet.

Circuit Switching

- At the start of a connection, a path is specified and connected.
- Connection stays on the path for the entire duration of the communication.
- High setup cost to create path.
- No processing or space cost as data can be sent straight down the link.
- If the link becomes slow (over-saturated) or breaks a new link must be obtained (slow).
- Network Resources are reserved at connection start, so quality of service is guaranteed.
- Reservation of a route leads to inefficient network resource utilisation.
- **Example** Older telecommunicaton systems (landline).

Telephone Network

The old telephone network (modern is digital, Voice over IP) ius a circuit switched network. A circuit (path through the network) is connected and maintained for the call's duration.

Internet Protocol Stack

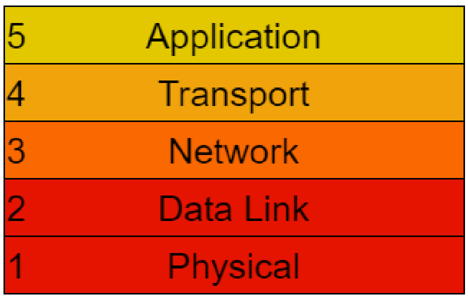
Definition: Communication Protocol

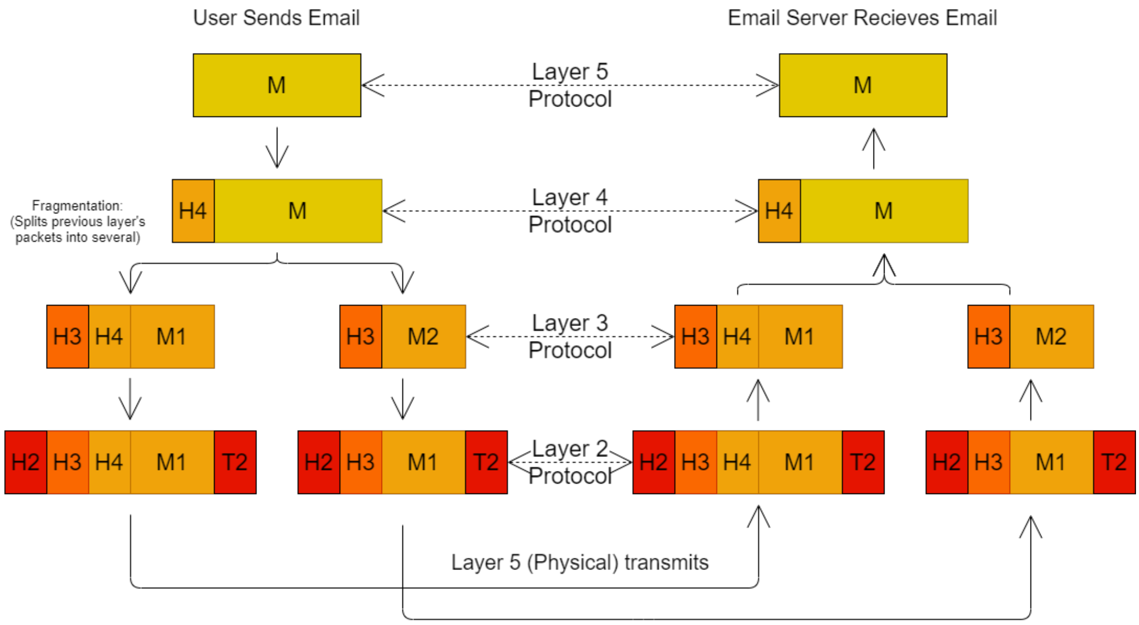
A network protocol is an established set of rules that determine how data is transmitted between different devices. (e.g describe layout and meaning of packets and the order they should be sent).

Phase	**Description**
Handshake	Establishes identities, and the context to begin the communication.
Conversation	Communication, exchanging data in the format & way specified by the protocol.
Closing	Terminates the conversation, performing any necessary cleanup/notification to other.

