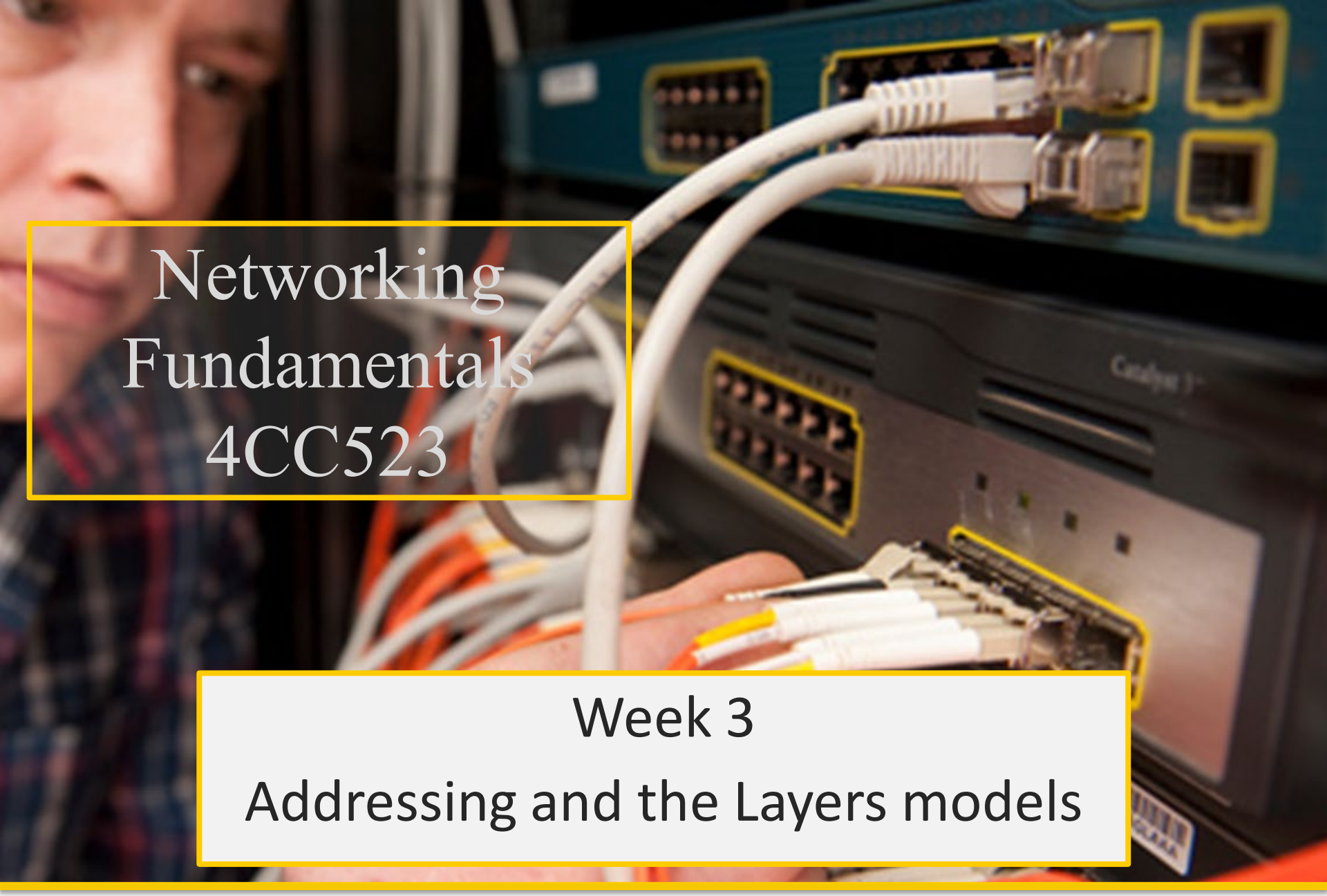


Networking Fundamentals 4CC523

Week 3

Addressing and the Layers models

UNIVERSITY OF DERBY

Individual Health & Safety Responsibilities

To help prevent the spread of the COVID-19 virus we ask you to carry out the following, to help keep everyone safe & aid with contact tracing.

- **If you have any COVID-19 symptoms leave the class immediately if possible & return to your accommodation. Contact the NHS on 111 & then contact student wellbeing.**
- **Wash your hands or sanitise them upon entering/exiting the area.**
- **Sign in/register via the Tap In system or alternative system in place.**
- **Identify a seating area & log the desk number on 'Who Sat Where?' on the Udo app. For people who don't have access to the app inform your lecturer to use the alternative system.**
- **Always maintain social distancing.**
- **When the class finishes, please use the cleaning materials provided to clean down the area you have used & then wash or sanitise your hands.**
- **We welcome the wearing of face coverings in teaching areas where possible.**

Thanks for your help and continued support.

