# Networking
# CM30078/CM50123

Russell Bradford

2019–2020

# Introduction

These days we are all networked

Whether on a PC, tablet, phone or other device we spend our time surfing the web, reading emails, streaming media

We find it hard to do anything when were are not connected

We feel cut off when we can't communicate

This Unit is about the technology that allow this to happen, in particular, the Internet

# Unit Outline

Aims To understand the Internet, and associated background and theory, to a level sufficient for a competent domain manager.

# Unit Outline

Learning Outcomes Students will be able to:

Explain the acronyms and concepts of the Internet and how they relate;

State and apply the steps required to connect a domain to the Internet and explain the issues involved to both technical and nontechnical audiences;

Discuss the ethical issues involved with the internet, and have an “intelligent layman's” grasp of the legal issues and uncertainties.

Be aware of the fundamental security issues;

Be able to advise on the configuration issues surrounding a firewall.

# Unit Outline

Syllabus:

The ISO 7-layer model. The Internet: its history and evolution - Predictions for the future.

The TCP/IP stack: IP, ICMP, TCP, UDP, DNS, XDR, NFS and SMTP. Berkeley. Introduction to packet layout: source routing etc.

Various link levels: SLIP, 802.5 and Ethernet, satellites, the “fat pipe”, ATM. versus carrying. Security and firewalls. Performance issues: bandwidth, MSS and RTT; caching at various layers.

# Unit Outline

Who 'owns' the Internet and who 'manages' it: RFCs, service Providers, domain managers, IANA, Jisc/UKERNA, MANs, commercial British activities. Routing protocols and default routers. HTML and Electronic publishing.

Legal and ethical issues: slander/libel, copyright, pornography, Publishing

# Unit Outline

We won't be covering the material in the above order, though, but in a more coherent fashion instead!

# Unit Outline

Structure of this unit: 3 hours lectures per week

Thursday 14.15

Thursday 15.15

Friday 17.15

The aim is to cover the necessary material early in the semester which will leave the last few weeks free for revision and problems classes

# Unit Outline

## Assessment

For the undergraduate CM30078:

End of unit exam 100%

# Unit Outline

## Assessment

For postgraduate CM50123

Coursework 25%

End of unit exam 75%

Coursework timelines (subject to change):

1. set Thu 24 Oct
   due Wed 13 Nov
2. set Thu 14 Nov
   due Fri 6 Dec

Feedback on coursework will be provided via Moodle. There will be general feedback that applies to many people and some individual feedback

# Unit Outline

Week 6 will be a “consolidation week”

No lectures, but only problems classes and labs as appropriate

For the whole of Computer Science (CM Units)

Presumably other Departments will carry on as usual

# Unit Outline

## Resources

Networking is now a mature subject (though still under development and change!) so there are many books available

I recommend

“TCP/IP Illustrated Volume 1” W R Stevens, Addison-Wesley

“Computer Networks, 5th Ed” A Tanenbaum, Pearson (4th Ed OK)

“The Art of Computer Networking” R Bradford, Pearson (Polish Edition: “Podstawy Sieci Komputerowych”, WK )

Stevens is available as an e-book in the library

# Unit Outline

## Resources

You don't need me to tell you that there is a large amount of material out there on the Web?

Wikipedia is fairly accurate in this area: but, as usual with Wikipedia, you should check with other sources

# Unit Outline

## Resources

There is a Unit Moodle page, but as Moodle is so horrible I tend to use my own Web pages:

> https:
> //people.bath.ac.uk/masrjb/CourseNotes/cm30078.html

# Unit Outline

## Content

We will revisit and expand on what you (may have) seen in CM10195 or other units

But is much greater breadth and some detail

# Unit Outline

Networking is a large subject with a lot of complicated detail

And there are very many acronyms

You'll need to remember the main acronyms, but a lot are less important

We shall cover much fine detail in lectures

It is very techy material: if you are not a techy person you should think very carefully about taking this Unit

# Unit Outline

But it is the big picture that is important for this Unit

For example, there are many packet headers that contain lots of flags and fields

You should have a general idea of what the important fields are and what their purpose is, but precisely where they appear in the header is generally less important

# Standard Introductory Slides

Remember:

You are expected to do some work outside of lectures

Lectures are the start of the learning process, not the end!

These slides are reminders to me on what to say in lectures

They are often abbreviated in style, and so are not the whole story and would not be suitable to be quoted verbatim in an exam

# Standard Introductory Slides

Conversely, the slides may contain supplementary material that I shall not be using this year

The exam will only be on material covered in lectures

Your lecture notes will tell you what was or was not covered in lectures

# Standard Introductory Slides

Do not rely purely on my notes for your revision

People who do this live to regret it

Like every Unit, you are expected to read around the subject for yourself

You need to take your own notes, read, and participate

You don't expect to get  t simply by paying to joining a gym. . .

> “If you have college courses in CS, buy the books and spend day and night the few days before class going through the books and taking notes and answering questions and programming examples before the first class even starts. If you really want to do this in your life, that's what you should do, not just wait for the education to be handed you. Those who finish at the top will always be in high demand. You can learn outside of school too but you have to put a lot of time into it. It doesn't come easily. Small steps, each improving on the other, is what to expect, not instant understanding and expertise.”

Steve Wozniak, co-founder of Apple

Computer Science is not a spectator sport

Anon

# Networks

Networks form a central role in the way computers are used today: these days it is very hard to do anything that is not networked

As commerce and big money have taken over the Internet the nature of networking has changed from a way of linking together some CS departments to a multi-billion pound enterprise

Thus a good knowledge of what networks are and how they work is essential to any good Computer Scientist

# Networks

And also to anyone who uses networks as part of their everyday activities

If more people realised quite how open, fragile and snoopable the Internet is, they would be a lot more circumspect in what they do on it!

# Networks

The Internet is familiar to everyone

But networks have been around for a long time

A network is any means to connect entities together so they can communicate

# Networks

Reasons to network include:

- Resource sharing
- Communication and collaboration
- Information gathering
- Reliability through replication
- Entertainment

# Networks

Existing networks include:

- The telephone system
- The mobile phone system
- TV and radio
- System control networks, e.g., Controller Area Network (CAN bus) in cars (and bicycles!)
- Sensor control networks, e.g., Bluetooth and ANT
- Cable (TV) networks
- The Internet

# Networks

Metcalfe's Law

> The value of a network expands exponentially as the number of users increases

The bigger the network, the more links it has

# Networks

There are many different kinds of network, thus meaning we need classifications to put things into easy boxes

# Networks

Classification by size

- LAN Local Area Network
- MAN Metropolitan Area Network
- WAN Wide Area Network
- PAN Personal Area Network, WPAN (wireless PAN)
- and so on

# Networks

Classification by speed

- Narrowband
- Broadband

# Networks

Classification by ~~speed~~ technology

- Narrowband
- Broadband

Actually these technical terms do not denote speed: their real meanings have been distorted by marketing

Optical fibre, while very fast, is actually technically narrowband

Exercise. Find the technical meanings for narrowband and broadband

# Networks

Marketing terms include:

Fast

Superfast

Ultrafast

etc.

No idea what these really mean

# Networks

Classification by technology

- Voiceband modem (V series of standards, V.92)
- Local Wired (Ethernet)
- Medium distance wired. ADSL (ADSL2, ADSL2+ , . . . )
- Optical Fibre (FTTP)
- Hybrid (VDSL with FFTC, G.fast with FTTdp, . . . )
- Cable (DOCSIS)
- Local Wireless (Wi-Fi, Bluetooth, . . . )
- Longer distance wireless (3G, 4G/LTE, 5G, WiMAX, . . . )
- Very long distance wireless: satellite
- Power line
- etc.

# Networks

Continuing Exercise. Find the meanings for the various acronyms

Exercise. Read some adverts for Internet connectivity products and determine what they actually are offering (e.g., “Superfast broadband fibre”)

Exercise. And read about the controversies about how they advertise speeds

# Networks

So what does a typical network look like?

# U Bath Campus Network

# Networks

No hosts shown: this is just the connectivity

Multiple paths between points

Gigabit and 10Gb links

Other big networks, e.g., in CS, are not shown

Connection to rest of world not shown

# South West Regional Network

# Joint Academic Network

# JANET

# Networks

# GÉANT

# CANARIE

# Hierarchy

We can see the Internet is a hierarchy of networks, managed by different groups

department

university

region

country

world

And this delegation of control is essential to the way the Internet works

# Networks

## Important Points

The “Internet” (capital “I”) is the world-wide collection of networks

An “internet” (lower “i”), an abbreviation of “internetwork”, is just some collection of networks

An “intranet” (with an “a”) is some collection of networks belonging to a single organisation

## The Web is not the Internet

Anyone caught saying so will be laughed at and will lose marks in the exam

# Networks

Tim Berners-Lee and Vint Cerf (photos from W3C)

# Networks

The basis of the Internet is collaboration between its member hosts

Data travels from source to destination by being passed from machine to machine

# Networks

```
traceroute to www.youtube.com (208.65.153.238), 30 hops max, 40
byte packets
 1  fire.cs.bath.ac.uk (172.16.0.1) 0.166 ms 0.171 ms 0.216 ms
 2  gw.cs.bath.ac.uk (138.38.108.254) 0.570 ms 0.448 ms 0.337 ms
 3  swan-wren-10g1.bath.ac.uk (138.38.255.1) 0.430 ms 0.470 ms 0.352 ms
 4  7200-bath.bath.ac.uk (138.38.1.1) 1.190 ms 1.431 ms 1.356 ms
 5  fren-bath-ph.swern.net.uk (194.83.94.65) 3.198 ms 2.548 ms 2.515 ms
 6  so-1-3-0.read-sbr1.ja.net (146.97.42.157) 7.978 ms 7.859 ms 8.305 ms
 7  so-1-0-0.lond-sbr3.ja.net (146.97.33.142) 9.287 ms 9.468 ms 9.207 ms
 8  195.219.100.13 (195.219.100.13) 9.320 ms 9.553 ms 9.760 ms
 9  195.219.195.21 (195.219.195.21) 9.458 ms 9.401 ms 9.407 ms
10  ge4-1-0-1000M.ar3.LON2.gblx.net (64.208.110.81) 14.544 ms 17.433 ms
    13.969 ms
11  te1-1-10G.ar2.SJC2.gblx.net (67.17.109.102) 165.984 ms 167.465 ms
    169.402 ms
12  YOUTUBE-LLC.po1.401.ar2.SJC2.gblx.net (64.212.108.162) 165.040 ms
    167.189 ms 165.938 ms
13  youtube.com.hk (208.65.153.238) 165.972 ms 165.825 ms 165.815 ms
```

gblx : Global Crossing; SJC: San José, California
youtube.com.hk is in San José

# Networks

This was done just after a major problem with the route to Youtube (a mistake in Pakistan lead to chaos, February 2008)

Allegedly, the Pakistan government was trying to censor a Youtube video by blocking all routes to Youtube in that country, but the block escaped to the whole Internet

A little later things settled down again. . .

# Networks

```
traceroute to www.youtube.com (208.65.153.238), 30 hops max, 40
byte packets
 1 fire.cs.bath.ac.uk (172.16.0.1) 0.205 ms 0.210 ms 0.091 ms
 2 gw.cs.bath.ac.uk (138.38.108.254) 0.446 ms 0.431 ms 0.341 ms
 3 swan-wren-10g1.bath.ac.uk (138.38.255.1) 1.185 ms 0.841 ms 0.648 ms
 4 7200-bath.bath.ac.uk (138.38.1.1) 1.247 ms 1.062 ms 1.214 ms
 5 fren-bath-ph.swern.net.uk (194.83.94.65) 2.808 ms 2.438 ms 2.653 ms
 6 so-1-3-0.read-sbr1.ja.net (146.97.42.157) 7.839 ms 8.265 ms 7.798 ms
 7 so-1-0-0.lond-sbr3.ja.net (146.97.33.142) 9.526 ms 9.520 ms 9.726 ms
 8 po1-0.lond-gw-ixp2.ja.net (146.97.35.250) 9.672 ms 9.338 ms 9.089 ms
 9 195.66.226.185 (195.66.226.185) 9.804 ms 9.840 ms 9.926 ms
10 te7-3.mpd02.lon01.atlas.cogentco.com (130.117.2.26) 9.823 ms
   te2-1.3493.mpd02.lon01.atlas.cogentco.com (130.117.2.18) 10.223 ms
   te7-3.mpd02.lon01.atlas.cogentco.com (130.117.2.26) 9.685 ms
11 <snip>
19 * * *
20 youtube.com (208.65.153.238) 154.886 ms 156.732 ms 156.480 ms
```

Step 10: multiple probes go different routes

Step 19: a machine that refuses to respond to the probes

Host 208.65.153.238 is now named youtube.com

# Networks

And again on 25 Sept 2017:

```
traceroute to www.youtube.com (216.58.204.14), 30 hops max, 60 byte packets
 1  fire-private.cs.bath.ac.uk (172.16.0.1)  0.109 ms  0.097 ms  0.088 ms
 2  gw-palo.cs.bath.ac.uk (138.38.108.254)  1.055 ms  1.048 ms  1.041 ms
 3  bath-gw-1-palo.bath.ac.uk (193.63.64.174)  1.608 ms  1.800 ms  1.703 ms
 4  xe-1-2-0.bathub-rbr1.ja.net (146.97.144.33)  1.287 ms  1.332 ms  1.330 ms
 5  xe-1-2-0.briswe-rbr1.ja.net (146.97.67.65)  2.286 ms  2.720 ms  2.707 ms
 6  ae22.londpg-sbr2.ja.net (146.97.37.201)  5.189 ms  4.652 ms  4.648 ms
 7  ae29.londhx-sbr1.ja.net (146.97.33.1)  5.089 ms  5.037 ms  5.012 ms
 8  193.62.157.22 (193.62.157.22)  5.270 ms  5.263 ms  5.246 ms
 9  108.170.246.225 (108.170.246.225)  5.938 ms  5.928 ms  5.869 ms
10  108.170.238.145 (108.170.238.145)  5.907 ms
    108.170.238.147 (108.170.238.147  6.141 ms  6.129 ms
11  lhr35s07-in-f14.1e100.net (216.58.204.14)  5.818 ms  5.820 ms  5.798 ms
```

Google are using a local server, probably in London

# Networks

And again on 3 October 2019:

```
traceroute to www.youtube.com (216.58.198.174), 30 hops max, 60 byte packets
 1  fire-private.cs.bath.ac.uk (172.16.0.1)  0.197 ms  0.174 ms  0.149 ms
 2  gw-palo.cs.bath.ac.uk (138.38.108.254)  0.708 ms  0.682 ms  0.661 ms
 3  bath-gw-1-palo.bath.ac.uk (193.63.64.174)  1.776 ms  1.531 ms  1.856 ms
 4  xe-1-2-0.bathub-rbr1.ja.net (146.97.144.33)  1.074 ms  1.061 ms  1.047 ms
 5  xe-1-2-0.briswe-rbr1.ja.net (146.97.67.65)  2.113 ms  2.103 ms  2.092 ms
 6  ae22.londpg-sbr2.ja.net (146.97.37.201)  4.314 ms  4.329 ms  4.274 ms
 7  ae29.londhx-sbr1.ja.net (146.97.33.1)  5.163 ms  5.878 ms  5.854 ms
 8  193.62.157.22 (193.62.157.22)  5.587 ms  5.586 ms  5.544 ms
 9  * * *
10  172.253.71.200 (172.253.71.200)  7.069 ms
    108.170.238.118 (108.170.238.118)  6.627 ms
    172.253.68.210 (172.253.68.210)  6.284 ms
11  74.125.242.114 (74.125.242.114)  8.502 ms
    108.170.232.99 (108.170.232.99)  4.818 ms
    74.125.242.82 (74.125.242.82)  5.622 ms
12  lhr25s10-in-f14.1e100.net (216.58.198.174)  4.574 ms
    216.239.57.207 (216.239.57.207)  6.150 ms
    209.85.250.185 (209.85.250.185)  7.028 ms
```

Now much more variation in routes and multiple servers!

# Networks

## History

The reason for this cooperative design comes from the history of the Internet

- 1957: The Russians launch Sputnik
- mid 1960s: Advanced Research Projects Agency (ARPA) formed. A project to share expensive resources: namely their computers
- The design was to be non-centralised to avoid single points of failure, particularly nuclear attacks
- So there is no single point of coordination or oversight of the network
- And there must be multiple paths between hosts

# Networks

## History

Using simple circuits (such as the telephone system used) between machines would be too vulnerable, so packet switching was devised

Data is chopped into small chunks, called packets, and each packet is sent individually, possibly over different paths

The original data is reconstructed at the receiving host

# Networks

## Warning

The word is “packet”, not “package”

Take care never to use the word “package” in a technical context

# Networks

1969 First Internet has just four nodes

Runs NCP Network Control Program

# Networks

## History

The Original Arpanet, 1969
Separate Interface Message Processors

# Networks

## History

Email and discussion groups are popular

1973 Internet reaches London

1974 TCP/IP replaces NCP

1980s 1000s of machines on the Internet

Domain Name System arrives

# Networks

## History

From “Computer Networks, Fundamentals, Practice”
Bacon, Stokes, Bacon, 1984

# Networks

## History

1980/90 Original ARPANET decommissioned and replaced

Commerce arrives

Other networks based on other protocols are replaced by the Internet

1992 1,000,000 hosts

Gopher

Tim Berners-Lee invents the Web

# Networks

## History

The Internet starts to enter the home

Microsoft gives up on its own network and falls into line

The Dot Com boom

The Dot Com crash

Broadband to the home

Large commerce over the Internet

# Networks

## History

Mobile revolution

. . . what next?