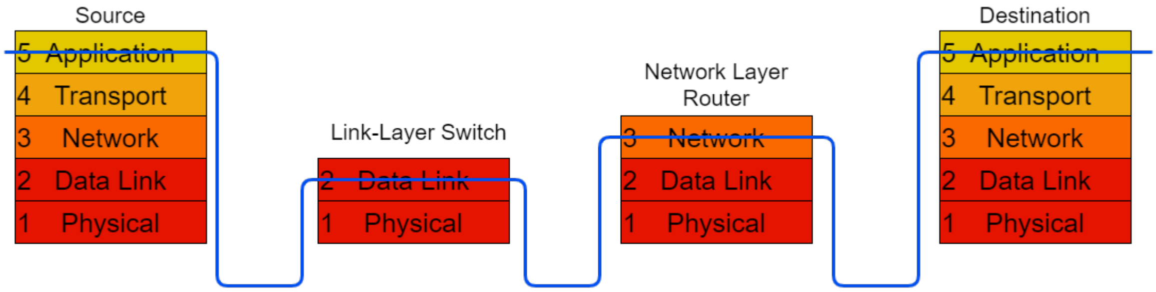
Hybrid 5-Layer Model

For this module we consider the 5 layer model of the networking stack.





For example when communication through other switches:



Alternative Models

In this course the 5 layer model is used. The reason being that Dr Gkoutzis claims it is the best. I (& you) have no reason to doubt this.

There is also:

7-Layer OSI

7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Data Link
1	Physical

4-Layer TCP/IP

4	Application
3	Transport
2	Internet
1	Network Access

Definition: Service

A set of primitives that the layer provides to the layer above. For example the **Transport Layer** packeting the data sent from the application layer.

Internet Protocol Design

When designing a protocol, one must consider:

- **Addressing** How to denote intended recipient.
- **Error Control** Detection and possible correction of inevitable transmission errors.
- **Flow Control** Prevent fast sender *swamping* slow receiver.
- **Demulti/Multi-plexing** Supporting parallel communications.
- **Routing** How to route packets to destination via best route with low processing/space overhead.

Most **network layers** have both connection-oriented and connection-less protocols:

Connection-oriented	Setup Connection with client, transmit data over channel. (e.g circtcuit switch, TCP on IP)
Connectionless	Send data to destination address, no formal connection created (*postal mode*). (e.g packet switching, UDP on IP)

Application Layer Protocols

- **Traditional** Name Services (DNS), Email (SMTP), FTP, Telnet, SSH, HTTP/S
- **Modern** Middleware to support distributed systems (Java RMI, Apache Thrift, Google Protocol Buffers - used for sending serialized data)
- **High Level** e-commerce, banking (visa) etc.
- **Peer-to-peer** BitTorrent, skype (old protocol)

Transport Layer

Offers both connection-oriented and connectionless protocols.

- Often provides network interface through sockets (e.g **UNIX** sockets).
- Provides support for secure connections.
- Support for **datagrams** (unreliable but fast per-message basis sending - connectionless, e.g **UDP**).
- Provides mechanisms to prevent fast senders overwhelming slow receivers.

Network Layer

Describes how routing and congestion is done.

- Determining best route.
- Dealing with router unreliability (e.g connection goes down).
- Supporting multicasting/broadcasting.
- Dealing with packet dropping (e.g when a router is overloaded).

Multicasting

Sending information to many recipients from a single source. Useful to reduce network traffic, for example a CCTV system sending a single video stream to be recieved by many screens, backups.

Data Link Layer

Reducing, detecting and rectifying bit transmission layers.

- Adding parity bits, checksum (e.g Cyclic Redundancy Check).
- Specifying how computers can share a common channel (**MAC** (Media Access Control) addresses).
- Specifying how network connects together (e.g Ethernet, FDDI (Fibre Distributed Data Interface)), and token rings (one holds token and listens at a time)

Physical Layer

Describe transmission of raw bits in terms of mechanical, electrical, optical means.

e.g set $0 : +4V, 1 : -3V$ change at frequency of $20KHz$.

