COMP2221 Networks

David Head

University of Leeds

Lecture 2

Reminder of the last lecture

In the first, introductory lecture we covered:

- General admin.
- Brief history of networking, networks of networks (*i.e.* the internet), WWW *etc.*
- Some basic concepts in modern networks.
  - **End systems** or **hosts** at the ends of communication chains. Includes servers.
  - **Routers** and **access networks/edge routers**.

Today's lecture

Today we will see how networks can be understood using **layered models**:

- The early, general, 7-layer **OSI** model.
- Some of its pros and cons.

For most of this course we will focus on the 5-layer model that is more suited to the internet:

- Follows the Kurose and Ross book *(see Lecture 1)*.
- Sometimes called the **TCP/IP** model.
- Sometimes described as a 4-layer variant *(see later)*.

Layered network models

- Network communication is implemented through a network **stack**.
- **Protocols** are defined to pass messages between layers.
  - For this reason, sometimes referred to as the **protocol stack**.
- This **modular design** allows internal components to be easily replaced.
  - Restricts functionality at each level, so easier to understand.
- Potential for **redundancy** — overlapping operations carried out at more than one level.

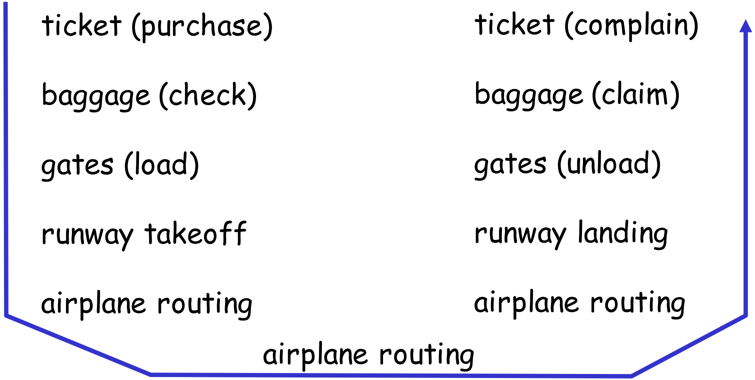
Principles

Ideally, a network layer model should:

- Create layers where a separate **abstraction** is required.
- Each layer has a well-defined **function**.
- Layer boundaries minimise the data flow across the boundary.

> Still leaves some choice about **how many layers** to create, and what **function** each performs.

Analogy: Air travel[1]



ticket (purchase)

baggage (check)

gates (load)

runway takeoff

airplane routing

airplane routing

airplane routing

runway landing

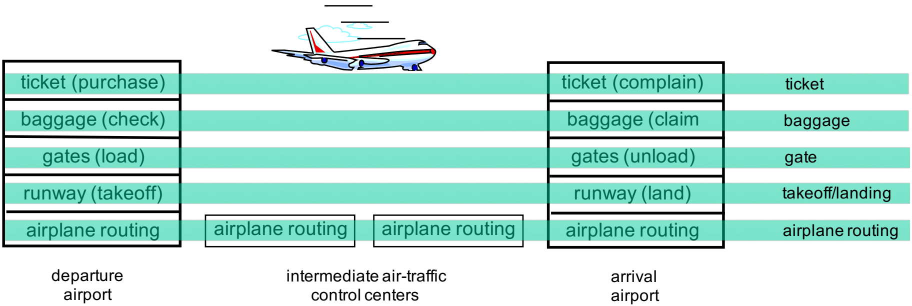
gates (unload)

baggage (claim)

ticket (complain)

[1] After Kurose and Ross, 7th ed., §1.5.

Layering of airline functionality



- Each layer implements a **service** . . .
- . . . relying on services provided by the layer below.

OO design

These concepts are identical to **object oriented** programming:

- The implementation within each layer is **private** *(encapsulated)*.
- Communication between layers is the public **interface**[1].

Modularisation eases maintenance and updating of the system.