# Networks

Already this decentralised and packet nature has implications on how the Internet must work

- how to chunk the data into packets?
- how are route(s) the packets use to get to their destination found?
- how do we reconstruct the original data as packets might be arriving in any order?

And higher level decisions like how we shall choose and build these multiple routes; what hardware to use, and so on

# Networks

A packet doesn't know how to get to its destination

Even the source host doesn't generally know a route to the destination (only if the destination is on the local network)

A packet is like a postcard with the address written on it: it relies on the routers it passes though to make the right decisions

“Please forward me to www.youtube.com”

It is not like a car driver with their own map making their own decisions!

The question now becomes: how do the routers know what to do? More on this later

# Networks

## Protocols

To ensure maximum interoperability, the Internet relies on standards and standardised protocols

The use of standards means that machine A will be able to communicate with machine B even if A and B are made by completely different companies, are of completely different technologies and have never previously interacted

It is clear that you must have standards for interoperability in hardware: you can't plug a electrical plug into an optical socket

But you must have standards in the software too, as becomes clear when you try to use a Web page authoring tool that doesn't produce standard HTML

Thus we must have standards for the protocols

# Networks

## Protocols

This is somewhat akin to people agreeing all to use English to communicate

A pair of random people meeting can talk if they both know English

If not, the chances are that they share their native languages are quite small

# Networks

## Protocols

Much of the Internet follows a collection of specifications called Request for Comments, or RFCs

RFCs are freely available on the Internet for everyone to read and implement

The RFC philosophy:

> Be as close to the RFC as possible in what you do yourself, but be as liberal as possible regarding what you accept from others

Continuing Exercise. When a topic is covered in lectures, read the relevant RFCs

# Networks

## Protocols

There are several bodies that oversee the structure and working of the Internet and the standards

Internet Society (ISOC); oversees the Internet standard development processes

Internet Architecture Board (IAB); ISOC committee that oversees the technical and engineering development of the Internet, particularly IETF and IRTF

Internet Engineering Task Force (IETF); IAB committee that develops standards and publishes RFCs

Internet Engineering Steering Group (IESG); executive sub-committee of IETF that has final say over RFCs

# Networks

## Protocols

Internet Research Task Force (IRTF); IAB committee that does long-term research and development of Internet technology

Internet Research Steering Group (IRSG); sub-committee of IRTF that manages the research groups

Internet Corporation for Assigned Names and Numbers (ICANN); nonprofit internationally-organised organisation to oversee global resources such as names and numbers

Internet Assigned Numbers Authority (IANA); an affiliate body to ICANN that deals with domain names, IP addresses and other thing, run by a company named “Public Technical Identifiers”

# Networks

## Protocols

IANA delegates management of various things to Regional Internet Registries (RIRs), e.g., domain names and addresses

Current RIRs:

- African Network Information Centre (AfriNIC); Africa
- American Registry for Internet Numbers (ARIN); North America and Antarctica
- Asia-Pacific Network Information Centre (APNIC); Asia, Australia, New Zealand
- Latin America and Caribbean Network Information Centre (LACNIC); South America
- Réseaux IP Européns Network Coordination Centre (RIPE); Europe, Russia, the Middle East, and Central Asia

# Networks

## Protocols

The RIRs further delegate things like management of domain names to commercial companies

E.g., 123-reg, GoDaddy and hundreds of others

Exercise. Trace the movement of money up this hierarchy

# Networks

## Protocols

Others important specification bodies include

IEEE Institute for Electric and Electronic Engineers; hardware like Ethernet and Wi-Fi

ISO International Standards Organisation; e.g., XML standards

IEC International Electrotechnical Commission; e.g., Digital Living Network Alliance (DLNA)

ITU-T Telecommunication Standardization Sector of the ITU (International Telecommunication Union); e.g., DSL standards

lots more national and international institutions

# Networks

## Protocols

There is quite a lot of overlap in what these institutions cover

Sometimes one institution will take a standard from another institution and put a new cover sheet on it and give it a new name or number

For example, the JPEG standard, from the Joint Photographic Experts Group, is the same as ISO standard 10918-6:2013 and ITU-T T.872

Exercise. Investigate these standards bodies

# Networks

## History

So we need a common language (protocols) for the Internet

This common language is called the Transmission Control Protocol/Internet Protocol (TCP/IP)

This name is more historical than accurate, but to see what it means we need to think of layers

# Networks

## Layering Models

What do we need to make two computers communicate?

We need to connect them, so there must be some kind of physical (electrical or other) thing between them

So they must be compatible on voltages, how bits are represented as electrical or optical signals, etc.

And they must agree on how to represent data as bits: recall the different ways of representing signed and unsigned integers; similarly there are several ways of encoding alphabetic characters as bits

And the same problem for all other kinds of data: how to represent that sound or that colour blue?

# Networks

## Layering Models

And how do we make sure data arrived safely and didn't get lost or corrupted in transmission?

And a lot of other problems that only become clear when you try to build a network

Getting this right all at once is very difficult

# Networks

## Layering Models

So how should we implement a network system?

First we need a standard to follow

So how should we design a network standard?

The standard must address all the issues (and more) mentioned previously

# Networks

## Layering Models

This is too big a problem to be tackled all at once

How about chopping the design into chunks, each chunk having a well-defined functionality?

Note this is just the way we approach writing large programs

Except we are not writing a program here, we are designing a standard

So we slice the problem into nice, bite-size pieces, called layers

# Networks

## Layering Models

So what should the chunks be?

A layering model for a system is a suggestion on how you might want to slice up the problem of designing it all

An oft-misunderstood point is that a layering model is not a networking standard

It is a recommendation on how you approach the design of the standard

After you have the standard, you can then make implementations

# Networks

## Layering Models

So:

We pick a layering model

We use this to guide us in making a standard

We use the standard to direct the implementations

We can then use the implementations

Note that there will likely be several differing implementations

But, if it is a comprehensive standard, and if they follow the standard, the implementations will interoperate

# Networks

## Layering Models

For networks, there are two main Models in use: the ISO Open Systems Interconnection (OSI) Seven-Layer Model; and the Internet Four-Layer Model

That is: two popular recommendations on how to design a networking standard

The OSI model is widely used while the Internet model is not, despite closely mirroring the Internet standard

# Networks

## The OSI Model

The seven OSI layers are

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

# Networks

## The OSI Model: Physical Layer

The physical layer (PHY) or layer 1 is the hardware layer and deals with the transmission of bits over a channel

For example:

> what voltages to use or colours of light pulses or radio wavelengths to use
>
> what encoding for bits; how long (in time) a bit should be
>
> how many wires to use in a cable
>
> what plugs to use on the cable
>
> and many more

Generally, anything to do with choices regarding hardware

# Networks

## The OSI Model: Data Link Layer

The data link layer, also called the media access layer (MAC) or layer 2, takes the physical layer and tries to create a channel where there are no undetected errors of transmission

Note “undetected”: we know networks are not 100% reliable (e.g., wireless networks) so we must take into account possible errors and deal with them: the ISO standard recommends you think about that here

A typical MAC layer sends the data as a sequence of frames (recall the packet nature of the Internet). A frame is a chunk of bytes, maybe tens or thousands of bytes long

# Networks

## The OSI Model: Data Link Layer

If a frame is corrupted, maybe the MAC layer can resend it; or send a message to the next layer indicating a problem

A popular choice is to do nothing at all: let a higher layer figure out what's gone wrong and choose a remedy

Again: it is up to the standard we are designing as to what actually happens. The layering model just says it is a good idea to consider this kind of thing here

In real implementations, this layer is often strongly intertwined with the physical layer

# Networks

## The OSI Model: Network Layer

The network layer, layer 3, controls the operation of the network, particularly the issue of routing data from source to destination

Also, it can deal with congestion: where there is too much data for a particular link it might route some data via another link, or use flow control to slow down the rate of transmission

Or speed up the rate if things are going well

Accounting might be managed in this layer: counting the number of bits so we can bill the user

And quality of service: e.g., ensuring there is always enough bandwidth to stream a video

# Networks

## The OSI Model: Transport Layer

The transport layer, layer 4, accepts data from the session layer (layer 5) and arranges it into packets suitable for the network layer: packetisation

Similarly, it takes packets from the network layer and reassembles it into the original data stream: depacketisation. This will need to deal with packets arriving out of order

Reliability: ensuring the data received is the same as the data sent. No corruption or loss in the data

Curiously, reliability is not always a requirement of a network!

# Networks

## The OSI Model: Session Layer

The session layer, layer 5, manages sessions between source and destination.

- Establishing and terminating connections; e.g., a remote login session
- Restarting interrupted connections

Sessions can be quite short, e.g., just long enough for an email or Web page to be transmitted; or arbitrarily long

In general, a session is just some logically connected set of exchanges that have some unified identity

# Networks

## The OSI Model: Session Layer

For example, if the network crashes and reboots halfway through a big data transfer, the session can be picked up from where it left off, rather than starting again

You may already know that protocols like HTTP don't automatically pick up from where they left off

This tells us there is possibly a gap or omission somewhere: something they didn't address in the design

This may have been through deliberate choice; but more likely it's just they didn't think about it

# Networks

## The OSI Model: Presentation Layer

The presentation layer, layer 6 provides some things so that we don't have to reimplement then in every application

In particular, it decides on representations of data, such as characters, integers and floating point values, so that the source and destination can agree on the data communicated

# Networks

## The OSI Model: Presentation Layer

So if the source wants to send the number 42, the presentation layer deals with encoding this in a suitable way as (say) some bits, which are then transmitted (passed to layer 5)

And the destination presentation layer can determine that this particular sequence of bits it has just received represents the number 42

They can agree on “42” regardless of how each host chooses to represent integers internally

# Networks

## The OSI Model: Application Layer

The application layer, layer 7, is the layer application programmers use: ideally they would not have to worry about lower layers in their application

It contains protocols like HTTP for the Web, SMTP for email, and so on

Built on top of these protocols are the applications that the users see, e.g., Firefox or Chrome for the Web, Pine or Thunderbird for email

# Networks

## Layering Models

Conceptually, data from an application is passed down through the layers until it reaches the hardware: i.e., through a sequence of pieces of software that perform the functions of each layer

As it passes from later to layer it is encapsulated: a transformation of the data in such a way that the layer below can cope with it transparently

And in a way that it can be untransformed back again

# Networks

## Layering Models

At each layer, the transformation might

- add an identifying header or trailer or both that is needed for the functionality of the layer
- encode any bit patterns that might be misinterpreted or mis-transmitted by the next layer
- put items in a standard form, e.g., integers into a well-known format
- do some arbitrarily complicated manipulation
- do nothing at all!

# Networks

## Layering Models

A possible (but unlikely) OSI encapsulation

# Networks

## Encapsulation

An example. Some early modems treated byte values less than 32 as commands to the modem, not data to be transmitted

E.g., value 4 might mean “end of transmission” and the modem should drop the connection

What do you do if your data happens to contain the value 4?

You can't just send it, as the modem would interpret the data as a command and end the connection

# Networks

## Encapsulation

So you need to transform the data somehow so that “4” is never seen by the modem in the datastream

And the transformation must be reversible, so the other end can reconstruct the 4

This is why encapsulation is necessary: so data can be transmitted accurately, even if you are using weird hardware

# Networks

## Byte Stuffing

In this case, the transformation often used was byte stuffing : the link layer could replace byte value “04” by, say, a pair of bytes “DB D4” (hexadecimal)

Both bytes will be transmitted unmolested by the modem

The link layer at the other end could recognise this pair and replace it by the single byte “04”

The “DB” is called an escape character, and its presence in the datastream means the next character is encoded, so special action must be taken

# Networks

## Byte Stuffing

Take a while to think of the issues this raises: what happens if our original data contained the pair of values “DB D4”?

Not only do the bytes under 32 need to be stuffed, so does the escape character

For example, “DB” in the original data could be stuffed as “DB FF”

The datastream “DB D4” becomes “DB FF D4”

With byte stuffing, we exchange some expansion of the data for the correct transmission of that data

# Networks

## Layering Models

Say you want to send an email. In a strict implementation adhering to the layers the following might happen

- The email application might add a standard email header (From, To, etc.)
- This is passed to the presentation layer. As far as this layer is concerned it gets a chunk of text from the application layer
- It doesn't (or shouldn't) know that the first few characters are a email header
- It may transform the characters in some way, e.g., putting characters into a standard format; it might prepend its own header

# Networks

## Layering Models

This is passed to the session layer. As far as this layer is concerned it gets a bunch of bits from the previous layer

It doesn't (or shouldn't) know that the first few bits are a layer header

It may transform the bits in some way; it might prepend a header

And so on down through the layers

Eventually, the physical layer transmits some bits

# Networks

## Layering Models

At the destination a bunch of bits is received by the hardware

We now proceed up the layers, unwrapping and untransforming as we go

And, eventually, we get the original data arriving at the application (we hope)

So why do this as it seems so wasteful?

# Networks

## Layering Models

If the original data are small the data transmitted on the wire can be mostly headers from the various layers

Encapsulation overhead

# Networks

## Layering Models

Surely it is easier just to put the original data on the wire?

> Encapsulation adds complexity to the implementation
>
> It adds overhead (both space and time)
>
> thereby reducing effective throughput

But it turns out layering and encapsulation actually reduces overall complexity, just like breaking a large program into functions/objects/whatever does for programming

It also gives flexiblity

# Networks

## Layering Models

Suppose we have a 1Gb network card in our machine and someone comes along with a 10Gb card

Because the physical layer is (mostly) separate from the data link layer we can just write a new standard for the 10Gb physical layer and slot it in where the old 1Gb standard was

The upper layers needn't know anything has changed

And we can slot in the implementation for the new hardware in exactly the same way

We don't have to rewrite our email application (and Web browser, and. . . ) because of the upgrade

# Networks

## Layering Models

Similarly for all the other layers: we can replace layers and implementations of those specifications without affecting the rest of the stack

In principle, you could use carrier pigeons as the physical layer and your browser should work unchanged

Apart, perhaps, from speed

Someone really did this once!

Exercise. Read RFC1149

# Networks

## Layering Models

And as each layer simply hands over to the next, it doesn't actually matter what the next layer “really” does

As long as it has the right behaviour, it doesn't matter how it is actually implemented

This enables useful tricks like tunnelling, which we shall look at later

# Networks

## Layering Models

Why seven layers in the ISO model?

History is a trifle vague on this, but it seems that IBM had a seven layer protocol and managed to persuade the ISO committee in charge of the model

Some people advocate more layers: e.g., splitting the hardware layer up

For example, a sublayer describing the physical medium, such as copper or fibre; and a sublayer describing the signals in that medium, such as various kinds of electrical signalling

Exercise. Reality is more complicated. Read IEEE 802 to see how the physical layer can be split into three sublayers; and the link layer can be split into two sublayers

# Networks

## The Internet Model

Others want fewer layers. A good example is the Internet Model

This is a four layer model, developed post-hoc after the Internet Protocol had gained prominence (RFC1122)

Link Layer

Network Layer

Transport Layer

Application Layer

# Networks

## The Internet Model

We shall describe this model, together with its primary instance TCP/IP

Take care to distinguish between the model and the instance

They often get confused as they seem so similar

It is possible, though unlikely, that there could be another network protocol, not TCP/IP, based on the four layer Internet model

# Networks

## The Internet Model: Link Layer

The Internet link layer corresponds to the OSI physical plus data link layers

The model does not say much about this layer, only that it has to be capable of sending and receiving the next layer packets

So what you do with your hardware is pretty much open

TCP/IP many realisations here, including Ethernet, ADSL and Wi-Fi

# Networks

## The Internet Model: Network Layer

Also known as the Internet layer, the network layer handles the movement of packets, particularly routing

This directly corresponds to the OSI network layer

In TCP/IP, the Internet Protocol (IP) is defined in this layer

IP is an unreliable protocol. This is a technical term that means that it does not guarantee delivery of packets

unreliable = not guaranteed reliable

# Networks

## Aside: Reliability

Sometimes it is better to deal with an occasional lost packet than to hold up the system while the lost packet is re-requested and resent, e.g., video, where fast delivery is more important than accurate delivery

So it is quite useful to have a “unreliable” delivery sometimes

Most Internet hardware is actually pretty reliable (non-technical sense) these days

But wireless (Wi-Fi, etc.) and some wired (ADSL) are more unreliable than you might think

# Networks

## The Internet Model: Transport Layer

The transport layer corresponds to the OSI transport layer, providing a flow of packets between source and destination

In TCP/IP, two protocols are in this layer: the transmission control protocol (TCP) and the user datagram protocol (UDP)

# Networks

## The Internet Model: Transport Layer

TCP is a reliable (guarantees delivery) protocol

It makes a reliable layer out of a potentially unreliable IP underneath by a complex mechanism of packet acknowledgements

We don't always want to pay the non-trivial cost of that mechanism, so the other protocol, UDP, is not reliable

Actually, it is reliable as the underlying layer, IP, is reliable

And IP is as reliable as its underlying physical/datalink layer is reliable

# Networks

## The Internet Model: Transport Layer

UDP was devised long after TCP when it was realised how useful unreliable protocols can be: this is why the protocol set is called “TCP/IP”, as that was the entire protocol set for a fair while

# Networks

## The Internet Model: Transport Layer

We shall see packets have a header field indicating what the protocol of the data is

TCP has protocol number 6

UDP has protocol number 17

Exercise. Have a look at /etc/protocols or “Protocol Numbers” at https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml

# Networks

## The Internet Model: Application Layer

The application layer covers (roughly) the OSI session, presentation and application layers

This means, in particular, Internet applications must take care over presentation issues if they want to be completely interoparable

Many forget this, e.g., many programmers forget that not all machines represent integers in the same way and so the bit pattern they use for the number they want to send is (mis)interpreted as a different number by the receiver

# Networks

## The Internet Model: Application Layer

In terms of implementation, typically an OS kernel will implement everything below the application layer (TCP, UDP, IP, Ethernet, Wi-Fi, etc.)

