1 Introduction

Networks and networking have been fundamental to human existence, since before we had computer technology to make it faster and easier. As soon as you have more than one person or source of knowledge, there's value in the two communicating together. This course is a "technology lead course" that gives you the basics of computer networking – providing both the theoretical understanding and how computer networks work in the real world.

This course assumes some familiarity with computers and basic computer concepts, but assumes no pre-knowledge of networking.

Coming out of this course, you will

- know how the core protocols work in a modern IP network
- understand the basic concepts behind these and why they are as they are, to enable you to make your own models of how things should work when faced with unfamiliar protocols or behaviour
- have concrete examples of how modern IP networks are built, and be able to build and debug your own
- have a set of resources and pointers to bootstrap further learning in this space.

1.1 Structure of the course

1.1.1 Contents

The information in this course is grouped in two ways.

- The majority of the course works looks at core networking concepts and then works through the layers of networking as we scale up from signals flowing on a wire to internet scale routing, including unpacking and understanding some of the core protocols.
- We then look at some broader aspects of understanding that are important to networking, such as virtualisation, security and trust.

1.1.2 Format

The format of the course is a "standard" Oxford Computer Science lecture course, consisting of the following.

- Lectures. For this course, the printed notes provides the core skeleton of the information provided in each lecture. The lectures then expand on this information explaining why and how, and providing examples. It is expected that you will take your own notes during the lectures to cement your understanding. It is encouraged to ask questions during the lectures.
- Exercise sheets. For this course, these questions are intended to explore application of the information from the lectures, to promote understanding and use of the information, rather than rote learning. Many of the questions in these sheets have no strict single correct answer.
- Lab Practicals. For this course, these include building software that implements some examples of networking elements from the course.

1.1.3 Covid-19

For Trinity Term 2020, the social distancing rules mean that changes have had to be made to the above lecture format which relies on synchronous information transfer between lecturer and students. To enable asynchronous transfer of knowledge, we have made the following changes.

- The skeleton lecture notes are significantly expanded and are made available before each formal lecture slot. These are intended to provide enough detail for students to use these to learn and understand the course without attending lectures.
- The lecturer is available for live discussion and questions during the formal lecture slots.
  - Week 1-6: Tuesday 10-12am (All times British Summer Time)
  - Week 1-4: Friday 10-11am.
- The lecturer is available to respond to emails sent from course students outside these times, with average turnaround 1 working day.
  - Email address: edmund.pringle@metaswitch.com

1.2 Resources

1.2.1 Wireshark

Wireshark (www.wireshark.org) is an *incredibly* useful open source tool for looking at information flowing in a network. It can both record copies of data flowing in and out of an interface, and provides both the raw data, and human-readable decoded explanation of what the values in the data means for almost all protocols imaginable. In networking, Wireshark is your friend. Download it, install it, and play with it. See the wiki at https://wiki.wireshark.org/FrontPage for how to use this.

Alongside several of the lectures in this course, I'll include real packet captures that show some of the protocol messages that the lecture has covered running in a live network. These are in a format that can be loaded into Wireshark. You can find some of these, and many more in the Wiki above.

2 Networking Basics and Background Knowledge

In networking, there are a number of basic building blocks that we'll refer to as we go through the course. These are introduced here for reference.

2.1 Models and Layering

Throughout all of science, all of reality, we use models to enable us to break down complicated things into a series of simpler things and thus understand them (to enable us to focus, think about and understand one area at a time).

Both the standard models that we use to simplify our understanding of networking are layered – with each layer responsible for providing a clear abstracted service interface to the layers above.

2.1.1 The OSI (Open Systems Interconnection Model

The OSI model consists of 7 layers. It is the more abstract of the two models that we'll cover here. Each layer has a specific set of responsibilities, and accepts data from the layer above, and provides it to the layer below.

1. The Physical layer. At the "bottom", the physical layer is concerned with getting individual bits of information from A to B. It is the only layer that involves transport over distance, and is responsible for taking an input stream of bits of data in one place, and outputting that stream of bits (or something close to it) at another place. Examples include an Optical cable

and the lasers/detectors used to transmit and receive/interpret the light in the cable, or an Ethernet cable and the ports at each end that push or detect current flow in it.