TEF Gold

derby.ac.uk

IMPORTANT INFO!!

- BE AWARE - YOU ARE BEING RECORDED

  This is part of the normal lecture process, recordings will be made available to you to support your studies

  Inform me as soon as possible if you object to recordings being made (and remind me every session!!)

- DID YOU TAP-IN?? DO IT NOW!

Week 3: Objectives

- Layers concept
- Encapsulation
- Data units
- Logical port numbers
- Logical addressing
- Physical addressing
- Segmenting

Recap…

- Last week we looked at the components of a network
- Terminal devices
- Connecting technologies
- Intermediary devices

- Mentioned something about layers, 1,2 and 3

The “Layers Models”

- IP communication is “Standards based”
- We do not need to re-write or redefine the process each time we use the network
- Can be implemented on any device and will communicate
  - No separate internet for Microsoft, Apple, Linux
  - Even the darkweb uses the same protocols

The “Layers Models”

- Can enter and leave the model at any layer to suit requirements – more later
- Do not need to implement every layer
- Each layer represents a “specialism”
- Very important concept!
  - Need to define only the boundary conditions
  - Works for many programming paradigms (and other stuff)

Cup of Tea (restaurant) process

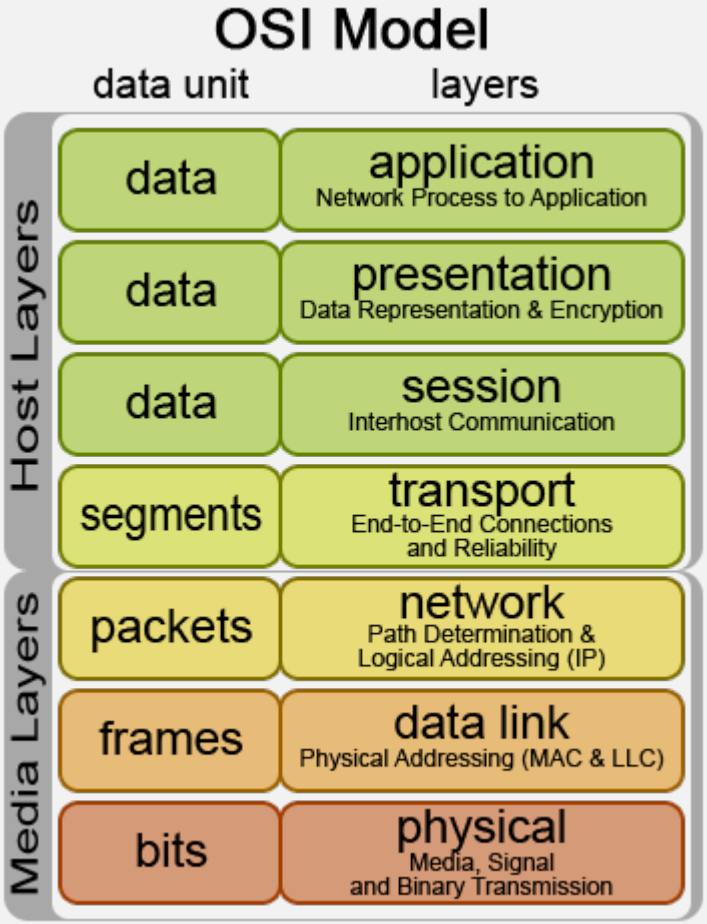
- At front of queue, ask for tea, pay, take tea away
- Layer A – want tea
  - Data exchange: Request beverage type ->, confirm availability <-
- Layer B – Check price list
  - suggest cost <-, agree price -> (Contract Law, Legal conditions)
- Layer C – make tea
  - Service provided (consumer has no concept of process)
- Layer D – pay for tea
  - Offer payment ->, accept, confirm completion <- (Contract Law)
- Layer E – take tea away
  - pass goods <-, confirm acceptance ->

Cup of Tea (restaurant) process

- 3 layers here, not 5 (and they are all complex!)