This is because they use system resources that must be shared fairly amongst applications

Anything above the transport layer must be done by the application programmer in their application code

# Networks

## The Internet Model: Application Layer

So a typical email application will need to apply a presentation encapsulation and add application layer headers (To, From, etc.)

The Multipurpose Internet Mail Extensions (MIME) standard is a way to encode data (e.g., text, sound, pictures, video) in a safe way

Originally developed in the context of email, it is now used in other areas like Web page delivery where there are mixed kinds of data to transmit

# Networks

## The Internet Model: Application Layer

Similarly for the session layer

If a persistent session is needed, the application must code it

Many applications, like HTTP, don't

Note: if the TCP/IP had session management, applications would get this “for free”

The counter-argument is that many applications do not want session management, and should not have to pay the overhead of supporting it

# Networks

## The Internet Model: Application Layer

In the real world, each application (running over TCP/IP) that needs session management has to re-implement it for itself

Of course, libraries of code exist to do these “missing” things (sessions, presentation and so on), but the programmer must write the code to incorporate them

# Networks

Example of layering: how an email might be transmitted over an Ethernet

We start with the text of the email

Application: the email application transforms the text using a MIME encoding (presentation)

Application: the email application adds an envelope header (From, To, etc.)

Transport: TCP adds its header (reliability)

Network: IP adds its header (routing)

Datalink: Ethernet adds a header (local routing) and a trailer (checksum)

Datalink: The bits are transformed using a 4B/5B encoding to smooth the bit patterns and are sent using a three-level electrical coding MLT-3 (physical)

# Networks

Going through all these in detail is the content of this Unit

# Networks

But first: we have two layering models, two approaches to designing a standard

How do they compare?

# Networks

## Layering Models

OSI vs. Internet Models

# Networks

## Layering Models

Comparing the two models:

- OSI was developed before an implementation; the Internet Model was created after TCP/IP
- OSI make a clear distinction between model and implementation; Internet is fuzzy
- OSI is general and can apply to many systems; Internet is specific, namely to TCP/IP
- Implementations following standards following the OSI model were dire; TCP/IP is wildly successful

# Networks

## Layering Models

Problems with the Internet Model (not TCP/IP) include

- it is only good for describing TCP/IP
- the physical and data link layers are merged; this makes it difficult to talk about, say, copper vs. optical fibre installations

# Networks

## Layering Models

Non-problems include

> “OSI is slower as data has to go through more layers”

This is confusing the model with the implementation and ignoring the standard in between them

An implementation need not have 7 separate modules: it only needs to behave as if it did

Early implementations of a standard derived from OSI made this mistake

There are good CS reasons why we should do this separation, but practically we have to make tradeoffs between maintainability and speed

# Networks

## Layering Models

“OSI has larger encapsulation overhead as data has to go through more layers”

As above

And you don't have to add a header at every layer: it depends on what the standard (not the OSI model) requires

# Networks

## Layering Models

“There are no decent implementations of OSI”

Again, confusing a model with a standard

And TCP/IP can be regarded a standard that fits the OSI model, anyway

If you squint a bit

# Networks

## TCP/IP

The OSI model is widely used; the OSI protocols never

The Internet model is rarely used; the TCP/IP protocols are everywhere

The main reason that TCP/IP is so successful is that its standards (RFCs) are open and freely available: anyone can join in

Furthermore, the code was also free and widely available

Not brilliant quality, but at least it worked. . .

# Networks

## TCP/IP

Networks before the Internet tended to be closed and proprietary, where you had to pay to get in

But all these failed to get critical mass: even Microsoft failed to get their own alternative to the Internet off the ground and they had (grudgingly) to join with the rest of the world in using TCP/IP

# Networks

## Layering

Other layering models exist, e.g., Tanenbaum's Five Layer Model

Physical

Data link

Network

Transport

Application

Still missing presentation, but a lot more useful in a world where the physical layer is often changed, e.g., 100Mb Ethernet to Wi-Fi

# Networks

## Layering Models

Exercise: identify the OSI and Internet layers as they apply to a cup-and-string network

# Networks

## Security in the IP

Despite being initiated by the Military (ARPA), the Internet was mostly designed(?) and developed in Academia

This has had a great effect on the security of the Internet

The Internet was developed in a “safe” academic environment where little regard was given to issues of privacy or authentication

And the models are also weaker on security than they ought to be

# Networks

## Security in the IP

OSI says “security should be involved at all layers”. Not particularly helpful

The Internet Model says even less

Early TCP/IP implementations were woefully poor

# Networks

## Security in the IP

By default:

- Data in transit is readable as it is passed through the various machines on the path to the destination
- Many protocols used are not resistant to malicious interference
- Authentication mechanisms are weak to non-existent

And the implementations were very fragile and easily hacked

# Networks

## Security in the IP

Note the two separate issues here:

- the protocols are fragile and easily breakable
- the implementations of those protocols were often poor

A good implementation of a bad protocol is bad

A bad implementation of a good protocol is bad

# Networks

## Security in the IP

Many of these issues have since been tackled (not always successfully), particularly when commerce got involved

But there are still several areas that could be improved: see the routing to Youtube problem earlier; and that wasn't even maliciously intended

New protocols and secure (we hope) extensions to existing protocols are now available: e.g., HTTPS for the Web, SMTPS for email

Management and use of cryptography has an overhead. This is an extra workload on servers: some people are unwilling to pay this price

More on this later

# Long term plan

We shall now work our way up the layers, looking in detail at what TCP/IP does for each

This is going to be a long journey!

# Networks

## Hardware

First, hardware

There are several popular hardware implementations. Some you should have come across are

- Ethernet: a wired network
- ADSL: telephone networks
- Wi-Fi: a short range wireless network
- Cellular: mobile phones

We shall look at some of these

# Networks

## Hardware

Exercise. How many different wireless systems does your mobile phone support?

# Networks

## Ethernet

Ethernet arose in 1982, from DEC, Xerox and Intel, based on the earlier Aloha protocol

The original Ethernet supported 10Mb/s

Note: Mb/s = megabit/sec; MB/s = megabyte/sec

It used carrier sense, multiple access with collision detection (CSMA/CD)

Current top-end Ethernet runs at 100Gb/s, with 400Gb/s coming soon and plans for 1Tb/s

More precisely, the original Ethernet had a 10Mb/s signalling rate

# Networks

## Ethernet

The signalling rate is the rate of delivery of bits across the physical network

Due to layering encapsulation and other physical overheads, this is overwhelmingly not the rate of delivery of bits to the application you are running

For example, there is always a gap between packets where data is not being transmitted!

However, this is the number marketers like to use

The rate actually realised can be much lower; e.g., a 54Mb/s Wi-Fi network might only deliver half that figure to an application

# Networks

## Ethernet

The Ethernet standard covers both the PHY and the MAC layers, so we shall look at them together

And we begin with the frame format

# Networks

## Ethernet

An Ethernet frame:

Numbers are byte counts: so the destination address is 6 bytes long

2 byte type indicates what kind of (network layer) data follows, e.g., (hex) 0800 for an IP packet

The data, maximum 1500 bytes

Minimum 46 bytes . The data must be padded with extra bytes if fewer than 46 bytes are supplied

# Networks

## Ethernet

A higher layer must detect and remove this padding when necessary

4 byte checksum, also called cyclic redundancy check (CRC)

Use to check for corruption errors in the frame

# Networks

## Ethernet

How is a frame matched up to the intended destination host?

(Original) Ethernet is shared, so every host sees every frame on the local network

However, every Ethernet card has a unique address built into it

(Not the full story, but true enough for now)

# Networks

## Ethernet

So the destination address allows an Ethernet card in a host to recognise that a frame is for it and so can read and process it

There is a security issue here. . .

The source address allows a host to determine who sent the frame and so it can reply if needed

000010000000000000100000100110100011010011011101 is an example Ethernet address, a 48-bit value

For convenience we write this as 08:00:20:9a:34:dd , six hexadecimal numbers

# Networks

## Ethernet

This is address is ok for when the destination is on the local Ethernet network: we have to work harder if the destination is non-local

The destination might not be on an Ethernet, so how can we specify such a destination?

This is the job of the next layer, IP, which we look at later

# Networks

## Ethernet CSMA/CD

Ethernet is a multiple access (shared) medium, meaning that several hosts use the same piece of wire to send data to one another

# Networks

## Ethernet CSMA/CD

Ethernet is a multiple access (shared) medium, meaning that several hosts use the same piece of wire to send data to one another

Original Ethernet

Suppose A wishes to send to B

If C is already sending to D, the whole network is occupied with its signal, so A must wait

# Networks

## Ethernet CSMA/CD

If two hosts try to send simultaneously, there will be a collision

This is an actual physical condition where the electrical signals from the two hosts get mixed and thus corrupted

So before they send data, a host listens to the Ethernet to see if anyone else is using it at the moment: carrier sense

If not, it sends the data

Otherwise it must wait, listening until the carrier is free

# Networks

## Ethernet CSMA/CD

This still isn't quite enough

So each host continues to listen while transmitting to make sure there are no collisions: collision detection

# Networks

## Ethernet CSMA/CD

If a collision is detected, each host stops transmitting, waits a (small) random period of time and retries with the carrier sense

The random wait means that another collision is less likely as the one host will come in slightly later and see the other's signal in its carrier sense phase

Detecting collisions on an Ethernet is simple: if the signal you are seeing on the network is not the same as the signal you are putting on the network, that means someone else is transmitting, too

# Networks

## Ethernet CSMA/CD

Exercise. What if there are three hosts? Explain why we need to go back to carrier sense after the random pause

Exercise. Read further about jamming signals and what to do if the transmission repeatedly fails

# Networks

## Ethernet CSMA/CD

Collision detection is why there is a minimum frame size

The frames must be on the wire long enough that the hardware can detect a collision

The speed of the signal in the wire is the problem here!

And this is made worse with later faster Ethernets

Exercise. Find out how CSMA/CD differs from Aloha

# Networks

## Physical Ethernet

There have been many Ethernet physical layers

| Standard | cable | max size | rate |
|---|---|---|---|
| 10Base5 | Thick coax | 500m | 10Mb/s |
| 10Base2 | Thin coax | 200m | 10Mb/s |
| 10BaseT | Twisted pair | 100m | 10Mb/s |
| 10BaseF | Fibre optic | 2000m | 10Mb/s |

Base means baseband, namely using a single chunk of frequencies from 0 (the base) up to a single cut-off point

# Networks

## Physical Ethernet

And these evolved (just a selection here):

| Standard | cable | max size | rate |
|---|---|---|---|
| 100BaseT4 | Twisted pair | 100m | 100Mb/s |
| 100BaseT | Twisted pair | 100m | 100Mb/s |
| 100BaseF | Fibre optic | 2000m | 100Mb/s |
| 1000BaseT | Twisted pair | 100m | 1Gb/s |
| 2.5GBaseT | Twisted pair | 100m | 2.5Gb/s |
| 5GBaseT | Twisted pair | 100m | 5Gb/s |
| 10GBaseT | Twisted pair | 100m | 10Gb/s |

# Networks

## Physical Ethernet

The cables used in these PHYs change over time. Unshielded Twisted Pair (UTP) comes in various qualities:

Category 1: No performance criteria

Category 2: Rated to 1 MHz (used for telephone wiring)

Category 3: Rated to 16 MHz (used for Ethernet 10BaseT)

Category 4: Rated to 20 MHz (used for Token-Ring, 10BaseT)

Category 5: Rated to 100 MHz (used for 1000BaseT, 100BaseT, 10BaseT)

# Networks

## Physical Ethernet

These days we use

- Category 5e (enhanced): Rated to 100 MHz with better crosstalk specification than 5 (used for 2.5GBaseT, 1000BaseT)
- Category 6: Rated to 250 MHz (used for 5GBaseT and 10GBaseT up to 55m)
- Category 6a (augmented): Rated to 500 MHz, extra crosstalk shielding (used for 10GBaseT up to 100m)

# Networks

## Physical Ethernet

Cat 5e and Cat 6 is what you will find most widely used today (they are roughly the same price while Cat 6a is a lot more expensive)

The NBASE-T Alliance claims “an estimated 70 billion meters of cabling, which is more than 10 trips to Pluto” has been installed

So people are trying hard to make new Ethernet standards that don't require ripping out the old cabling and installing new

Thus we have intermediate curiosities like 2.5GBaseT and 5GBaseT (standards developed after 10GBaseT)

The higher speeds and more expensive cabling is usually found only in specialist installations like data centres, HPC and Internet exchanges

# Networks

## Physical Ethernet

Down the line are (probable figures)

Category 7: 600 MHz (10GBaseT)

Category 7a: 1000 MHz (40GBaseT, 100GBaseT)

Category 8.1 and 8.2: 1600-2000 MHz (40GBaseT, 100GBaseT)

# Networks

## Ethernet

Evolution of Ethernet

# Networks

## Ethernet

Twisted pair uses hubs or (these days) switches

Hubs were simple electrical repeaters. An incoming signal is sent out on all outputs

There is a single collision domain as all hosts see all signals: any pair of signals between any hosts will collide

The available bandwidth is shared amongst all the hosts

# Networks

## Ethernet

A switch understands the link layer. It only sends the signal out on the single wire that has the destination host

# Networks

## Ethernet

This requires the switch to read and understand the MAC addresses in the frames and to track the socket where each host is plugged in

This is extra complexity in the switch hardware, but reduces the number of possible collisions, increasing throughput

Each output cable is now a separate collision domain

The full bandwidth is available on each output, simultaneously

Collisions only if two hosts send to the same destination simultaneously

# Networks

## Ethernet

If an output is busy, rather than have a collision, a switch may choose to store and forward a packet later when that output is free

Now there can be no collisions and we might think we can do away with CSMA/CD

But buffers in the switch can fill up and then packets would have to be dropped by the switch

So the switch can send a jamming signal on an input to get it to back off and resend later: thus still using CSMA/CD

# Networks

## Ethernet

Some switches can cut through, sending the start of the packet onwards before the tail has arrived

Less latency through the switch, but would forward corrupted packets that store and forward would discard

Switches can run full duplex, with independent inward and outward traffic to each host

This gives twice the total bandwidth of previously

No collisions are possible between opposing traffic as inward and outward traffic runs over different twisted pairs (below 1Gb)

# Networks

## Ethernet

Ethernet is moving faster: 10Mb/s to 1Gb/s and more, all using the same basic CSMA/CD protocol, but using differing electrical signalling

Ethernet cards can autonegotiate to select optimum speed

But it's not just a case of increasing the frequency of the signal, there are other complications to get around the limitations of the cables

# Networks

## Ethernet

1Gb/s Ethernet is everywhere, while 10Gb/s Ethernet is currently gaining popularity

40Gb/s and 100Gb/s Ethernet are available (really only useful for data centres)

These faster rates are mostly optical fibre, but 40Gb/s can run over Cat8 twisted pair

# Networks

## Ethernet

Exercise. Read about Ethernet to the Home (ETTH); also called Ethernet in the first mile , mostly working over optical fibre, designed to deliver networking to the home (just data, no voice)

# Networks

## Ethernet

What are the physical encodings of bits on a 10Mb/s Ethernet?

A simple way would be 0V for 0 and 1V for 1, running at 10MHz

But this has a number of problems

# Networks

## Ethernet

1. An empty network and a stream of 0s looks the same

And so you could not do carrier sense

2. Bits need to be synchronised to prevent drifting out of step (was that 1000 or 999 0s?)

3. A long stream of 1s is a steady 1V: this is electrically a bad design, an average 0V is best

To connect devices easily you need an AC signal, not a DC one

# Networks

## Ethernet

So 10Mb/s Ethernet uses a Manchester Encoding

- Split the time interval for a bit into two parts
- Low then high voltage is a 0
- High then low voltage is a 1

So the average is 0V

-0.85V for low, +0.85V for high

Easy to synchronise: transit through 0V is the middle of a bit

# Networks

## Ethernet

This does double the frequency of the signal to 20Mhz

We can use Cat 4 (or better) cable for this

Manchester encoding solves the above problems neatly and actually simplifies the hardware needed

It is described as self clocking, as the reading end does not need a clock to determine where the bits are

# Networks

## Ethernet

What of 100Mb/s Ethernet?