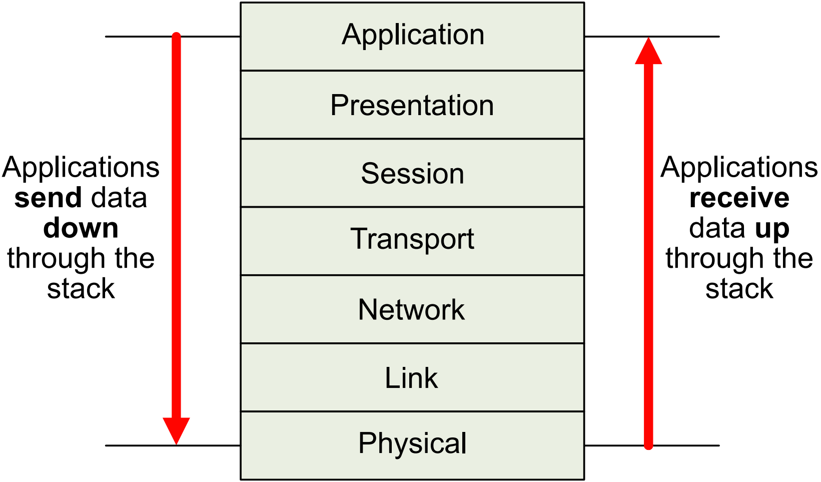
- Changes in implementation of one layer's service is transparent to the rest of the system.
- *e.g.* change in gate procedure does not affect the rest of the system.

[1] Interestingly, Java `interfaces` are actually <u>called</u> `protocols` in Objective-C.

The OSI Reference Model

- **OSI** (Open Systems Interconnection) reference model.
- Developed in the 1970's by the International Organization for Standardization (ISO).
  - At a time when networking protocols were still being defined.
  - Probably not intended for the internet.
- Applies to networks in general.
- 7 layer stack.
- Sometimes used in text books and training courses.

Layers in the OSI model


Application
Presentation
Session
Transport
Network
Link
Physical
Applications **send** data **down** through the stack
Applications **receive** data **up** through the stack

Application, Presentation and Session layers

The **application** layer contains the user-facing protocols.

- `ftp`, `http`, `smtp`, . . .

The **presentation** layer is used to interpret data.

- Includes compression and encryption protocols.
- Also data description (*i.e.* format).

The **session** layer organises and structures the dialogue between applications.

- Delimiting and synchronisation.

Transport layer

(Downwards) Accepts session data and splits into **segments**[1] before passing to the *Network* layer.

(Upwards) Receives *Network* layer segments and constructs *Session* layer data.

Data integrity depends on the protocol.

Host-to-host communication

- Includes destination address in a header.
- Oblivious to network infrastructure.

[1] The terms *packets*, *segments*, *datagrams* and *frames*, are used differently by different authors, and not always consistently.

Network layer

Controls the operation of the sub-network between hosts.

Determines how packets (often called **datagrams** for this layer) are routed **dynamically**.

- Links may be made or broken as machines are added to the network, break down *etc.*

May also offer **congestion control** and **quality of service**.

The highest level of the protocol stack for simple routers.

- *i.e.* routers do not need *Transport* or *Application* layers, but will often have software to help routing, firewalls *etc.*
- Will mention this again in Lecture 17.

Link and Physical layers

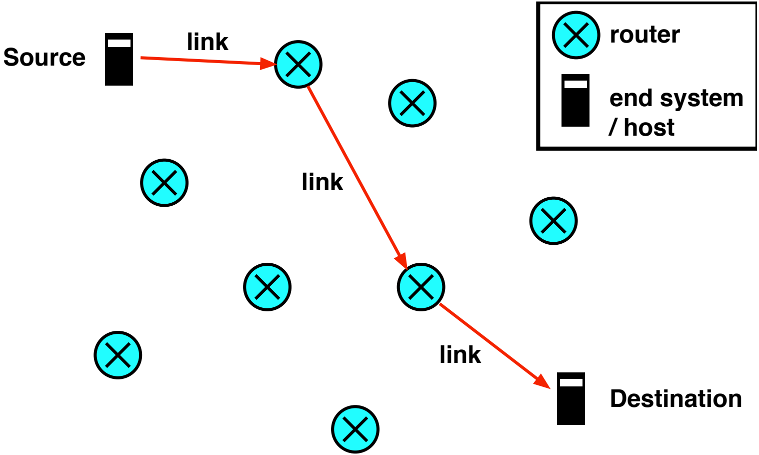
The *Link* layer, sometimes known as the *Data Link* layer, handles the movement of **frames** *(packets)* from node-to-node along the route.