- Layer A – Concept
- Layer B – Contract
- Layer C – Process
- Layer B – Contract
- Layer A – Concept (need)

The Open Systems Interconnection (OSI) "7 Layer" Reference Model



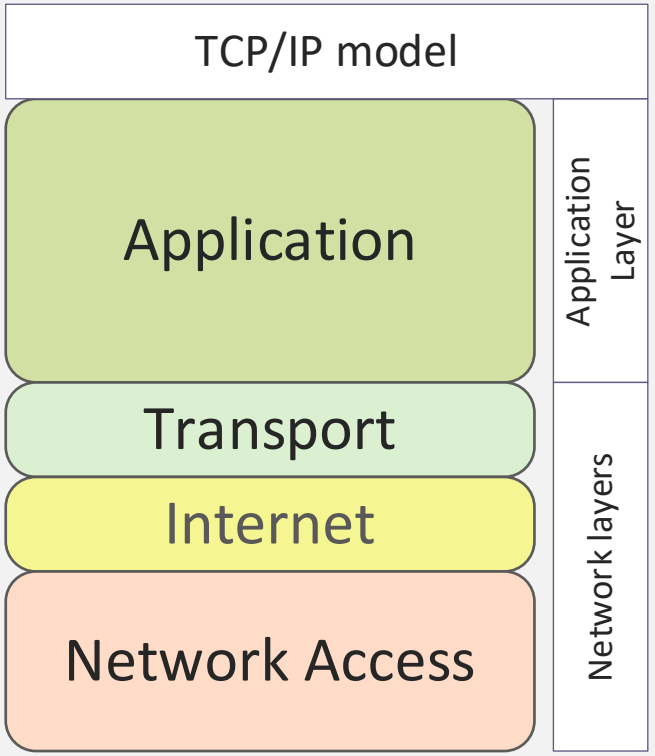
- A lot of information in one small diagram
- We will be coming back to most (all) of the layers later
  - Sometimes the data link layer is split into sub layers of LLC and MAC. This does not make this an 8 layer model.

The TCP/IP 4 layer model

- An **alternative** view
- Combines some layers
  - Good in places
  - Bad in places

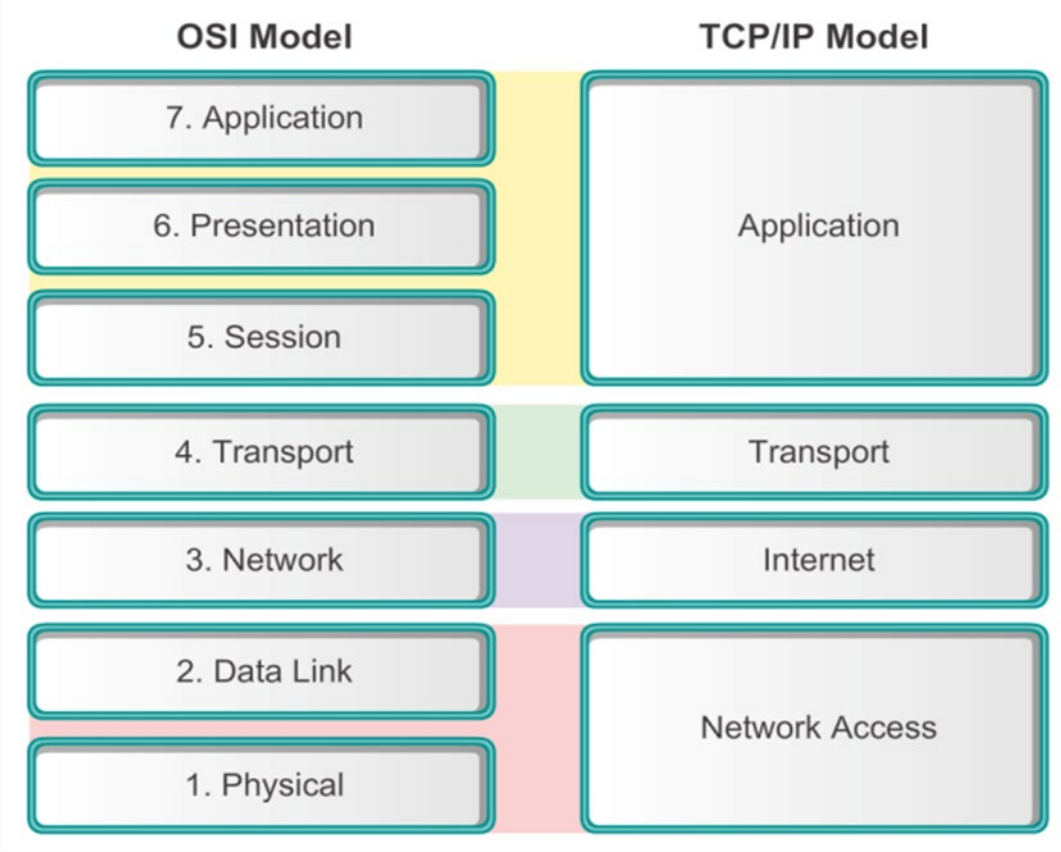
  (in the view of a network engineer)