We can't use even Cat 5 cables with Manchester as it is only specified to 100MHz, and we would need 200MHz

# Networks

## Ethernet

Instead we start by encoding 4 data bits as 5 physical bits in a 4B/5B encoding; e.g., 0000 become 11110

| Input | 4B/5B | Input | 4B/5B |
|---|---|---|---|
| 0000 | 11110 | 1000 | 10010 |
| 0001 | 01001 | 1001 | 10011 |
| 0010 | 10100 | 1010 | 10110 |
| 0011 | 10101 | 1011 | 10111 |
| 0100 | 01010 | 1100 | 11010 |
| 0101 | 01011 | 1101 | 11011 |
| 0110 | 01110 | 1110 | 11100 |
| 0111 | 01111 | 1111 | 11101 |

With some control patterns, e.g., IDLE 11111.

# Networks

## Ethernet

Hasn't 4B/5B made things worse: 5 bits where there were 4?

But now we use a three level physical encoding MLT-3

This has +, 0, and - levels (  0:85V), again using transitions to encode bits

# Networks

## Ethernet

Transitions are cyclical

- to 0<br>
0 to +<br>
+ to 0<br>
0 to -

A transition marks a 1, no transition marks a 0

The 4B/5B translation ensures that every chunk of 5 symbols has at least two transitions, so average voltage is roughly 0

E.g., input 0000, with no transitions becomes 11110 with four transitions

# Networks

## Ethernet

An example. Hex value 0E = 0000 1110

# Networks

## Ethernet

Some words:

A physical representation is called a symbol

Symbols need not be binary

And need not represent a whole number of bits

The baud rate is the number of symbols per second

# Networks

## Ethernet

100Mb/s Ethernet runs at up to 31.25MHz for a symbol rate of 125MBaud: all 1s output (IDLE) is four transitions (- to 0, 0 to +, + to 0, 0 to -) per cycle
(4 symbols/cycle  31:25MHz = 125MBaud)

This has a symbol rate of 125MBaud for a data rate of 100Mb/s: 80% efficient or 1 physical symbol is 4 =5 = 0:8 bits

# Networks

## Ethernet

For Gigabit Ethernet 1000Base-T: 8 bits become 4    3 physical bits in a continuously changing encoding (not a table lookup)

Each 3 bit chunk is encoded using transitions between 5 levels (PAM-5)

Over all four pairs in the cable simultaneously, in both directions on all pairs

10Gb Ethernet uses a PAM-16 over a very complicated coding (Tomlinson-Harashima Precoding)

(SATA and USB 3.0 use 8B/10B; USB 3.1 uses 128B/132B; etc.)

# Networks

## Analogue

Before digital networks were common, the physical layer of choice was an acoustic modem, using the existing analogue telephone network

This used MOdulation and DEModulation to convert bits into acoustic symbols, i.e., sounds

The early Internet (Arpanet) ran over the existing analogue telephone network

# Networks

## Analogue

Exercise. Read about the V series of modem standards

Exercise. Read about amplitude modulation, frequency modulation and phase modulation and Quadrature Amplitude Modulation (QAM) constellations

# Networks

## Digital

After analogue, public telephone systems started to support purely digital networks

Exercise. Read about Integrated Services Digital Network (ISDN)

# ADSL

Asymmetric Digital Subscriber Line (ADSL) is a popular method of delivery to the home

Analogue modems are limited to 56Kb/s, the maximum speed available from a standard analogue telephone line where all frequencies apart from a 3KHz chunk centred on the human voice are filtered out and thrown away

The telephone wire — while only originally specified to be capable of sending voice — is capable of more, ADSL tries to take advantage of this

# ADSL

The data rate you get depends on the length of the cooper loop connecting you to the telephone exchange: the longer it is the harder it is to get a clean signal down it

It tops out at 24Mb/s, dropping to 2Mb/s at the longest reach

It is asymmetric in that is divides the available bandwidth unequally into (say) 24Mb/s downstream (towards the user) and 2Mb/s upstream (towards the Internet)

Which is what most home users want: a few clicks on a Web link (low bandwidth) resulting in a large page download (high bandwidth)

Exercise. ADSL is just one in a series of DSL standards, collectively called xDSL. Read about these

# Fibre

A brief word on optical fibre

Ideally we would each have a high-bandwidth optical fibre to our home

Optical fibre is not subject to electrical interference like copper wires, and can carry huge (terabits is possible) datarates

# Fibre

It would be very expensive to provide everybody with a fibre connection: a lot of digging up the road would be needed

The copper telephone network was put into place over many decades

Though progress is continuing and there is preliminary talk about decommissioning the copper network at some point in the distant future

The UK government has announced that there will be 100% coverage of gigabit “broadband” by fibre or 5G by 2025

Previously it had an “aspiration” to have 100% optical fibre by 2033 — much more realistic

# The Last Mile Problem

This is part of last mile problem: how to bridge the gap between the local telephone exchange and the final user

Also called the first mile problem

# Fibre Hybrid

We would like Fibre to the building/business (FTTB) or Fibre to the premises (FTTP), where fibre comes to a building (business or multiple occupancy building); or Fibre to the home (FTTH) where fibres come to individual occupancies

Currently the popular solution is to lay new Fibre to the street cabinet (FTTC) and then use a DSL over the existing copper wire (the copper loop) to the home

VDSL2 is used on the copper from the cabinet to the home: with an “up to 80Mb/s” downlink

# Fibre Hybrid

The distance you live from the distribution cabinet now governs what speed you really get

In contrast, ADSL uses the cooper loop from the exchange to the home: this can be many kms, making the maximum achievable bandwidth about 24Mb/s

The FTTC “Fibre broadband” hybrid represents the current cost-effective way of getting decent bandwidth to the home

FTTH/P is sometimes marketed as “full fibre” to distinguish it from FTTC

# Fibre Hybrid

Other developments: Fibre to the distribution point (FTTdp) lays new fibre from the cabinet to the (normally underground or on a telephone pole) distribution point where the cable bundle currently splits into individual wires to the premises

A box at the DP converts the optical signal to an electrical signal along the final copper loop to the premises

Using G.fast, another DSL for very short loops: up to 500m

# Fibre Hybrid

G.fast offers speeds up to 1Gb/s, but typically achieves a few 100s of Mb/s

Using frequencies that are sufficiently high that they will interfere with FM broadcasts!

Thus needing great care over power limitations

# Fibre Hybrid

BT are currently upgrading their network to use G.fast, but from the street cabinet, not the distribution point

We might get 100-330Mb/s at those premises sufficiently close to the cabinet

For others further away, it will be no better than standard FTTC

# Fibre

For the last mile fibre would be ideal, so FTTH/FTTB is also being offered in some places

With this you can get 330Mb/s from BT and Gb speeds from other vendors (Virgin are starting to roll out 1Gb)

At a price!

Exercise. The physical encodings that are used in fibre are much the same as in copper: read about these

# The Last Mile

In the UK we have:

| | | | |
|---|---|---|---|
| 5500 exchanges | ADSL | copper | 24Mb |
| 10,000s street cabinets | VDSL<br>FTTC | fibre +<br>copper | 80Mb |
| 1,000,000s distribution points | G.fast<br>FTTdp | fibre +<br>copper | 300Mb |
| 30,000,000 premises | Ethernet<br>FTTP | fibre | 1GB |

# Cable TV

The cable TV system, where available, is also used to deliver Internet connectivity

Newer installations are full fibre, but there is also a lot of another fibre/copper hybrid, with fibre to cabinets and then copper to the home

However, the copper wires used is good(ish) quality coaxial cable that is well screened against interference and crosstalk, and so the data rates it supports are much higher

# Cable TV

Telephone wire and coaxial cable
Picture from Virgin Media

Exercise. Read up on DOCSIS

# Wireless

The next physical medium we look at is wireless

Wireless networks have been around for a long time: for example cellular telephone systems

Everything wireless is overseen by national and international bodies: we can't have a free-for-all in a wide area shared resource

One wireless system can affect another hundreds or thousands of miles away: there must be some sort of cooperation

So some wireless systems are only allowed with very low power, e.g., Wi-Fi

# Wireless

Europe has the European Telecommunication Standards Institute (ETSI)

USA has the Federal Communication Commission (FCC)

Such bodies manage the radio spectrum, allocating various frequencies to various purposes, ensuring minimal interference between the competing concerns for parts of the spectrum

# Wi-Fi

The IEEE 802.11 group of standards deal with “wireless Ethernet”, more commonly known as Wi-Fi

In principle, much like CSMA/CD over wireless, but with some extra problems unique to wireless

The shared medium is now all around, not just within a wire

So signals from multiple networks can interfere; not just the hosts within one network

# Wi-Fi

Wireless networks generally have fairly high error rates due to interference from electrically noisy environments, signal reflections, etc.

So the bandwidth achievable is dependent on the circumstances of the environment

Conversely, wireless networks generate interference themselves which must be controlled so not to be to annoying to other people

# Wireless Problems

In 802.11, the allowed power of transmission is generally kept quite low by the standards bodies to minimise interference

E.g., a typical laptop will transmit at about 32mW; it can read a signal as low as 0.00000001mW

A typical digital TV mast transmits at 100kW

Thus the range achievable by Wi-Fi is often quite limited (deliberately)

And limited range can cause complications

# Wireless Problems

When we have wireless, we get the hidden host problem:

Hosts A can B can “see” each other; B and C can see each other, but A cannot see C, so A cannot tell if its packets to B are colliding with C's to B

# Wireless Problems

In reality, the ranges will not be circular, but something rather complicated dictated by the environment

But the limited ranges mean that CSMA/CD will not work for wireless

CSMA/CD relies on everyone's signals being visible to everybody

# Wireless Problems

Furthermore, as packets are broadcast, wireless networks are intrinsically insecure, so extra effort must be taken over security and authentication

War Driving is driving with your laptop around the neighbourhood until you find an unsecured wireless signal: then you have free access to the Internet!

This is illegal in the UK and elsewhere

These days, many fewer people forget to secure their networks than was common in the early days of Wi-Fi

Only use a Wi-Fi network if you have permission to do so

# Wireless 802.11

There are several parts to the 802.11 standard, including 802.11a, 802.11b, 802.11g, 802.11n, 802.11ac, 802.11ax and more

You may see them under the brandings:

| | |
|---|---|
| Wi-Fi 6 | 802.11ax |
| Wi-Fi 5 | 802.11ac |
| Wi-Fi 4 | 802.11n |
| Wi-Fi 3 | 802.11g |
| Wi-Fi 2 | 802.11a |
| Wi-Fi 1 | 802.11b |

Other parts of 802.11, like 11c, 11d, 11e, 11f, 11h, 11i deal with things like power management, quality of service, security and authentication and so on

# Wireless 802.11

The original standard specified signalling rates of up to 2Mb/s

Up to 100m (300 feet) indoors and 300m (1000 feet) outdoors

There was an infra-red mode as well as a radio mode, but this was not widely implemented

802.11b extended this to rates of 5.5Mb/s and 11Mb/s

# Wireless 802.11

They use the unlicensed 2.4GHz waveband

That means you do not need to get a licence to use that frequency at low power

This was a frequency that was otherwise unusable commercially and is subject to interference from microwave ovens and other things

And the frequency fell within the capabilities of low-power chips that were buildable at the time

# Wireless 802.11

802.11a: 54Mb/s, using the 5GHz waveband

802.11g: 54Mb/s, using the 2.4GHz waveband

802.11n: 150Mb/s, using either waveband

802.11ac: 867Mb/s, using the 5GHz waveband

802.11ad: 6.75Gb/s, using 2.4GHz, 5GHz and 60GHz wavebands (“WiGig”)

802.11ax: 14Gb/s using both the 2.4 GHz and 5 GHz wavebands and others if available (also reduces latency)

802.11ay: proposed 40Gb/s using the 60GHz waveband (high frequencies don't go through walls; positioned as an Ethernet replacement)

Improvements are achieved through more sophisticated encodings and using more wireless channels simultaneously

# Wireless 802.11

Each will fall back to previous standards to maintain compatability with earlier devices

For example, 60GHz will not go through walls, so 11ad falls back to 11ac if you move to the next room

Exercise. Look these up. Particularly the use of multiple aerials for beamforming and spacial multiplexing

# Wireless 802.11

802.11 hardware is branded “Wi-Fi”, which is actually a certificate of interopability given to manufacturers whose equipment demonstrably works with other manufacturers'

Administered by the Wi-Fi Alliance, a consortium of interested companies

# Wireless 802.11

The bits in 802.11 are not simply transmitted directly: there is a lot of environmental interference to overcome

Instead the signal is spread over many frequencies using variety of techniques collectively called spread spectrum

Exercise. Read about Direct Sequence Spread Spectrum (DSSS)

# Wireless 802.11

For Wi-Fi, the allocated frequency band (2.4GHz) is split into 14 overlapping 22MHz channels each centred on specified frequencies

The number of channels available depends on the country

| | |
|---|---|
| Most of Europe: | 13 |
| North America: | 11 |
| Japan: | 14 |

# Wireless 802.11

| Channel | GHz |
|---|---|
| 1 | 2.412 |
| 2 | 2.417 |
| 3 | 2.422 |
| 4 | 2.427 |
| 5 | 2.432 |
| 6 | 2.437 |
| 7 | 2.442 |
| 8 | 2.447 |
| 9 | 2.452 |
| 10 | 2.457 |
| 11 | 2.462 |
| 12 | 2.467 |
| 13 | 2.472 |
| 14 | 2.484 |

# Wireless 802.11

These channels are 5MHz apart, so neighbouring channels overlap (as they are 22MHz wide) and interfere. Therefore you need to take care which channels you use

There are recommendations on using channels

# Wireless 802.11

Separate channels by at least 2 (e.g., use 1 and 4) to reduce interference

Separate by 4 (e.g., use 1 and 6) to have no interference at all

This means we can have three non-interfering co-located networks on channels 1, 6 and 11

# Wireless 802.11

Separating networks physically gives more leeway:

- Separate by 1 (e.g., use 1 and 3) if the networks are more than 40m apart
- Adjacent channels (e.g., use 1 and 2) are OK over 100m
- Channels can be reused when the networks are sufficiently separated

# Wireless 802.11

More subtle channel allocations allow a little overlap (e.g., using channels 1 and 3) that have a little interference, but a greater overall aggregate bandwidth

Exercise. Mobile phones have wireless apps that display the wireless environment. Walk around and see what it is like

# Wireless 802.11

# CSMA/CA

802.11 uses carrier sense, multiple access, collision avoidance (CSMA/CA)

This is similar to CSMA/CD in Ethernet, but with a big difference

- Carrier sense: to deal with the common case of non-hidden hosts, first listen for a signal
- If free, send a packet
- If busy, wait until the end of the transmission and then enter a contention period: wait a random period
- Go back to carrier sense

# CSMA/CA

Waiting for the contention period is the collision avoidance

A random wait mean that several hosts wanting to transmit are unlikely to all start transmitting simultaneously

We are trying to avoid a collision in advance rather than detect one after the fact

But collision avoidance does not guarantee no collisions, particularly with hidden hosts, so we need more

# CSMA/CA

Thus, on successful receipt of a packet, a host will broadcast an acknowledgement (ACK) packet

This is just a packet to inform the sender that everything worked well and there was, in fact, no collision or other loss

If the sender never gets the ACK, it will resend, starting from the CA again

This ACK is important, as measurements have found typical loss rates on the order of 30%

Note the ACK is also visible to everyone in range of the destination, giving extra indication when a transmission has finished

# CSMA/CA

Why use collision avoidance rather than collision detection?