Network Performance

Digital Units

$\times 1000$

Term	Bytes	
KiloByte	KB	1000
MegaByte	MB	1000^2
GigaByte	GB	1000^3
TeraByte	TB	1000^4
PetaByte	PB	1000^5
ExaByte	EB	1000^6
ZettaByte	ZB	1000^7
YottaByte	YB	1000^8

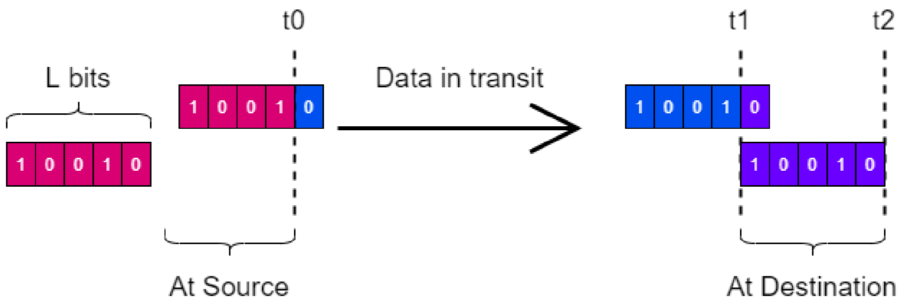
$\times 1000$

Term	Bits	
Kilobits	Kb	1000
Megabits	Mb	1000^2
Gigabits	Gb	1000^3
Terabits	Tb	1000^4
Petabits	Pb	1000^5
Exabits	Eb	1000^6
Zettabits	Zb	1000^7
Yottabits	Yb	1000^8

$\times 1024$

Term	Bytes	
KibiByte	KB	1024
MebiByte	MB	1024^2
GibiByte	GB	1024^3
TebiByte	TB	1024^4
PebiByte	PB	1024^5
ExbiByte	EB	1024^6
ZebibByte	ZB	1024^7
YobiByte	YB	1024^8

Speed and Capacity



Term	Description	Formula
Throughput	Total data received per time (link bandwidth).	$R = \frac{L}{t_2 - t_1}$
Latency	Time taken for a single bit to be transmitted (propagation delay).	$d = t_1 - t_0$
Packetization	Time per bit to be received (transmission delay).	$\frac{L}{R}$
Bandwidth	Maximum possible throughput.	
Transfer Time	Send time per bit	$\Delta = d + \frac{L}{R}$

Example: Small Email

We are transfering a file from London → Edinburgh.

$$L = 4KB \quad d = 500ms \quad R = 1MB/s$$

What is the transfer time (Δ)?

To get transmission delay. $\dfrac{4KB}{1MB/s} = \dfrac{4}{1000}s = 4ms$
Hence transfer time is: $500ms + 4ms = 504ms$.

Example: Big File

We are transfering a large file:

$$L = 700MB \quad d = 500ms \quad R = 1MB/s$$

Find the transfer time (Δ).

To get transmission delay: $\dfrac{700MB}{1MB/s} = 700s$
Hence transfer time is: $500ms + 700s = 700.5s$

Processing Delay

Processing delay d_{proc}:

- Check for bit errors.
- Determine output link.
- Negligible (¡msec).

Queueing delay d_{queue}:

- time waiting at output link for transmission.
- If link is congested, packet might be queued for a long time before being sent.
- If in queue too long, packet may be dropped.

$R:$ link bandwidth(bps) $\quad L:$ packet length (bits) $\quad a:$ average packet arrival rate $\quad \dfrac{L \times a}{R}:$ traffic intensity

$$\text{Delay } = \begin{cases} \text{small} & \dfrac{L \times a}{R} \approx 0 \\ \text{large} & \dfrac{L \times a}{R} \to 1 \\ \infty & \dfrac{L \times a}{R} > 1 \end{cases}$$

If more work is arriving that can be processed, the delay becomes infinite (and packets will likely be dropped).