- Ethernet, Wi-Fi, and combinations.
- This is the layer at which **MAC** (Media Access Control) addresses are relevant.

The *Physical* layer handles movement at the bit level.

- Copper wire, fibre optic, radio waves *etc.*
- Purely 'hardware.'

Network-layer routing



Each link in the **‘chain’** handled by the **(Data-) Link layer**.

Pros and cons of the OSI model

Pros:

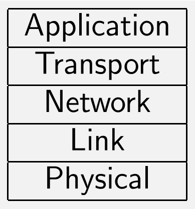
- Clearly defined layers and protocols.
- Very general design that could be applied to *any* network (defence, cash machines *etc.*), not just the internet.

Cons

- Often overly complex.
- The *Presentation* and *Session* layers have minimal functionality that could easily be subsumed into the **Application** layer.

The TCP/IP 5-layer model

The most widely used mode for the internet, and the most relevant for this course, is a simplified version of the OSI model:



Application

Transport

Network

Link

Physical

- *Presentation* and *Session* layers merged into *Application*.
- *Network* layer sometimes called the *Internet* layer.
- (*Data-*)*Link* and *Physical* layers sometimes merged to give a 4-layer model (or the *Physical* layer simply dropped).

Link and Physical layers

The *Physical* layer is where data is converted to/from *e.g.* electrical or radio signals.

- May involve digital-to-analogue conversion.
- Sometimes merged with the *Link* layer.

The *Link* layer controls how packets are transported between network nodes.

- How *e.g.* your device or laptop is detected by a Wi-Fi network.
- Error checking, **which may be redundant** (*i.e.* repeated at another level).

We will briefly cover these layers in Lecture 19.

Network layer

Most common protocol is the **Internet Protocol**, or IP.

- Describes how data is grouped into **datagrams** *(packets)*.
- The network **addressing scheme**.

A **datagram** is produced containing the data and header information.

Also ICMP (Internet Control Message Protocol), which is how routers communicate to ensure efficient transport of messages through the network.

Java only understands IP.

Transport layer

Raw data packets can be corrupted, arrive out-of-order, or not arrive at all!

Essentially two protocols exist at this layer:

TCP	Transmission Control Protocol
	Ensures all packets/segments are received. May ask for re-transmission. Can have a high overhead.
UDP	User Datagram Protocol
	Detects corruption, but not ordering or lost packets/segments. Low overhead.

Application layer

User code and **interface** reside here.

Sends data into the *Transport* layer, and delivers data to the user from the *Transport* layer.

The protocol used determines what is done with incoming data:

- `http` - display in a browser.
- `ftp` - deliver a data file.
- `smtp` - for sending email.
- *etc.*

Can also create user-defined protocols.

Pros and cons of the TCP/IP model

Pros

- Suited to network *programming* (particularly `java.net`, which is designed around TCP/IP).
- Simpler than the OSI model.
- Well suited to the internet.

Cons

- Layers and protocols are not always clearly defined.
- For example, some security protocols sit ‘between’ layers – see Lecture 13.
- Lacks generality.

Course structure

Lectures	Content
1	Introduction and admin
2-4	Network layers and internet architecture
5-15	Application layer
16	Transport layer
17-18	Network layer
19	Link and Physical layers
20	Course review

Summary and next lecture

Today we have seen how networks can be modelled in layers:

- The general, 7-layer OSI model.
- The 5-layer TCP/IP model most suited to the internet.

In the next few lectures we will look at the basic architecture of the internet:

- Ports and IP addresses.
- More on TCP and UDP.

This is all we need to know to start the programming from Lecture 5 onwards.