Clearly, the contention period means more latency in transmission

We do it because with wireless, collisions can be very hard to detect

With Ethernet, detecting another host's signal on a wire is easy as the power of its signal is roughly the same as yours

# CSMA/CA

In contrast, detecting another host's radio signal can be very difficult as it can be a tiny fraction of the power of yours, and your signal will drown out the colliding signal and make it undetectable

Recall the wide range of power that Wi-Fi signals encompass: another destination might be transmitting quite powerfully, but its signal can be very small by the time it reaches you

# Wi-Fi

To help with the visibility problem, there is optional RTS/CTS handshaking

# RTS/CTS

1. Before sending a data packet the source A can send a request to send (RTS) packet to B

# RTS/CTS

2. If the destination B is happy (it is not already receiving from another host that A cannot see) it responds with a clear to send (CTS) packet

# RTS/CTS

2. Every other host within the range of the destination will see the CTS and so know not to send themselves

# RTS/CTS

3. The RTS and CTS contain the length of the desired transmission so other hosts know how long they will have to wait

# RTS/CTS

4. Similarly, the final ACK is visible to everyone

# RTS/CTS

5. Then C can start with its own RTS

# RTS/CTS

This means there is even more latency overhead before data starts to be transmitted, so RTS/CTS can be switched off or on as required:

RTS/CTS always on: good for large or busy networks

RTS/CTS never on: good for small or lightly loaded networks where every host can see all other hosts

RTS/CTS for large packets only: a compromise that reduces the relatively large overhead for small packets

# Wireless Rates

Although 802.11b is nominally 11Mb/s and 802.11g is nominally 54Mb/s remember these are the signalling rates, not the data rates

The signalling rate is the raw bit rate over the airwaves: a lot of that is consumed in overheads

Realistically, 802.11b gives about 3 to 4Mb/s and 802.11g about 20Mb/s

Some of the later 802.11 standard improve speeds by reducing overheads (as well as using better encodings)

# 802.11

Exercise. 802.11ac (branded “Wi-Fi 5”) is common and 11ax (“Wi-Fi 6”) hardware has just started to appear. Read up on what they promise and what they deliver

# Wireless Networks

While the use of access points is common, this is not the only way to set up a wireless network

802.11 can be arranged in point-to-point networks called Ad-Hoc or Independent Basic Service Set (IBSS)

# Wireless Networks

Each host communicates directly with each other without an access point

# Wireless Networks

More common is Infrastructure or Basic Service Set (BSS), where a central hub (access point) relays traffic between hosts

This is more expensive to set up (as you have to buy an AP), but covers a larger area

Also the AP can connect into a wired network and so the rest of the Internet

# Wireless Networks

Extended Service Set (ESS) connects several APs by a wired network

This allows hosts to roam and they can be configured to handoff automatically between APs if the required authentication infrastructure is set up in the APs

An ESS can cover an area as large as you like

# Wireless Security

Wireless packets are readable by anybody in the neighbourhood, so security is essential in a wireless network

Original 802.11 employed the Wired Equivalent Privacy (WEP) encryption scheme

Both ends of a communication share a secret key that is used to encrypt the traffic between them

WEP is now easily breakable: after collecting a modest amount of traffic the system can be broken

As can its successor, Wi-Fi Protected Access (WPA)

# Wireless Security

Currently we use WPA2, (IEEE 802.11i-2004)

Exercise. Read about the break of the WPA2 protocol (Oct 2017)

Exercise. Read about the new WPA3

# Wireless Security

Two major ways to set up authentication are

- WPA-Personal: also called WPA-PSK (pre-shared key), where an access point has a secret key, and a host authenticates directly with the AP using the secret key
- WPA-Enterprise (802.11X): requires a separate authentication server (typically a RADIUS server) that the AP will contact. Much more fiddly to manage, but allows roaming across an ESS. Also roaming across institutions using hierarchical RADIUS servers

We usually find BSS using WPA-PSK and ESS using WPA-Enterprise, but either can use either

# Wireless Security

For WPA-PSK the secret key is usually derived from a password for ease of use

The password is communicated off-line, e.g., written down somewhere

Everybody on the network shares the same key/password; authentication is done in the AP

# Wireless Security

RADIUS authentication

# Wireless Security

Multi-institution

# Wireless Security

For WPA-Enterprise each user has their own key/password

Authentication is done in the server on both the username and the password

# Wireless Security

Exercise. Read about how Eduroam uses WPA-Enterprise

Exercise. Read about RADIUS: Remote Authentication Dial In User Service

# Wireless Security

Some APs have Wi-Fi Protected Setup (WPS), a simplified way of setting up WPA/WPA2 security

Designed for those people who find typing in a password too challenging

It is seriously broken and should be disabled on your AP

Exercise. A common system we see on public Wi-Fi is a redirect to a login web page: sometimes called a captive portal. What kind of security (privacy and authentication) does this provide? Note this is not WPA-Enterprise

# Wireless 802.11

The frame layout for Wi-Fi is the same as Ethernet

In particular it has the same format MAC addresses, e.g., 00:04:ed:f1:ef:8a

This allows the transparent mixing of Wi-Fi and Ethernet in a single network

An AP can pass on a Wi-Fi frame unchanged to an Ethernet; and vice versa

Exercise. What implication does this have for Ethernet collision domains?

# PHY Sublayers

This is a good argument for sub-dividing the physical layer!

Exercise for hardware hackers: read about the IEEE layers:

Physical Medium Attachment (PMA) for things like frames
Physical Coding Sublayer (PCS) for things like 4B/5B
Physical Medium Dependent (PMD) for the hardware

# Other Wireless

Many other wireless networks exist, from local to wide-area

# Other Wireless

Bluetooth gives short range, point-to-point communication

Point-to-point: just two hosts in the network

A range of 10m

Also uses 2.4GHz band

Not really designed to run IP, but can by layering a suitable protocol (see PPP, later)

Bluetooth Low Energy (BLE), is a non-backwards-compatible evolution designed to reduce power consumption

# Other Wireless

Exercise. Read about Adaptive Network Topology (ANT and ANT+) for short range low power wireless, similar to BLE, but for use with fitness (and other) sensors (by Garmin)

Exercise. Read about ZigBee for short range low data rate, low power wireless, for use in home automation and control

# Other Wireless

And more:

IEEE 802.16 Wireless Man standard, also known as WiMax for a Metropolitan Area Wireless (MAN)

802.22 Wireless Regional Networks

802.16e Mobile Wireless MAN

802.20 Mobile Broadband Wireless Access

HiperLan and HiperLan II

802.1s Mesh Networking

And so on

# Other Wireless

We should also mention cellular networks, as used by mobile phones

The first important digital system was Global System for Mobile Communications (GSM)

Retrospectively named 2G

(1G was the preceding analogue system)

Rates of 9.6Kb/s to 14.4Kb/s (similar to old analogue wired modems)

High Speed Circuit Switched Data (HSCSD) takes this up to 57.6Kb/s

# Other Wireless

General Packet Radio Service (GPRS), packet based, up to 171.2Kb/s

Uses several GSM channels

Enhanced Data rates for GSM Evolution (EDGE) uses better encodings to get up to 384Kb/s, again using several channels

EDGE used by Third Generation (3G) systems

High-Speed Downlink Packet Access (HSDPA) ups this to 42.2Mb/s

Evolved High-Speed Packet Access (HSPA+) will do 168Mb/s

# Other Wireless

4G is well established

Using Long Term Evolution (LTE) with the promise of 300Mb/s

Marketed as “4G”, it originally did not meet the proposed 4G standard as it did not satisfy the proposed technical specifications of a 4G system

In particular, a 4G network should support 1Gb/s (for a stationary host)

The ITU (who say what “4G” is supposed to mean) actually gave in to commerce and retroactively changed the definition of 4G to allow for LTE

# Other Wireless

LTE is data traffic only, and does not have a voice channel

Currently on most LTE systems if you want to make a voice call it has to drop back to 3G (or even 2G)

Just being introduced is voice over LTE (VoLTE) using a suitable digital encoding of sound

# Other Wireless

5G is on track for about 2020 for widespread deployment

It uses the available spectrum much more efficiently than 4G, and employs frequencies up to 86GHz (LTE uses up to 6GHz)

Projections indicate users connected to a base-station will share 20Gb/s download and 10Gb/s upload rates

And base-stations will support “millions” of devices per square mile (enabling the Internet of Things)

A device will be able to connect even if it is moving at 500km/h (e.g., in a plane); latencies will be 1ms, compared to the current 20ms on LTE

# Other Wireless

Current sticking points over the adoption of 5G are:

5G chipsets currently suck a lot of power

the need to build a lot more base stations

fighting amongst the phone companies over the radio spectrum

# Other Wireless

6G? A new “G” appears roughly every 10 years, so maybe 2030

With targets of 100Gb/s using 100GHz to 1THz (terahertz) frequencies

(So the base-stations will need really good connectivity!)

# Other Wireless

Satellite networks can be used outside of well-connected urban areas

There are two main variants

# Other Wireless

One way satellite: this employs the usual asymmetry. Data away from the home travels by telephone wire; data towards the home travels through a satellite connection

# Other Wireless

Two way satellite: satellite connections both ways. More expensive in equipment in the home, but not reliant on a telephone network

# Other Wireless

Satellites are very expensive to put up and to run

The signal has a large latency: about 1/10 sec, which can be very noticeable in highly interactive applications (games)

They cover a large area with a reasonably good bandwidth

They are good for remote and undeveloped areas with no other local infrastructure

# The Physical Layer

We have seen some implementations of the physical layer

There are very many more

There are many implementations as there are many physical requirements of networks (distance, speed, power, etc.)

Fortunately, as we go up the layers, the amount of variety decreases!

# Link Layer Protocols

We now turn to some other link layer protocols

Serial Line Internet Protocol (SLIP) is an early protocol used on modems to encapsulate IP traffic over serial (telephone) lines

It is a point-to-point protocol, meaning it links just two machines to each other: the normal requirement in early dial-up systems

# SLIP

A very simple frame encapsulation with a terminating byte (hex) c0; also often a starting c0 byte, too

# SLIP

So how to send data that contains the byte c0?

Use byte stuffing

To send c0 actually send two bytes db dc

The pair db dc is reconstructed as c0 at the other end

Stuff db as the pair db dd, which the other end reconstructs as db

A minor expansion of data, but it enables transparent transmission of data

# SLIP

There is no frame size limit, but it is suggested you should support at least 1006 bytes

296 bytes was common (40 bytes of TCP/IP headers plus 256 bytes of data)

Larger frames have relatively less overhead, but at 9600b/s (a typical early modem speed) 1006 bytes takes about 1 sec to transmit

If there is a bulk transfer of full-sized frames at the same time as an interactive session, the session's frames would have to wait 0.5 sec on average to get through, much too slow

An interactive response of over 100-200ms is felt to be slow

# SLIP

296 is a compromise: not too slow for interactive, not too small for bulk transfer, but not particularly good for either

On more modern modems (56Kb/s) it was increased to 1500 bytes

# SLIP

SLIP has several problems

Only IP in the next layer is supported (no type field in frame)

The ends must have pre-agreed IP addresses: no mechanism for agreeing addresses

No checksum: telephone lines were noisy and created data corruption

No authentication: no way to check who is connecting

# PPP

Thus the Point-to-Point Protocol (PPP) was developed

Like SLIP it is a point to point protocol

It has three parts

- A framing layout for packets
- A link control protocol (LCP) for managing and configuring links
- A set of network control protocols (NCP) to manage network layer specific options

# PPP

Frame delimiters 7E, start and end

Address (always ff), Control (always 03)

Protocol: tells us what the next layer is, e.g., IP is 0021

Cyclic redundancy check to spot corruption

But no address fields

# PPP

Up to 1500 bytes of data (but can be negotiated)

Values are escaped (like SLIP) by 7d

7e ! 7d 5e

7d ! 7d 5d

x, where x < $20_{16}$ ! 7d [x + 20], so 01 ! 7d 21

# PPP

NCPs can negotiate extras, like compression, frame size, etc.

And authentication, e.g., passwords

While it was devised to be used over telephone modems, PPP is still actively used, e.g., in PPP over Ethernet (PPPoE) as it allows authentication of a connection

Current FTTC products use PPPoE over VDSL to pass authentication to the ISP

# Link Layers

Several other link layers exist

We have already seen the Ethernet frame for a local area network

There are many link layers for carrying data over long distances, at high data rates

# Link Layers

For example, Asynchronous Transfer Mode (ATM) was popular for a while

Designed by telephone engineers, it was really a connection oriented digital voice network into which you could squeeze data packets

Exercise. Read about ATM and PPPoA, that layers (IP over) PPP over ATM, as used in ADSL and DOCSIS

# Link Layers

Multiprotocol Label Switching (MPLS) was designed post-ATM when the technology decisions that drove the design of ATM were deemed no longer applicable

Designed by network engineers to be a general long-distance network, it is much better suited to modern data networks

Exercise. Read about MPLS and how BT uses it in its 21C Network

# Networks

## Ethernet Link Layer

We want to move up to the network layer: but before doing so there is one more remark on the link layer

Recall Ethernet. The data on the wire:

# Networks

## Ethernet Link Layer

The interesting fields here are the addresses

The addresses allow a frame to get from source to intended destination

This works well enough when the destination is on the local Ethernet network

Which is shared (or switched), so the frame has no problem being seen by the destination host

# Networks

## Ethernet Link Layer

What to do when the destination is non-local?

We can't simply treat the world as a shared medium and broadcast the packet to everybody

And the network at the destination might not even be an Ethernet and not have an Ethernet address

So we need hardware independent addresses to identify hosts that work independently of the physical network

In the Internet Protocol, these addresses live in the network layer

# Networks Link Layer

## IP

The network layer used in the Internet Protocol is called the Internet Protocol (IP)

It has the major function of dealing with routing, determining where a packet should go

Amongst a lot of other stuff, the IP header has network layer addresses

These are hardware independent, and in the same format across the entire Internet

# Networks

## IP

Each host on the Internet has an IP address that identifies it uniquely over the entire Internet

At least, that was the original intention

This is certainly no longer true, for reasons we shall explore later

But, for now, assume this is true

# Networks

## IP Header

A bit hard to see, so conventionally we stack the header vertically

# Networks

## IP Header

# Networks

## IP

The source and destination addresses are both four bytes long: we shall come back to the other fields later

10001010001001100010000000001110 is an example IP address, a 32-bit value

This is 2317754382 in decimal: not terribly easy to work with

So for convenience we write this as 138.38.32.14 , decimal representations of four 8-bit values. The dots are purely to make the number visually easier to read

But, importantly, there is structure in an IP address which helps with routing

# Networks

## IP

In this example, 138.38.32.14 . the first half 138.38 is a 16-bit network address, which identifies the University of Bath

And 32.14 is a 16-bit host address, which identifies a single machine on the University's network

Note that we write 138.38, but this should really be thought of as 1000101000100110, a bunch of 16 bits

Always remember that the dotted decimal notation is just a convenient way of writing a chunk of bits: there are no decimal numbers in the header!

# Networks

## IP

This division into network and host parts helps immensely in routing, as all packets destined for the University of Bath can be routed in the same manner

Only when a packet reaches the University is some local knowledge of the network needed

Indeed, the host part of this address splits further into subnet addresses that help local routing within the University

But the main point for now is that this IP address is independent of Ethernet and so can be used regardless of the hardware used

# Networks

## IP

But, now, there is an new problem

Suppose I want to send a packet to 138.38.32.14 on the local network. My data is (ultimately) encapsulated in an IP packet, with my IP address as source and 138.38.32.14 as the destination

(The question of how do we know the destination IP address must be answered later)

Now the IP packet must be further encapsulated in a hardware frame, Ethernet in this example. The OS can't send the packet on the physical medium until it knows the Ethernet address of the destination

# Networks

## IP

Ethernet does not know about IP addresses

IP does not know about Ethernet addresses

And this separation of layers, as we know, is desirable

# Networks

## IP

We need some kind of address discovery, so given the IP address we can find the corresponding Ethernet address

This is done by the Address Resolution Protocol (ARP)

ARP is a very simple link-layer protocol that essentially broadcasts a special frame on the local medium to the effect of “who has IP address 138.38.32.14 ?”

All hosts on the local network hear this broadcast and the host with that address replies “Me: and I have Ethernet address 08:00:20:9a:34:dd ”

(Another security problem. . . )

# Networks

## IP

The OS gets the ARP reply and can now use this information to write the correct address in the Ethernet frame

Only now can the original packet be sent

# Networks

## IP

We don't want to use ARP for every packet we send, so there is an ARP cache kept by the OS kernel that records the relation 138.38.32.14 $ 08:00:20:9a:34:dd

Entries in the cache time out and are removed after, say, 20 minutes

This is in case the host using 138.38.32.14 goes away and is replaced by a different host with the same IP address, but a different Ethernet address: recall IP addresses are not associated with the hardware

Once expired, the next packet to 138.38.32.14 will need a fresh ARP

# Networks

## IP Routing

A quick note regarding when the destination is not on the local network

IP routing for the source host is quite simple: if the destination is on the local network, send the packet directly. This probably uses ARP (on the first packet) to get the hardware address of the destination

# Networks

## IP Routing

If the destination is not on the local network, to solution is to send the packet to a gateway host and let it deal with where to send it next

A gateway is just a machine on more than one network

This keeps the complexity of the software needed on the hosts down: only the gateway will need to have a bit of intelligence about routing

# Networks

## IP Routing

The only information a source host needs to know to do routing is:

its own address

the address of a gateway machine

We shall see later how it gets this information

# Networks

## IP Routing

So, for a host the routing software is:

> is the destination on the local network?
>
> yes: send it directly, possibly with an ARP, if needed
>
> no: send it to the gateway, possibly with an ARP, if needed

Note in the latter case, the host might need to do an ARP for the gateway

# Networks

## IP Routing

In the non-local case, the packet is going to the gateway, so we would need to ARP for the hardware address of the gateway

The packet, with IP address of the final destination, is put into a frame with Ethernet address of the gateway

Since the packet needs to go to the gateway

So, here, the physical and network addresses in the Ethernet frame are completely unrelated!

# Networks

## IP Routing

This is another reason why we need both hardware and software addresses

The IP address is for the ultimate destination; the hardware address is for the next hop

ARP is not restricted to Ethernet and IP, but can be used to pair any physical and network layer addresses

# Networks

## IP Routing

Exercise. Is ARP needed on a PPP connection?

# ARP

ARP is a simple protocol

On an Ethernet, it has to use an Ethernet frame, so what destination address does it put in the frame?