2. Data Link Layer. The data link layer is concerned with getting coherent clumps of bits between two points. It takes clumps of data from Layer 3, and shoves streams of bits to Layer 1. It also provides some basic error detection/correction to cope with Layer 1 failures. An example is Ethernet.
3. Network Layer. The network layer is concerned with getting coherent clumps of bits between hosts. It differs from the Data Link Layer, in that the Data Link Layer does this at a local (small-scale) level whereas the Network Layer does this at a global level. Layer 3 provides some error detection, but no error correction – your clumps of data may not arrive or may arrive corrupted. Layer 3 is de-facto Internet Protocol (IP).
4. Transport Layer. The transport layer provides a service link between two hosts, that transports data while providing additional service function. For example, Transmission Control Protocol (TCP) provides a guarantee of ordered delivery of data and proof of delivery to the sender.
5. Session Layer. The session layer provides logical connections between applications running on different host, establishing and maintaining the connections.
6. Presentation Layer. The presentation layer provides a contextual link between applications, under a defined syntax. For example the ASN.1 interface and schema.
7. Application Layer. The application layer provides access to network resources. For example an HTTP client.

   In all honesty, I've always found layers 6 and 7 are a bit of a fudge, and suspect that they originate in the originators wanting there to be 7 layers.

2.1.2 The TCP/IP model

The TCP/IP model consists of 4 layers, and is more tightly bound to the protocols used at each layer.

1. The Network Access Layer is responsible for transmitting data between two adjacent devices – equivalent to layers 1 and 2 of the OSI model.
2. The Internet Layer is responsible for moving data packets between a source and a destination – equivalent to layer 3 of the OSI model.
3. The Transport Layer is responsible for providing reliable connections between applications running on hosts – roughly matching layer 4 of the OSI model
4. The Application layer provides access to network resources – equivalent to layers 5+ of the OSI model.

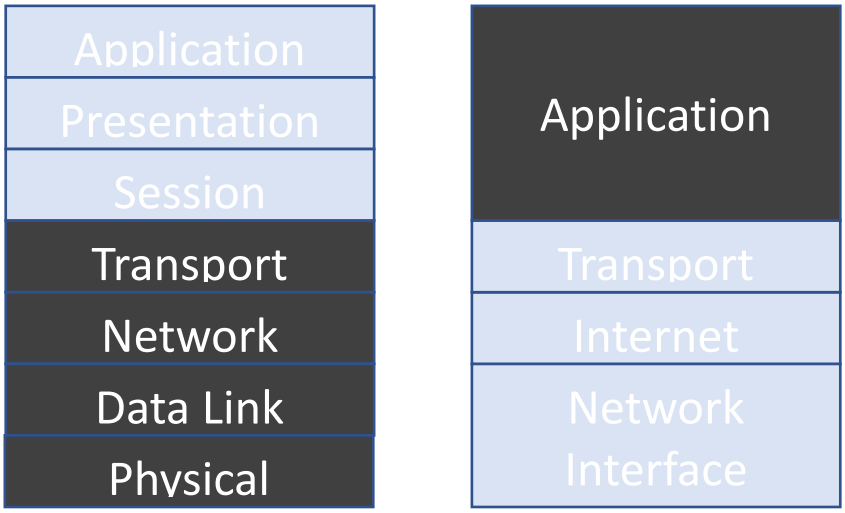
2.1.3 Models vs. Reality and their use in this course.

The layers in the OSI (Open Systems Interconnection) model and TCP/IP model match up as shown in the diagram below.

All models are leaky abstractions,that don't *exactly* match reality – and the same is true of these models. The layers are not *perfectly* separated, which is fine – when it's expedient to break the model, people break it and we'll see examples over the course. For us in this course, the lower levels of the OSI model provide some useful distinctions, but the upper levels are a bit of a fudge. In this course, we'll use the OSI model up to the Transport layer and above that use the TCP/IP Application layer.

Both layered models are also called stacks, and networking software is often referred to as a networking stack (with the software working at the different layers referred to using names like The

Layer 2 Stack).Figure 1: Networking models side by side – the OSI model on the left and the TCP/IP model on the right



2.2 Protocols

Protocols are agreements between peers about how to implement an exchange of messages. Think of them as the types of conversations.

For example, the human conversation protocol typically starts by establishing a connection, by one person saying "Hello." "Hello." moves to information flow, for example "Did you *see* the news last night!?" "I know, what were they thinking?", handles transmission failures "Sorry – I didn't catch that, could you repeat?" and has a clean termination "Bye." "See you tomorrow."

Networking protocols operate (primarily – there are exceptions as we'll see later in the course) within one layer of the OSI networking stack, with each layer having its own suite of different protocols. Some further examples:

The **application layer** protocol HTTP consists of agreements about exchanges of formal texts.