It broadcasts an ARP Request packet (protocol number 0806) in an Ethernet frame with destination hardware address ff:ff:ff:ff:ff:ff and source its own Ethernet address

All hosts on the local network read the frame

The target host recognises the request for its IP address

# ARP

The target sends an ARP Reply packet containing its own Ethernet address (the other hosts need do nothing)

It knows the source's Ethernet address as read from the request packet

The source gets the reply and reads out the target's Ethernet address. It can now use that Ethernet address to send IP packets

# ARP

# ARP

Contained within an Ethernet frame, of course

The frame type for ARP is 0806

The Ethernet frame type field allows the software that reads the packet from the Ethernet card to pass the contents of the packet to the software that implements ARP

# ARP

Hardware type: 1 for an Ethernet address

Protocol type: 0800 for an IP address

Sizes: sizes in bytes of the address fields, 6 for Ethernet, 4 for IP

# ARP

OP: 1 for a request, 2 for a reply

Address fields: the data

In a request the destination hardware field is not filled in as this is what we are trying to find!

In a reply the sender Ethernet address is the address we seek

# ARP

If no machine on the local network has the requested IP address, or that machine is down, no reply will be forthcoming

In this case, after a few seconds, and a few repeated ARP requests, the OS returns an error message to the application trying to make the IP connection

This might be “no such host” or “host unreachable”

# ARP

It is sometimes useful to give an ARP reply even if nobody has asked for it. For example a new machine joins the network or an existing machine changes its IP address for some reason

This is a gratuitous ARP

All machines on the local network are free to read any ARP request or reply they see and modify their own ARP caches accordingly

# ARP

So a gratuitous ARP would help break old associations that are no longer valid but still cached

Without a gratuitous ARP a host might send an IP packet to the old cached, but now wrong hardware address

# ARP

ARP is purely a local network thing: discover a hardware (next hop) address on the local network

And it makes no sense for an ARP to be forwarded to another network, which might not even be of the same physical type

But there is a interesting trick that shows ARP can be used for things other than it was designed to do

Used in the days before switches were common: unlikely to be used these days

# ARP

This trick allows us to extend an Ethernet (or other network) over a physically larger distance than its specifications allow, and to join a wireless network to a wired one so they appear to be a single network

A bridge is a host that joins two physical networks into one. It has two interfaces, one on each network

# ARP

## ARP Bridging

This examples joins a Wi-Fi to an Ethernet, but we could have any two networks that share a MAC address type

If host h1 wishes to send to host h2 it must determine its hardware address (as it is on the “same” local network)

So h1 does an ARP broadcast for h2

The bridge sees this request and responds on behalf of h2 (a proxy ARP), but it supplies its own hardware address b1

# ARP

## ARP Bridging

Now h1 sends data to what it thinks is h2, but is actually the bridge

The bridge reads the packet, sees it is destined for h2 (by its IP address) and forwards it to the other network where h2 can read it

Furthermore, it rewrites the forwarded frame's header to have h2 as destination and b2 as source

If h2 replies, it can either use h2 which it got from the original packet or do an ARP request, which the bridge proxies in a symmetrical way

# ARP

## ARP Bridging

In either case the packet goes to the bridge, which forwards it to h1, again rewriting the frame addresses appropriately

This is all transparent to h1 and h2 who believe they are on the same network

If h1 is communicating with both h2 and h3 its cache will show them to have the same hardware address b1: this is not a problem

# ARP

## ARP Bridging

Exercise. Find out if your home network does ARP bridging, or if it simply acts like a switch on a single network

Exercise. Make sure you understand the difference between what a gateway does, what a switch does and what a bridge does

# Bridging

## Bridging

While we are talking about bridges:

ARP bridging is fine for joining a pair of small networks, but less so for larger collections of networks

The IEEE 802.1d Ethernet Bridging standard addresses this, dealing with the cases of multiple routes between hosts

# Virtual Bridging

And a common variety is 802.1q virtual bridging

More commonly called Virtual LANs (VLANs)

This is a kind of reverse of the ARP bridge: it allows more than one network to run on a single physical network

# Virtual Bridging

A company has two separate sites 1 and 2 with a single dedicated link between them

# Virtual Bridging

They want to run two separate LANs, A and B, but not to buy a second link between the sites

# Virtual Bridging

They can use 802.1q tagging

# Virtual Bridging

A packet from LAN A in Site 1, say, arrives at the switch

# Virtual Bridging

The switch knows to route the packet over the remote link: it places a 802.1q tag on the frame

# Virtual Bridging

A tag is an extra four byte header containing a Virtual LAN Identifier (VID), a 12 bit integer

# Virtual Bridging

The frame type in the physical layer (typically Ethernet) is changed from 0800 to 8100 to indicate a tagged packet

# Virtual Bridging

The switch in Site 2 receives the packet, sees the tag, reads and removes it and forwards the packet to its part of LAN A

# Virtual Bridging

This generalises well to many virtual LANs and allows many networks to share infrastructure, thus saving on cost

Note: this is quite different from Virtual Private Networks (VPNs), which we shall talk about later

Exercise. Look up the structure of a VLAN tag

Exercise. The University uses VLANs extensively. Find out about this

Exercise. How does tagging interact with maximum frame sizes, e.g., in Ethernet?

# Virtual Bridging

Bridging is useful, but shouldn't be taken too far

Larger networks have more traffic

Just think of the ARP broadcasts alone!

It is often better to split a large network into several smaller ones: see subnetting, later

# RARP

Exercise. Read about Reverse ARP (RARP): given a hardware address find the IP address

# Network Layer

We have briefly seen the Network Layer in the IP to consider its addresses

We now need to look at IP in more detail

It is the basis the Internet is built upon

It is actually quite simple, but allows more complex stuff to be layered on top

We shall start by describing IP version 4, IPv4

And talk about IPv6 later

# IP

IP is a best-effort, connectionless, unreliable, packet based protocol

Recall: “unreliable” means “not guaranteed reliable”

It represents the lowest common denominator of network properties

IP doesn't rely on any particular property of a link layer, so it can run on top of almost any link layer, even unreliable ones

# IP

IP is a cooperative system: for a packet to get from source to destination it is handed from one network to the next, hop by hop

No single machine anywhere has any idea what the entirety of the Internet looks like

# IP

The nodes in the network have various roles:

Host. A machine you actually use to do some work

Bridge. Connects two physical networks together

Gateway. Provides a connection off the local network

Router. A machine joining two or more networks and whose primary function is to determine where a packet goes next

These are not mutually exclusive: gateways and routers can be hosts; gateways do trivial routing

# IP

Marketing alert: things you see described as “routers” in the shops are unlikely to be actual routers, which are specialist bits of equipment

The word “router” seems to means “a box you plug into the network”

# IP

The basic idea is that a packet does not know how to get from source to destination: this is the routers' job (and it can be quite complex: see later)

The IP layer takes bytes from the transport layer and prepends a header, producing packets often called datagrams in this layer

The IP specification says datagrams can be up to 64KB in size, but they are usually in the region of 1500 bytes (Ethernet, again)

We return to the IP header

# IP

IPv4 datagram header

# IP

Version. Four bit field containing the value 4. A later version of IP (IPv6) contains 6

Header length. There are some optional fields, so the header can vary in size, so this is needed to distinguish the end of the header. Given as a number of 4 byte words. Four bits, maximum value 15, so maximum header length of 60 bytes

Type of service. Eight bits. To indicate to a router how this datagram should be treated in terms of cost, speed and reliability (if possible)

E.g., for audio it is better to get data through quickly rather than 100% reliably as the human ear is more sensitive to gaps than occasional errors

# IP

## TOS/DS

The TOS field, these days called the Differentiated Services Field (DS field), is to inform routers on the best way to treat this datagram

This allows the implementation of Quality of Service (QoS)

The full range of options available is complex (see RFC2474 for details), but can indicate things like

> Minimise delay. Do not hold onto this datagram longer than necessary, and perhaps prioritise it over others

# IP

## TOS/DS

Maximise throughput. Not quite the same as minimising delay, since collecting together several small datagrams and sending them off together may be more bandwidth efficient

Maximise reliability. Try not to drop this datagram if the router is becoming overloaded; drop another datagram first

Minimise cost. For this datagram cost is more important than reliability or speed. This datagram can be delayed if it makes transmission cheaper

# IP

## TOS/DS

Early routers ignored the TOS field, but these days QoS is very important

Modern routers do (or should) pay attention to the DS field

Here, as in other parts of the IP specification, a router may ignore some information if it wishes. It might be the software is so old it does not recognise a modern field; or it might simply be unable to make use of the information. Hosts are strongly recommended to act on the information, though

Exercise. Look up the problems Explicit Congestion Notification (ECN) had when it was introduced

# IP

Total Length. Of the entire datagram, including header, in bytes. 16 bits, so giving a maximum size of 65535 bytes. Much larger than domestic networks need, but too small for high-speed networks.

As usual, larger packet sizes mean lower overheads:

Time overhead in hosts of splitting data into datagrams, adding headers, then removing headers and reassembling

Bandwidth overhead as each header is 20 or more bytes that is not data

Time overhead in routers of processing packets

# IP

Identification. 16 bits. A value that is unique to each (source) datagram, often incrementing by 1 for each successive datagram sent

Used in fragmentation to reassemble the fragments of a single datagram. All the fragments get their own IP header, but share the same identification

So we need to discuss fragmentation

# IP

## Fragmentation

The path a packet takes from source to destination will typically go through a wide variety of differing kinds of hardware

# IP

## Fragmentation

Thus IP must face the problem of differing link layer properties, in particular maximum packet size

Exercise. Reflect on why this hardware issue can't really be dealt with in the hardware layer

If a big datagram hits a part of the Internet that only allows small datagrams, there is a problem

IPv4 deals with this by fragmentation: a datagram can be subdivided by a router into several smaller datagrams; it is the the destination's problem to glue them back together

In the right order

The fragmentation fields in the IP header deal with this

# IP

## Fragmentation

Flags. Three bits: two used and one reserved

1. RF. Reserved for later use, must be 0 (see RFC3514 for a suggested use)
2. DF. Don't fragment. If a host can't (or doesn't want to) deal with fragments this bit is set to inform the routers on the path to the destination. A router might choose an alternative non-fragmenting route, or simply drop the datagram and send an error message back to the source which can then send smaller datagrams
   All hosts are required to be able to accept datagrams of 576 bytes
3. MF. More fragments. All fragments except the last have this set

# IP

## Fragmentation

Fragment Offset. Where this fragment came from in the original datagram

# IP

Fragment Offset. 13 bits, giving the offset divided by 8. E.g., value of 20 means an offset of 160

So 13 bits is enough to cover the 16 bit range of sizes

And every fragment (apart from the last) must be a multiple of 8 bytes long: the router doing the fragmentation must ensure this

# IP

## Fragmentation

Fragment Offset. Every fragment has a copy of the original IP header, but with the various fragmentation and length fields set appropriately

In more detail: each fragment header will be a copy of the original header apart from

- total length: set to the fragment size
- MF: set to 1 if this is not the end fragment
- fragment offset: set appropriately
- (TTL and checksum: set appropriately)

# IP

## Fragmentation

In particular, all the fragments of the original datagram have the same the identification field value

When the fragment with MF = 0 is received, its fragment offset and length will give the length of the original datagram

The destination can then reassemble the original datagram when all the fragments have arrived

As always, they can arrive in any order; or not at all

# IP

## Fragmentation

IPv4 spends a lot of effort coping with fragmentation

It is costly and should be avoided

- Performing fragmentation in a router takes time
- More overhead as more datagrams for a given amount of data
- More overhead as more datagrams are traversing the network
- More datagrams means a greater probability one will be lost or corrupted

# IP

## Fragmentation

If a fragment is lost, the entire original datagram must be retransmitted: there is no mechanism in IP to indicate which fragment was lost

Fragments are datagrams in their own right and can themselves be fragmented

Fragment processing software (particularly reassembly) has a history of buggy implementations leading to hacked machines

# IP

## Fragmentation

Setting DF in the header prohibits fragmentation; if a router cannot avoid fragmenting it drops the datagram and returns a “fragmentation needed but DF set” error message back. The sender can then send smaller datagrams

DF allows MTU Discovery. The Maximum Transmission Unit (MTU) is the largest datagram a host or network can transmit. The path MTU is the smallest MTU for the entire path from source to destination

A datagram not larger than the path MTU will not get fragmented

MTU Discovery works by sending variously sized datagrams with DF set, and monitors the errors returned

# IP

## Fragmentation

When a datagram reaches the destination with no fragmentation error we have found a lower bound for the path MTU

This bound is approximate as the network is dynamic and paths may change!

This is the approach IPv6 adopts: don't have fragmentation in routers, but require MTU discovery

# IP

## Fragmentation

In IPv6 a datagram is never fragmented, but a router will always just drop a too large datagram and return an error message

MTU discovery is a required behaviour in IPv6, optional in IPv4

This is simpler, and so faster. Also, the IPv6 header is greatly simplified as it has no fragmentation fields

# IP

Back to the IPv4 header fields

> Time To Live. An eight bit counter used to limit the lifetime of a datagram

Poorly configured routers might bounce datagrams back and forth or in circles indefinitely, thus clogging the network with lost datagrams

The TTL starts at 64, or 32, say, and is reduced by one as it passes through each router

# IP

If a TTL ever reaches 0, that datagram is discarded, and an error message (“time exceeded in-transit”) is sent back to the source

This limits errant datagrams: eventually the TTL must reach 0 and the datagram is dropped

Eight bits means a maximum path of length 255, but this seems enough for the current Internet: no valid paths as long as this are known

The width of the Internet is the length of the longest path: this is uncertain and constantly changing but definitely over 32

# IP

Originally the TTL was to be a measure of time, reducing by one for each second in a router. In practice no implementations did this, but just decremented by one regardless. This is now the expected behaviour

Again: this is IP being pragmatic, following what people actually do in implementations, rather than the letter of the specification

Exercise. Why doesn't everyone simply put 255 into the TTL field?

# IP

Protocol. This eight bit field connects the IP layer to the transport layer. This is a value indicating which transport layer to pass the datagram to. For example, UDP is 17 and TCP is 6

Header checksum. As for the Ethernet header, this is a simple function of the bytes in the IP header. If the checksum is bad, the datagram is silently dropped. A higher layer must detect this and perform whatever action it needs. Recall that the IP layer is not guaranteed reliable

The checksum includes the TTL field so it must be recomputed and rewritten in the datagram by each router the datagram passes through

# IP

> Source and Destination Address. 32 bit numbers that uniquely determine the source and destination machines on the Internet

“Uniquely” is now not true

32 bits limits us to at most 4,294,967,296 hosts on the Internet

Not enough, and a significant chunk of those addresses are reserved for special purposes and can't be used for host addresses

# IP

The optional header fields allow for items that are either

- not common, so you don't want to pay the overhead of always having them, or
- extensions that are useful but were not included in the original IP specification

When IP was first devised, networks ran over telephone modems at a few thousand bits/sec. Now we expect giga and terabits. The physical layer has changed immensely while basic TCP/IP is pretty much unchanged!

A flexible approach across the entire TCP/IP suite such as allowing for options is part of this

# IP

IP layer options are not much used. Options include

Security

Record Route. Each router records its address in the option header as the datagram passes by

Timestamp. Each router records its address and the current time in the header as the datagram passes by

Strict Source Routing. A list of addresses that give the entire path from source to destination

Loose Source Routing. A list of addresses that must be included in the path from source to destination

# IP

Most IP options are for debugging or profiling behaviour. Mobile IP uses Source Routing (see later, perhaps)

The IP header length field has a maximum value of 60 bytes; the fixed part of the header is 20 bytes; thus options can take up to 40 bytes

This severely restricts what we can do in the options

For example recording paths: at 4 bytes of IP address per hop we can only record 9 addresses (when taking the length of the RR option into account)

Many real paths are over 30 hops

We use other techniques to map paths (see traceroute later)

# IP Addresses

We now go back and look at the IP addresses in more detail

Roughly (and incorrectly) speaking, every machine on the Internet has a unique address

These are not random, but allocated in such a way to make routing between hosts much easier

If there were no structure on the addresses every router everywhere would have to know where every host in the world was

Impossible, for technical, political and security reasons

# IP Addresses

Recall the Internet is a collection of networks

An IP address is split into two parts:

- A network number
- A host number on that network

The host number defines the host uniquely on a network

The network number defines a network uniquely on the Internet

# IP Addresses

As we have already seen, to an end host routing is trivial

- If the destination is on the same network, simply put the packet out on the network
- If not, send the packet to a gateway, and let it deal with the problem

It can tell if the destination is on the same network as itself by comparing the network part of their addresses

If they are the same, they are on the same network!

# IP Addresses

To a gateway or router the problem is to send the packet on towards the destination network

Which one? This is the difficult bit

But there are very many fewer networks than hosts, so this is already a great simplification

A router contains a table of IP addresses, with next-hop routers associated with those addresses

(Actually a more complicated datastructure than a simple table, but we can think of it as a table)

Containing network addresses, but individual host addresses are possible, too

# IP Addresses

Each row in the table contains

A destination address. This can be the address of a single host, but is usually a network address

The address of the next hop router, i.e., the address of where to send the packet next. This is the address of a router that is directly connected to the current one

Which interface to send the packet out on to get to that router. A router has many interfaces and this describes which one to use

# IP Addresses

When a packet arrives at a router it checks the table

- If the packet destination matches a host address in the table, send the packet to the indicated host on the indicated interface
- Else if the packet destination's network part matches a network address in the table, send the packet to the indicated router on the indicated interface
- Else find an entry in the table marked “default”, and send the packet to the indicated router on the indicated interface
- Else error

# IP Addresses

If there is no default entry, drop the packet and return an error message “network unreachable” to the source

For now, we can regard routers as machines with tables that tell them where to send packets. We will see how the tables are created later

End hosts have routing tables, too: they are very simple, just encoding the local/non-local decision

# IP Addresses

```
Destination      Gateway          Genmask          Flags  Metric  Ref  Use  Iface
138.38.96.0       *               255.255.248.0   U       0       0     0  eth0
127.0.0.0         *               255.0.0.0       U       0       0     0  lo
default           138.38.96.254   0.0.0.0         UG      0       0     0  eth0
```

A simple routing table as might be found in an end host on network 138.38.96

Send local traffic directly to the destination out on interface eth0

Otherwise send to the default gateway 138.38.96.254 , also on interface eth0

# IP Addresses

The netmask (Genmask in the example) tells us how to divide an IP address into network and host parts. More details later, but a 1 bit set in the mask indicates this bit is part of the network address

Work down the table ANDing the destination address on a packet with each netmask in turn. If the result equals the Destination value, we use this row to route the packet

A mask of 255.255.248.0 is 1111111111111111111110000000, so for this example the network part is the top 21 bits of the IP address

“default” is actually destination 0.0.0.0, and so always matches any address after an AND with mask 0.0.0.0

# IP Addresses

There is also a loopback address 127.0.0.1 for a virtual internal network connecting the machine to itself on (virtual) interface lo0

This is useful for many things, such as testing

# IP Addresses

A table from a machine with more than one real interface:

```
Destination      Gateway          Genmask          Flags  Metric  Ref   Use  Iface
213.121.147.69   *                255.255.255.255  UH     0       0     0    ppp0
172.18.0.0       *                255.255.0.0      U      0       0     0    eth0
172.17.0.0       *                255.255.0.0      U      0       0     0    eth1
127.0.0.0        *                255.0.0.0        U      0       0     0    lo
default          213.121.147.69   0.0.0.0          UG     0       0     0    ppp0
```

There are three interfaces: eth0 , eth1 and ppp0 (as well as lo )

A packet with address 213.121.147.69 goes directly out on interface ppp0

# IP Addresses

A table from a machine with more than one real interface:

```
Destination      Gateway          Genmask          Flags  Metric  Ref    Use  Iface
213.121.147.69    *               255.255.255.255  UH       0        0       0  ppp0
172.18.0.0        *               255.255.0.0      U        0        0       0  eth0
172.17.0.0        *               255.255.0.0      U        0        0       0  eth1
127.0.0.0         *               255.0.0.0        U        0        0       0  lo
default          213.121.147.69  0.0.0.0           UG       0        0       0  ppp0
```

There are three interfaces: eth0 , eth1 and ppp0 (as well as lo )

Packets with addresses in the network 172.18 go directly on interface eth0

# IP Addresses

A table from a machine with more than one real interface:

```
Destination      Gateway          Genmask          Flags  Metric  Ref   Use  Iface
213.121.147.69   *                255.255.255.255  UH      0       0       0  ppp0
172.18.0.0       *                255.255.0.0      U       0       0       0  eth0
172.17.0.0       *                255.255.0.0      U       0       0       0  eth1
127.0.0.0        *                255.0.0.0        U       0       0       0  lo
default          213.121.147.69   0.0.0.0          UG      0       0       0  ppp0
```

There are three interfaces: eth0 , eth1 and ppp0 (as well as lo )

Packets with addresses in the network 172.17 go directly on interface eth1

# IP Addresses

A table from a machine with more than one real interface:

```
Destination      Gateway          Genmask          Flags  Metric  Ref   Use  Iface
213.121.147.69    *               255.255.255.255  UH      0       0      0  ppp0
172.18.0.0        *               255.255.0.0      U       0       0      0  eth0
172.17.0.0        *               255.255.0.0      U       0       0      0  eth1
127.0.0.0         *               255.0.0.0        U       0       0      0  lo
default          213.121.147.69  0.0.0.0           UG      0       0      0  ppp0
```

There are three interfaces: eth0 , eth1 and ppp0 (as well as lo )

Otherwise packets are routed to the gateway 213.121.147.69 on the interface ppp0

# IP Addresses

A table from a machine with more than one real interface:

```
Destination      Gateway          Genmask          Flags  Metric  Ref   Use  Iface
213.121.147.69   *                255.255.255.255  UH     0       0       0  ppp0
172.18.0.0       *                255.255.0.0      U      0       0       0  eth0
172.17.0.0       *                255.255.0.0      U      0       0       0  eth1
127.0.0.0        *                255.0.0.0        U      0       0       0  lo
default          213.121.147.69   0.0.0.0          UG     0       0       0  ppp0
```

There are three interfaces: eth0 , eth1 and ppp0 (as well as lo )

The first row of the table is actually redundant here

# IP Addresses

A table from a machine with more than one real interface:

```
Destination      Gateway          Genmask          Flags  Metric  Ref   Use  Iface
213.121.147.69   *                255.255.255.255  UH     0       0     0    ppp0
172.18.0.0       *                255.255.0.0      U      0       0     0    eth0
172.17.0.0       *                255.255.0.0      U      0       0     0    eth1
127.0.0.0        *                255.0.0.0        U      0       0     0    lo
default          213.121.147.69   0.0.0.0          UG     0       0     0    ppp0
```

There are three interfaces: eth0 , eth1 and ppp0 (as well as lo )

Other information, in particular the flags, will be explained later

# IP Addresses

Use the route  command under Linux to see its routing table; or
ip  -r  route  show

# IP Addresses

How should we divide the 32 bit IP address into network and host parts?

8 bits for network? Then $2^8 =$ 256 networks each with $2^{24} =$ 16777216 possible hosts

- Not enough networks
- Too many hosts for most installations

# IP Addresses

24 bits for network? Then 16777216 networks each with 256 possible hosts

- Plenty of networks
- Not enough hosts per network for large installations

# IP Addresses

16 bits for network? Then 65536 networks each with 65536 possible hosts

- Not really enough networks
- Plenty of hosts per network

# IP Addresses

One solution: do all of the above

Divide the space of IP addresses into parts, where each part has a different network/host split

# IP Addresses

Class A networks. From 0.0.0.0 to 127.255.255.255

The leading bit of the address is 0

Have 7 bits for network and 24 bits for host

This is 126 networks (networks 0 and 127 are reserved) each with 16777216 host addresses

The address x.y.z.w has x as network, y.z.w as host

# IP Addresses

Class B networks. From 128.0.0.0 to 191.255.255.255

The leading bits of the address are 10

Have 14 bits for network and 16 bits for host

This is 16384 networks each with 65536 host addresses

The address x.y.z.w has x.y as network, z.w as host

# IP Addresses

Class C networks. From 192.0.0.0 to 223.255.255.255

The leading bits of the address are 110

Have 21 bits for network and 8 bits for host

This is 2097152 networks each with 256 host addresses

The address x.y.z.w has x.y.z as network, was host

# IP Addresses

The remaining addresses are kept for separate purposes

Class D. 224.0.0.0   to 239.255.255.255 ; leading bits 1110. Used for multicasting (details later)

Class E. 240.0.0.0   to 255.255.255.255 ; leading bits 1111. Reserved

# IP Addresses

An example: the University of Bath has been allocated addresses in the network 138.38

This is in the class B address range and so there are 65534 possible hosts

Network 3, a class A address, used to be allocated to the General Electric Company

Network 193.0.0, a class C address, used to be allocated to Réseaux IP Européens (RIPE), the Internet Registry responsible for the allocation of IP addresses within Europe

# IP Addresses

Two of the host addresses on each network are treated specially

Host parts of “all zeros” and “all ones” are not used as general host addresses, but are reserved for a special purpose

E.g., 138.38.0.0   and 138.38.255.255   in a class B

Thus the number of usable host addresses in a network is 2 fewer than you might think

# IP Addresses

Host part all 0s: “this host”. Originally specified to refer back to the originating host. But some implementations mistakenly used this as a broadcast address, so for safety it is not commonly supported as a valid host address. For, say, a class B network 17.16, a packet sent to 172.16.0.0 should boomerang right back to the sender. But rarely does

Host part all 1s: broadcast address to network. E.g., 172.16.255.255 sends to all hosts on the 172.16 network; very commonly used

(Network part all 0s: “this network”. E.g., 0.0.12.34 would send to a host on the current network. Again, not often implemented)

# IP Addresses

So this is why you have two fewer addresses available than you might think

. . . 255 is a broadcast address

. . . 0 may or may not be supported, so best to avoid it

# IP Addresses

Loopback addresses:

Network 127.0.0.0 : the loopback network. Always implemented. The address 127.0.0.1 is commonly used as a way for a host to send a packet to itself over the internal loopback network on interface lo .

Notice this is different from the same host sending to itself via an external network (e.g., using the interface's own address) as the former packet possibly won't go through the normal Ethernet/whatever software and hardware.

The loopback network is there even if there is no real network hardware attached

# IP Addresses

So the class scheme allows IANA to allocate large chunks of addresses to people who need them, and small chunks to those that only need a few

This scheme has been historically very successful, but with the growth of the Internet has revealed several weaknesses. These days, a classless allocation is used (CIDR, later)

Thus this allocation is sometime called classful

To understand classless allocation, we first need to look at subnetting

# IP Address Subnetting

Suppose you have been allocated class B network: 64 thousand host addresses are very hard to manage

Think of the broadcast traffic (e.g., ARP)

Physical/Technical issues (e.g, limits on Ethernet)

Political issues (e.g., traffic from one department must be kept separate from another department)

A single big network is not a very good idea

# IP Address Subnetting

We can use subnetting to split our network into smaller pieces

Subnets can be administered by separate departments and are joined by routers

Just like the Internet!

And to do this, also just like the Internet, we further split the host part into some bits for the subnetwork and the rest for the actual hosts

# IP Address Subnetting

Hosts will need to know which bits are the subnet part to be able to decide how to route packets: there is no class system here

We use a subnet mask

For example, the University of Bath has a class B, address 138.38. The top 16 bits are the network address

The netmask 111111111111000000000000 indicates which bits are in the network part

# IP Address Subnetting

The Department of Mathematical Sciences has a subnet consisting of addresses 138.38.96.0 to 138.38.103.255 (2048 host addresses)

This corresponds to the netmask
11111111111111111111100000000000

# IP Address Subnetting

| | | |
|---|---|---|
| network address | 138.38.96.0 | 10001010 00100110 01100000 00000000 |
| broadcast address | 138.38.103.255 | 10001010 00100110 01100111 11111111 |
| netmask | 255.255.248.0 | 11111111 11111111 11111000 00000000 |

A machine can tell if an address is on a network if the address ANDed with the netmask gives the network address

This is not on a nice byte boundary, so visually is harder for humans to work with using decimal x.y.z.w style notations

# IP Address Subnetting

So 138.38.100.20 is on the subnet

| | | |
|---|---|---|
| host address | 138.38.100.20 | 10001010 00100110 01100100 00010100 |
| netmask | 255.255.248.0 | 11111111 11111111 11111000 00000000 |
| AND | 138.38.96.0 | 10001010 00100110 01100000 00000000 |
| network address | 138.38.96.0 | 10001010 00100110 01100000 00000000 |

# IP Address Subnetting

But 138.38.104.20 is not on the subnet

| | | |
|---|---|---|
| host address | 138.38.104.20 | 10001010 00100110 01101000 00010100 |
| netmask | 255.255.248.0 | 11111111 11111111 11111000 00000000 |
| AND | 138.38.104.0 | 10001010 00100110 01101000 00000000 |
| network address | 138.38.96.0 | 10001010 00100110 01100000 00000000 |

# IP Address Subnetting

138.38 is split into many subnets of appropriate sizes for each Department, Centre or other sub-part of the University

Outside of 138.38 the subnetting is invisible so no changes to global routing tables are necessary if we rearrange our network

Subnets can be further subnetted for exactly the same reason

# IP Address Subnetting

The subnet is described as “138.38.96.0 , netmask 255.255.248.0 ”

More commonly as “138.38.96.0/21 ”, where 21 is the number of 1 bits in the netmask

You don't have to use the top n bits for a netmask, but it is overwhelmingly common to do so

The /n notation is only for a top-n-bit netmask

The “all 0s” and “all 1s” addresses now apply within the subnet: all 1's broadcasts to the subnet; and don't use all 0s

# IP Address Exhaustion

Everybody wants a class B as C is too small and A is too large

Called the Three Bears Problem

There are no class Bs left: they have all been allocated

# IP Address Exhaustion

Can we split some class As?

Doable, but needs everyone to take care their software understands that those addresses are no longer class A

Most class A's have now been split and the subnets allocated to various institutions

# IP Address Exhaustion

Can an institution simply use several class Cs?

Yes, but awkward as this leads to multiple networks, each needing separate routing

For example, having eight class C networks 194.24.0.0 to 194.24.7.0 would require everyone's routing tables to have eight entries that all point to the same destination

And internally to the institution there are eight separate networks, too

# IP Address Exhaustion

Class E has 286 million reserved addresses; can we use them?

Wouldn't last long; perhaps under a couple of years if allocated

More problematically, class E addresses are treated as illegal by much software, particularly on routers, so they are difficult to bring into play

# IP Address Exhaustion

Some while ago it was recognised that the growth of the Internet meant that a new way of allocating addresses was needed

Three solutions are used:

Change the way classes are defined and used

Use private addresses with network address translation

Increase the number of addresses available by changing the IP

We shall be looking at each of these

# CIDR

Classless Interdomain Routing (CIDR) takes class C networks and joins them together in such a way that simplifies routing

Blocks of C addresses are allocated to regions, e.g.,

| | |
|---|---|
| 194.0.0.0-195.255.255.255 | Europe |
| 198.0.0.0-199.255.255.255 | North America |
| 200.0.0.0-201.255.255.255 | Central and S America |
| 202.0.0.0-203.255.255.255 | Asia and the Pacific |

# CIDR

Starting with about 32 million addresses per region

This allows easy routing: anything 194 or 195 goes to Europe

Repeat the idea within each region: contiguous block of C networks are allocated to ISPs or organisations

Keeps simple routing within the region

# CIDR

E.g., 194.24.0.0 to 194.24.7.255 , normally written 194.24.0.0/21 or even 194.24/21 : exactly like subnetting

| | |
|---|---|
| 194.24.0.0 | 11000010 00011000 00000000 00000000 |
| 194.24.7.255 | 11000010 00011000 00000111 11111111 |
| 255.255.248.0 | 11111111 11111111 11111000 00000000 |

Any packet with address that has addr AND 255.255.248.0 = 194.24.0.0 should be routed to that ISP or organisation

A network of $2^{32\ \ 21} = 2^{11} =$ 2048 addresses, i.e., 2046 hosts

# CIDR

This is a very flexible and backwards-compatible scheme

End hosts do not need to know about CIDR

Classless networks can be subnetted

CIDR has allowed the continued growth of the Internet well beyond the original possible size

We have repurposed class A and B networks similarly

# CIDR

In fact, classful networks are no longer used : CIDR is the only way addresses are currently allocated

CIDR merges small networks into a larger one

Subnetting divides a large network into smaller ones

CIDR is sometimes called supernetting

Thus we have:

> Classful: implicit, fixed split of network/host
>
> Classless: explicit (netmask), variable split of network/host

# CIDR

CIDR has been very successful, and has extended the life of the Internet significantly by providing a source of addresses from the previously underutilised classful ranges

Not enough. . .

# CIDR

There are currently about 26 billion devices connected to the Internet (statistica.com ; 2018)

But there are only about 4.3 billion usable IPv4 addresses

How is this possible?

# NAT

This brings us to the second approach to address exhaustion

Some IP addresses are reserved for private networks, originally reserved to allow local experimentation:

10.0.0.0-10.255.255.255 (Class A)

172.16.0.0-172.31.255.255 (Class B)

192.168.0.0-192.168.255.255 (Class C)

One class A-size network, 16 class B and 256 class C-size networks are guaranteed never to be allocated for public use in the Internet

Routers on the public Internet will never forward packets with such addresses, and will simply drop them immediately

# NAT

But such addresses can be used by anyone locally for any purpose: a common use is NAT

Network Address Translation (NAT) uses the malleability of packets to map many hosts onto a single address

A private network can be set up, using one of the above address ranges, e.g., 10/8

A gateway host joins the private network to the public Internet, modifying the addresses on packets as they go past

# NAT

A packet from 10.0.1.1   (A) is sent to 212.58.226.33   (B)

# NAT

The gateway overwrites the source address with its own public address (G)

# NAT

The packet reaches B in the normal way

# NAT

B replies with a packet with destination address G

# NAT

The gateway recognises this packet as a reply to A and rewrites the destination address to A before passing it on to the private network

# NAT

A thinks it is connected to the public Internet, and B thinks data is coming from G

# NAT

G needs to keep a record of connections from A to the world and recognise replies to outward travelling packets

C will want to do the same as A; so G must be able to distinguish replies to A from replies to C; even if both were communicating with B

And rewrite the replies to C with C's address

This is all doable in practice! Explanation later, in the next layer

Exercise. If both A and C are communicating with B, what are the addresses on their packets as they reach B? And on the replies as they reach G?