At the **transport layer**, the TCP protocol describes opening and closing byte-streams and how to move bytes from one end of that pipe to the other. TCP connects sockets with a bidirectional pair of reliable byte streams.

At the **network layer**, IP delivers packets from one host to another. IP delivers individually addressed packets to a destination address offering "best effort" delivery.

At the **data link layer**, protocols such as Ethernet provide packet delivery, and in turn these rely on physical protocols (e.g. agreements on what voltages are used on copper wire.)

2.3 Standards and standards bodies

Distributed, heterogeneous, collaborative enterprises like the internet rely on standards to allow interworking of different implementations. There are two key bodies that you will come across regularly in networking.

2.3.1 Internet Engineering Task Force (IETF) (https://www.ietf.org/)

The IETF identifies problems that we would like to solve in networking and invents ways to solve them. It is the source of a lot of the protocols that we find.

It's an egalitarian group. Anyone can attend a meeting or register interest in a working group, and when voting, it's one person, one vote. Groups of people from different companies get together to form working groups on a particular problem.

Each working group has a mailing list that you can join. The output of a working group is an "Internet Draft" which is a work-in-progress proposal for a new standard. When finalised these become RFCs (Request For Comments, which are explicitly *not* requests for comments – they're the done deal.) These are often surprisingly readable, and occasionally humorous – for example the technical solution to solve all networking malware: https://www.ietf.org/rfc/rfc3514.txt.

2.3.2 IEEE (Institute of Electrical and Electronics Engineers) (https://www.ieee.org/)

The IEEE issues and manages standards for electrical and electronic devices (including networking devices, cabling, connectors, etc.) The IEEE and IETF operate in similar spaces and work in similar ways (Working groups, mailing lists). The IETF have produced an RFC (of course!) standardising the cooperation. (https://tools.ietf.org/html/rfc7241).

2.3.3 Internet Assigned Numbers Authority (IANA) (https://www.iana.org/)

There are lots of numbers and signifiers that need to be globally unique for us humans to use them (for example, everyone's computer needs a unique address). IANA is the body responsible for organising and arbitrating who gets what. Key things they are responsible for include:

- DNS top level domains. (Who owns www.ox.ac.uk ?)
- IP addresses and ports. (Who is allowed to have IP address 8.8.8.8? Which programs are allowed to use Port 79 on a computer?)
- Protocol assignment. (If a computer says it wants to start using protocol 75 to talk to another computer, what does that mean?)

There are another two standards bodies that you should be aware of, although we will not cover the output of these in the course.

2.3.4 ITU (International Telecommunication Union) (https://www.itu.int/)

The ITU handles defining protocols related to telephony (and telephony over IP). Historically ITU specs are incredibly long and cover everything in minute detail. We won't cover these protocols in the course.

2.3.5 International Organisation for Standardisation (ISO) (https://www.iso.org/home.html)

The ISO orchestrate definitions and standards for everything. For internet-related and networking standards and protocols, they mostly stay clear and endorse/defer to the IETF.

2.4 Underlying meta-constraints

When building networks in the real world, there are 4 fundamental underlying constraints to be aware of. These explain why much of modern computer networks and networking are as they are. To answer the question "Why is this as it is?", the answer is usually one of these.

1. Physics. Ultimately, for information to transfer from A to B in a network, there must be some physical manifestation of that information travelling from A to B. Examples include laser light down an optical fibre, the flow of electricity down a loop of cable, homing pigeons with capsules tied to their legs. Information transfer is ultimately limited by the physical constraints on the system, such as the speed of light orthe speed and accuracy with which

hardware at each end of a link can indicate changes (and detect those changes at the other end!)

2. Money. In networking, there are two questions of money. Capital Expenditure (CapEx) is the is the outlay to build and set up your network. This includes digging trenches and laying cables, purchasing or building hardware, purchasing or writing software. Operational Expenditure (OpEx) is the cost to run your network (the electricity and cooling for your network nodes, the operators configuring and monitoring the network and responding to issues, the people driving vans with spades to dig up and repair cables). At global scale, CapEx and OpEx costs are $billions or $trillions, and even small percentage changes are significant.
3. Politics. The real world is not run by an altruistic group of technical gurus working for the betterment of mankind. There is undoubtedly significant cooperation in networking, but companies want solutions to make money, and states and other organisations want power for and/or over people, and different groups are optimising for success now versus success in future.
4. History. The basics in this course are pretty solid now and unlikely to change fast, and much of the contents of this course would be recognisable to someone who was looking at networking 10-20 years ago or more, and is likely to still be recognisable in 20 years time. This is because global scale networking requires consistency across a huge breadth of companies and people – meaning that changes tend to built on what is already there. Our networks today are built on top of decisions made in the past, and the decisions we make Today will become the problems of tomorrow.

From a history point of view, a glib summary of networking was summarised by Mike O'Dell in 1994: "The only real problem is scaling."

2.5 "Better"

In networking, when we're considering trading off two solutions, what does "better" mean? Leaving aside the meta-constraints above, technically better generally means better in terms of one or more of the following concepts – where again we often have to trade off one against another.

- Bandwidth. Bandwidth is a measure of how much information you can get from A to B per unit time. This is typically measured in (Tera/Giga/Mega/Kilo)bits per second. The units matter and show the scale of differences here. Consider how long it would take to download a modern AAA game (50GB) over a 28kb/s modem…)
- Reliability. Reliability is a measure of what proportion of information going from A to B makes it to B. This is valid both at an individual link level (if A signals 100 bits of data, how many does B pick up correctly) and at a network level (if A is passing a stream of data across a network to B, and one node in that network crashes, what is lost?)
- Latency (propagation delay). Latency is a measure of how long it takes to get a bit of data from A to B.
- Jitter. Jitter is the variation in latency. As an example of how this is important independently of latency, consider a video conference call between Earth and the ISS. With no Jitter, the call is beautifully clear – just with a lag between each end, but with jitter, the picture becomes glitchy with patches of video movement and pauses.

2.6 Terminology / Glossary

Throughout the course there will be various terms used. The core ones are defined here for reference back to during the course. You are not expected to understand all of these right away,

and you may choose to (and are encouraged to) extend this Glossary as you work through the course.

1. Graph Edges
   - **Link**. A link is a point to point connection between two nodes in a network at a particular Layer.
   - **Interface** (also **Port** – though note that Port also has separate technical definition below). An interface is the physical connection on a node where a link enters it.
2. Graph Nodes
   - **Host** (also **Endpoint**). A host is any computer at the edge of a network that is the source or destination of a piece of communication.
   - **Switch** (also **Hub)**. A switch is a Layer 2 node in a network, that connects other nodes together at Layer 2. Technically, when a Hub receives Layer 2 data on an interface, it blindly sends it out *all* other interfaces, whereas a Switch has logic to determine which interfaces to send it out of. Today, switching technology is so cheap, and switching protocols so ubiquitous that Hubs are almost never used.
   - **Router.** A router is a Layer 3 node in a network that connects other nodes together at Layer 3.
   - **Client / Server**. The term Server is generically used in networking to refer to any router or endpoint. Formally many protocols run Client / Server, where the Server runs continuously, with Clients initiating contact and driving the connection.
   - **Peer**. A peer is any other node at the other end of a Link operating at the same Layer. For example, consider two Routers (Layer 3) in a network with a Switch (Layer 2) between them. The Routers are peers at Layer 3, but each router also uses Layer 2 to communicate with the switch and so the routers are also are peers of the switch at Layer 2.
3. Protocols/Addresses
   - **MAC Address (Media Access Control)**. The MAC address is the unique address of an endpoint or switch in a layer 2 Ethernet network.
   - **ARP (Address Resolution Protocol)**. The Address resolution protocol determines the MAC address (Layer 2) associated with an IP address (Layer 3).
   - **IP Address (Internet Protocol Address)**. The IP address is the unique address of an endpoint or router in a layer 3 IP network.
   - **Port**. A Port is a number (0-65535) to allow disambiguation of sessions across multiple programs running on a single host. (Think of them like apartment numbers in a block of flats with one street address). Note that (unhelpfully) Port is also used as a term for a physical interface on a router or switch.
4. Traffic and addressing types
   - **Unicast.** Unicast traffic is sent with a single intended destination, typically uniquely defined by some piece of addressing information within the data being sent.
   - **Broadcast**. Broadcast traffic is sent to every possible receiver in the network (called a broadcast domain). Note that in some circumstances Unicast traffic is broadcast (for example an IP packet sent over a wireless link is literally broadcast through the air and will be picked up by any other receiver). However, the traffic is not Broadcast *traffic* unless the data is intended and addressed such that it is clear that it should be received and processed by every endpoint.
   - **Multicast.** Multicast Traffic is traffic destined for multiple, but not every, endpoint in a network, and is marked with a multicast address, that is not unique to a given endpoint.

Multicast Traffic is typically spread through a network in two ways – it can either be broadcast everywhere (and picked up by endpoints based on the address) or endpoints interested in traffic bound for a particular multicast address can proactively register interest and ask to receive that traffic.