Exercise. Compare with bridging, a similar idea but for very different reasons

# NAT

As a fortunate side-effect, NAT provides some measure of protection to hosts on the private network from external attack

Machines on the public Internet (e.g., B) cannot initiate traffic to A as 10.0.1.1 is a private, unroutable address

No public router will forward a packet with such an address: it will simply drop it

External hosts will generally not even know what A's (private) address is as they never get to see it

Even if a packet somehow gets to the gateway, the gateway will not know how to rewrite its address as this was not a reply to an outgoing packet; so it get dropped here, too

# NAT

NAT has helped immensely to mitigate the address exhaustion problem

Previously, every host on a network would need a separate public IP address

The growth of the Internet at home, for example, would have sucked up addresses at a huge rate

But now all your home appliances can share just one public address

Exercise. Count the number of network attached devices you have at home

# NAT

Problems arise when the data in the packet contain IP addresses that, say, will be used to set up new connections. E.g., the File Transfer Protocol (FTP)

(Original) FTP would send an IP address to the server to indicate where to set up a new connection

In our example, this would be its private, unroutable address that the external server couldn't contact

Unless the gateway is intelligent enough to look inside the data and know where the IP addresses are to be found (in the application layer data) and rewrite them the addresses will remain untranslated and the protocol will fail

# NAT

Not many protocols do this kind of thing these days, but each one of those that do must be treated specially by the NAT gateway

Note this is a problem due to a violation of layering in the protocol: IP layer information in the application layer

Exercise. Read about FTP, Universal Plug and Play (UPnP) and the Simple Service Discovery Protocol (SSDP)

# NAT

NAT is used widely as it is very effective

It allows you to have many machines but only use one public address

Many mobile phone companies are using carrier grade NAT to supply IP connectivity to the millions of phones they manage

Carrier grade NAT: NAT done in the ISP rather than by the end-user

Exercise. What IP address does your phone have for its mobile data connection (not its Wi-Fi connection)?

Exercise. Read RFC6598 and about 100.64.0.0/10

# NAT

Without NAT, public IP addresses would have run out years ago

But there are costs to NAT

- Complexity in the gateway software
- Scalability problems in the gateway tracking large numbers of connections
- Bad interactions with some protocols
- Difficulty of making end-to-end connections when both ends are behind a NAT gateway (e.g., Skype, SIP)
- Loss of “an IP address identifies a host uniquely” (a problem for law enforcement)

# NAT

There is also the inability for external hosts to initiate connections to hosts behind NAT

So you can't run servers on hosts behind the NAT

But this invisibility is often regarded as a good security feature

This can be worked around, though not neatly

Exercise. Read about port forwarding

Exercise. Read about STUN

# NAT

NAT is the reason the Internet did not grind to a halt many years ago through the lack of available addresses

Thus putting off the need for a proper solution to the problem

Some people still argue that there is no reason to do anything else than use more NAT

Even to the extent of using multi-level NAT (NAT within NAT!)

# NAT

But even with CIDR and NAT, the entire range of usable IPv4 addresses has now been allocated

Which is to say that IANA has distributed all its reserves of addresses to the Regional Internet Registries (RIRs)

And now the RIRs are running out

RIPE (covering Europe, Middle East and Central Asia) say that their allocation will run out in November 2019

We need a more radical solution

# IPv6

The next approach to the IP address exhaustion problem is to change IP itself

The next version of the IP is IPv6 (sometimes called IPng for IP next generation)

Slowly growing in use, but it will take a while to supplant IPv4

128 bit addresses; CIDR-style allocation only

Exercise. Find out about IPv5. And IPv0-IPv3

# IPv6

IPv6 was designed to

- have a larger address space
- reduce the size of router tables
- simplify the protocol so routers can process packets faster
- provide security and authentication
- pay proper attention to type of service (DS)

# IPv6

have better multicasting support

have mobile hosts with fixed IP addresses

allow room for evolution of the protocol

permit IPv4 and IPv6 to coexist during the transition

# IPv6

IPv6 Header

# IPv6

Version, 4 bits. The number 6. This is identical in position to IPv4 and can be used to distinguish packets in mixed-version environments. In an Ethernet frame, IPv4 has protocol number 0800, while IPv6 is 86DD, but remember you might be using a different physical layer that does not give the type of its data

Traffic class, 8 bits. Like TOS (DS) in v4

Flow label, 20 bits. Allows routers to recognise packets in a single flow and treat them identically. In essence a virtual circuit identifier

# IPv6

Payload length, 16 bits. The number of bytes following the fixed 40 byte header. Unlike v4, does not include the header in the count

Next header, 8 bits. Like the protocol field in v4, but also allows for v6 optional header fields, if any

Hop limit, 8 bits. The TTL field, renamed to make it clear how it is actually used

# IPv6

Source and destination addresses, 128 bits each.

Four times as long as v4 addresses

$2^{128} = 3 \quad 10^{38}$ addresses, enough for an address for every molecule on the surface of the Earth

There are unicast, multicast and anycast addresses: details later

# IPv6

Addresses are typically written in hex, with colon separators, e.g., fe80::21c:c0ff:fea3:99f4

The :: may appear once as a shorthand for a string of 0s. As many as you need to make the address up to 128 bits

Thus the above address is
fe80:0000:0000:0000:21c:c0ff:fea3:99f4

Or:
1111111010000000 0000000000000000 0000000000000000
0000000000000000 0000001000011100 1100000011111111
1111111010100011 1001100111110100

# IPv6

The University of Bath has been allocated 2001:0630:00e1::/48

Meaning 128  48 = 80 bits of address for hosts on the University network

$2^{80}$ = 1:2  $10^{24}$ addresses, which is about 280 trillion times the size of the whole current IPv4 Internet!

Exercise. Check my arithmetic

# IPv6

Exercise. Look up the IPv6 address of facebook.com

# IPv6

There are no fragmentation fields

A router never fragments, but drops the packet and sends back a “packet too big” message to the source. The source can then send smaller packets

Processing within a router is therefore much simpler and packets can be sent on much faster

Every IPv6 host is required to do path MTU discovery

# IPv6

The flow label was intended to help identify packets within a single “flow”, i.e., a connection or session

Packets with the same flow label can be treated identically and so sent on faster by a router

# IPv6

No header length field: the header is always 40 bytes

No checksum field: there are checksums in other layers and networks are reasonably reliable. The protocol designers thought that yet another checksum would not be helpful here

Also we don't have to recompute a checksum in every router as the TTL decreases. Again, faster

# IPv6

v4 has 13 fixed fields; v6 has 8; much simpler for a router to process

v6 addresses are 4 times the length, but the header is only twice as long

# IPv6

The next header field daisy-chains options, called extension headers, or gives the protocol (TCP, UDP, etc.) of the next layer

Option Header

Thus the only limit on the options is the total datagram limit

Furthermore, most options are not even looked at by routers: again to get faster processing in the routers

# IPv6

Optional headers include:

Routing options: c.f., loose source routing in IPv4

Authentication

Security

Jumbograms: packets up to 4GB in length!

And others

Note the type of the header option is given in the previous header option, or the main IPv6 header for the first option

# IPv6

## Fragmentation

A note on fragmentation in IPv6: end hosts are expected to use path MTU discovery to find the MTU; or just send packets no larger than 1280 bytes

Exercise. But IPv6 does allow for the unusual case where the source can't produce small enough packets via an extension header: read about this

# IPv6

## Transition to v6

IPv4 address allocations have run out, so we need to move to IPv6

But it is expensive to do so, as it needs software rewrites, so many people (ISPs, websites etc.) are pretending the problem does not exist

Even though the majority of modern routers and end hosts contain the necessary IPv6 software support

We can't turn off the Internet and replace v4 by v6 overnight

By design, the two protocols can run side-by-side on the same networks

# IPv6

## Transition to v6

IPv6 was devised in 1996, but has yet to achieve mainstream use

As of November 2019 figures from akamai.com say they see about 33% of UK traffic is IPv6

India: 65%

USA: 51%

The majority of countries are under 1%

# IPv6

## Transition to v6

Some large companies, e.g., Google, support IPv6 connections: they want to encourage the transition

But many ISPs don't: so most home users can't use it

There have been a variety of transition mechanisms suggested, often based on NAT-like packet mangling

But they are all complicated and unsatisfactory, for the same reasons NAT is unsatisfactory

# IPv6

## Transition to v6

Exercise. Read about NAT64 (RFC6146) and DNS64 (RFC6147) for connecting IPv6-only clients to IPv4 servers

Exercise. Read about IPv4 mapped addresses, that allows server code that is purely IPv6, but accepts IPv4 client packets

Exercise. Read about 464XLAT (RFC6877) for IPv4-only clients that translates IPv4 addresses to IPv6 addresses for transport and then back to IPv4 addresses for the destination server

# IPv6

## Transition to v6

In the near future IPv6 will need to be supported properly by everybody

Exercise. Find out if your home ISP supports IPv6

Exercise. RFC6177 suggests giving home users a /56 network. How many host addresses does this correspond to?

# Addresses

We now take another look at IP addresses

In particular there are several types of address that can refer to one or more than one host at a time

# Addresses

IPv4 has three types of address

Unicast: an address refers to a single destination (ignoring NAT!). A “normal” address

Broadcast: as in the link layer, a single packet goes to every host in the local network. But, now, the “network” is at the IP layer, so may comprise more than one link layer network

Multicast: in between uni- and broadcast. A single packet goes to one or more hosts

# Addresses

IPv6 adds

> Anycast: a packet goes to any one of a selection of servers, usually the “closest” in some sense

In fact, IPv6 also removes broadcast as its job can be done by multicast

So we need to look at four types of address

# Unicast Addresses: v4 & v6

Unicast

- 1-to-1 data flow; one source, one destination
- Most current IP traffic is unicast

# Broadcast Addresses: v4

Broadcast

- 1-to-many data flow; one source, “all” destinations
- Broadcast is simple: a single packet read by all hosts on the local network
- Reduces traffic on the local network as (for most link layers) we don't have copies of mostly-identical packets, one for each destination, but just one packet that is read by every host
- Scales well (locally): it is independent of the number of destination hosts
- Don't have to know how many destination hosts there are

# Broadcast Addresses

Broadcasts are generally limited to the local network: otherwise the entire Internet would be permanently flooded

We have seen IPv4 broadcast addresses before: when the host part of the IP address is all 1s

E.g., 172.16.1.255 on the subnet 172.16.1/24

We can also use 255.255.255.255 as a broadcast to the local network for when we don't yet know our network address

# Broadcast Addresses

As mentioned, IPv6 does not support broadcast separately, so there are no IPv6 broadcast addresses per se

IPv6 uses multicast to achieve the same effect

# Multicast Addresses: v4 & v6

Multicast

For sending a single packet to multiple hosts, not necessarily all hosts

E.g., for streaming radio we could send individual unicast packets to all listening hosts, but it would be much more efficient to send a single packet that the listening hosts pick up and the non-listening hosts don't

Also, we can't use broadcast as broadcast is network-limited: perhaps listeners are spread far and wide over multiple networks

# Multicast Addresses: v4

One class of IPv4 addresses is reserved for multicast

In IPv4, class D (224.0.0.0 to 239.255.255.255 ) addresses are used for multicast

# Multicast Addresses: v4

Multicast groups are formed from those hosts that wish to receive packets from a given source. e.g., a group to listen to BBC Radio 4

A multicast group id is a 28 bit number with no further structure: about 270 million possible groups

The set of hosts listening to a particular multicast address is a host group

Host groups can cross multiple networks and there is no limit on the size of a group (and generally you can't know how big the group is)

# Multicast Addresses: v4

Some group addresses are preallocated by IANA: the permanent host groups

| | | |
|---|---|---|
| 224.0.0.1 | : | all multicast aware hosts on this subnet |
| 224.0.0.2 | : | all multicast routers on this subnet |

# Multicast Addresses: v4

Not all IPv4 hosts support all of multicasting

- level 0: no support
- level 1: can send multicast, but can't receive (receiving is harder as it involves understanding groups)
- level 2: can send and receive

# Multicast Addresses: v4

The process of joining and leaving groups is governed by the Internet Group Management Protocol (IGMP)

A host that wishes to join a multicast group provided by a server sends an IGMP message towards the server

The routers on the path to the server take note and so know to route multicast packets for this group towards the joining host

The server itself is not interested or involved in the IGMP message

# Multicast

Unicast vs. Multicast

# Multicast Addresses: v4

Similarly for a host leaving a group: a host is supposed to send an IGMP message towards the server that the routers can read and act upon

Extra complication arises as hosts may not (or can't if they crash) always send “group leave” messages

So there is more protocol to monitor and maintain groups using timeouts and maintenance messages

Exercise. Read about this

# Multicast Addresses: v4

TTL plays a special role in multicast on IPv4: it defines the scope of a group, which is how wide an area a group may range over

0 Host

1 Subnet

< 32 Organisation

< 64 Region

< 128 Continent

< 255 Global

# Multicast Addresses: v4

The definition of “organisation”, “region”, etc., is left open for the network administrator to define as they wish. Routers are set to discard multicast packets that would cross a boundary

Exercise. Also see RFC2365 for an alternative mechanism that uses addresses (239.0.0.0 to 239.255.255.255 ) to limit scope

# Multicast Addresses: v4 & v6

In IPv4 IGMP is a separate protocol and its packets are layered directly over IP

In IPv6 the corresponding protocol is called Multicast Listener Discovery (MLD), and forms part of a larger protocol suite: ICMPv6 (see later)

All IPv6 hosts are required to implement multicast

# Multicast Addresses: v6

IPv6 multicast is much as v4, but simplified

Addresses start with hex FF

Four bits of flags, including the T bit which means transient group (as opposed to a permanent IANA allocated group)

Four bits of scope. Rather than using TTL

# Addresses

## Multicast

Note that we should be careful when talking about the number of packets on the local network

When we say “a packet only goes to a certain set of machines” we really mean “only a certain set of machines process the packet”

The network interface card may filter packets, but those packets are still occupying the network in a shared medium like Ethernet

# Addresses

## Multicast

A broadcast and a multicast will result in the same number of packets on the local network, but

- a multicast will only be read and processed by a subset of machines
- a broadcast is limited to the local network, while a multicast can spread far and wide

And both are better than multiple near-identical unicast packets

Exercise. Find out what IPv6 needs to do to broadcast to the local network

# Addresses

## Multicast

Multicast is not used as much as it should be

It is used in routing protocols (i.e., those protocols that help routers create their routing tables), but relatively little elsewhere in IPv4

Exercise. Read about the Simple Service Discovery Protocol (SSDP)

Exercise. And the Multicast Domain Name System (mDNS)

# Addresses

## Multicast

Multicast is hard to use for an on-demand system (e.g., BBC iPlayer, Netflix) as it requires everyone in the group to be receiving the same thing at the same time

There has been some experimentation with hybrid systems (a “buffer-up” unicast burst followed by a shared multicast stream), but not much uptake

Most big streaming providers rely on having many local distribution points containing identical data

# Addresses

## Multicast

The source supplies (relatively few) distribution points using unicast, which serve content directly using unicast

Exercise. Read about content delivery networks

# Addresses

## Multicast

Furthermore, while ideal for a TV broadcast service, most people (e.g., BBC and the like) use unicast as multicast is not well supported in home systems

Routing companies want to avoid supporting multicast, claiming undue complexity to support it: each group needs extra state in every router the multicast traffic passes through, making scaling to the full Internet a problem

A router must keep a record of all multicast paths passing through it, so routers on popular paths (e.g., in internet exchanges) might need to keep a large amount of data

# Addresses

## Multicast

Multicast is used by some pay-tv services, but usually in the context of a closed and controllable system, e.g., a institutional intranet multicasting a seminar, or holding a multi-way video conference

Generally in the case where the same institution owns all the infrastructure from source to destinations

Exercise. Read about BT TV

# Addresses

## Anycast: v6

Anycast

Anycast in IPv6 sends a single packet to a single destination chosen out of several possible destinations

For example, replicated Web servers: have many servers around the world with identical content and the same anycast address. A browser would get pages from the closest server, thus sharing load

The reply would be unicast

# Addresses: v6

## Anycast

Only works well with connectionless transport protocols (see later) as multiple requests might go to different servers: this doesn't fit well with connection-oriented protocols

Address format?

Any unicast address that happens to be assigned to more than one server. It is up to the routers to figure this out

# Addresses: v6

There are anycast groups, much as multicast groups and a join/leave protocol

Notice the symmetry: muticast is groups of clients, while anycast is groups of servers

Anycast has plenty of potential, but we need to be using IPv6 to get it properly, though some people do support it in IPv4

# Addresses

Exercise. More allocations of addresses: read about 192.0.2.0/24 , 198.51.100.0/24 and 203.0.113.0/24 that are reserved for documentation purposes (RFC5737)

# Addresses

How does a host get an IP address?

An Ethernet address is burned into the hardware, so there's no problem there

IP addresses are software addresses, so they must be set up somehow

The simplest way is for the host simply to be configured to have that address, stored in a configuration file on the host somewhere

An administrator takes into account certain criteria, e.g., network or subnetwork addresses, and gives the machine a currently unused address

But it is not always feasible to do this

# DHCP

Not all machines have administrators, e.g., home PCs

Some administrators are not sufficiently competent to allocate addresses correctly, e.g., home PCs

Some installations have too many machines to get around and configure them all, e.g., in the library

Some installations have machines that come and go all the time, e.g., laptops in the library