  - Sometimes Network Access is split into two – becomes 5 layers



Comparing OSI and TCP/IP Models

- OSI
  - Numbers
  - Presentation
  - Session
  - Data Link
  - Physical



- TCP/IP
  - Internet
  - Network Access

Models Concept

- The important point:

- The main processes are predefined, only need to know the information exchange needed between levels (processes)

Encapsulation

- We add information to our data as we move from “Upper Layers” to “Lower Layers”
- We wrap the original information in enabling information – we add “headers”
- This is called encapsulation
  - As we transition back “up the layers” we use and strip off the enabling information to get back to the original data
  - This is de-encapsulation (decapsulation?)

The principle of tea

- Concept: thirst
- Data “I want tea”
- Encapsulated data “Cost will be £7.90 (I want tea)”
- Encapsulated data “I agree to the price (Cost will be £7.90 (I want tea))”
  - Tea is made
- Decapsulated data “Cost will be £7.90 (I want tea)”
  - Money exchanged
- Decapsulated data “I want tea”
  - Tea delivered
- Concept satisfied

Network Protocols

- Perform different functions within the network
- How the message is formatted or structured
- The process by which networking devices share information about pathways with other networks
- How and when error and system messages are passed between devices
- The setup and termination of data transfer sessions

Message Formatting and Encapsulation

- Example: A personal letter

- Suppose we need to send a message to someone else

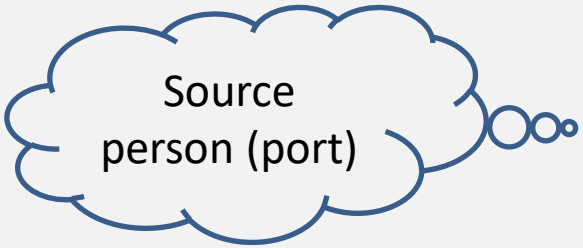
12 July 1983

Dear Mrs Smith,

I would like to invite you for tea on Saturday week. We shall have some jolly nice cakes that I have baked myself.

I do hope that you can come,

Yours Sincerely
Mrs L Jones
(aka The Boston Poisoner)



Letter Encapsulation

Inside the envelope:

- The message (Payload)

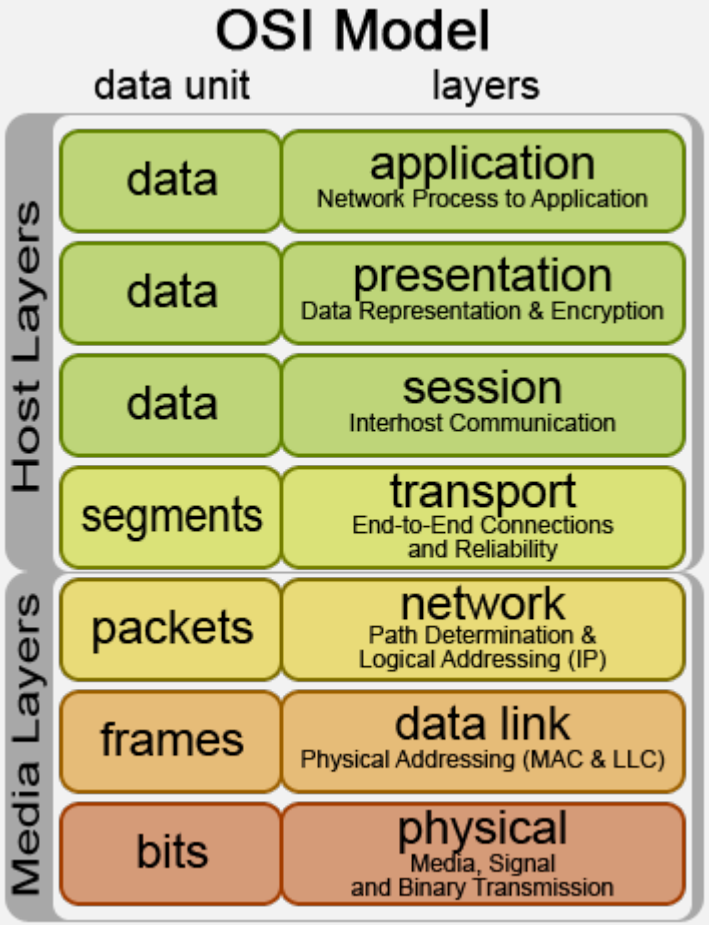
Wrap the message in an envelope:

- The intended (logical) destination
- The originator (logical) location for any reply

State the individuals at the addresses (never changes)

- The intended (physical) recipient
- The (physical) originator

The Open Systems Interconnection (OSI) "7 Layer" Reference Model



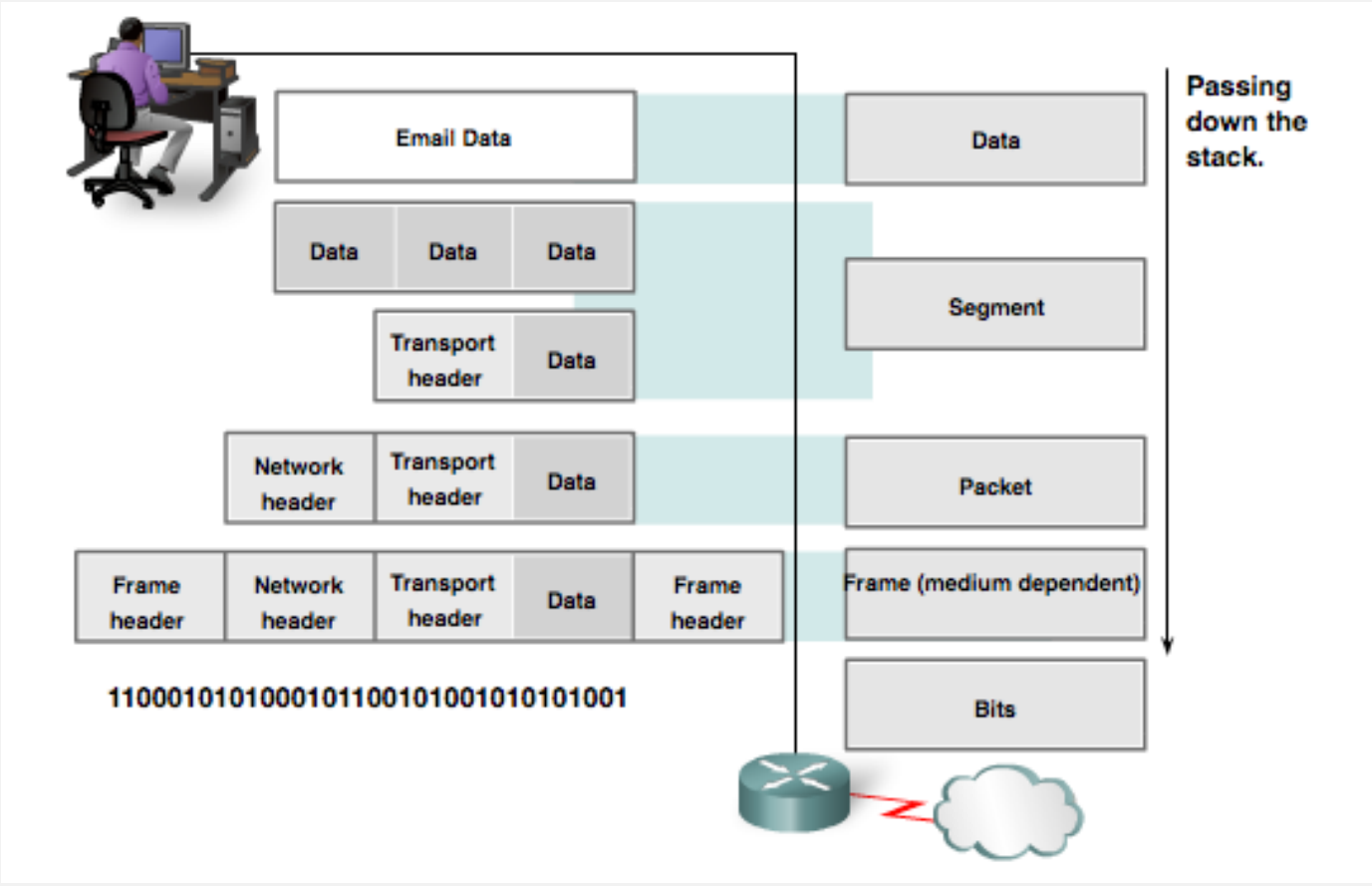
The Letter
The concept / thought
The layout / language
Conversation? Formality
Normal/recorded delivery
The envelope address
Intended recipient
Mailbox, collect, post office

Encapsulation simplifies interaction

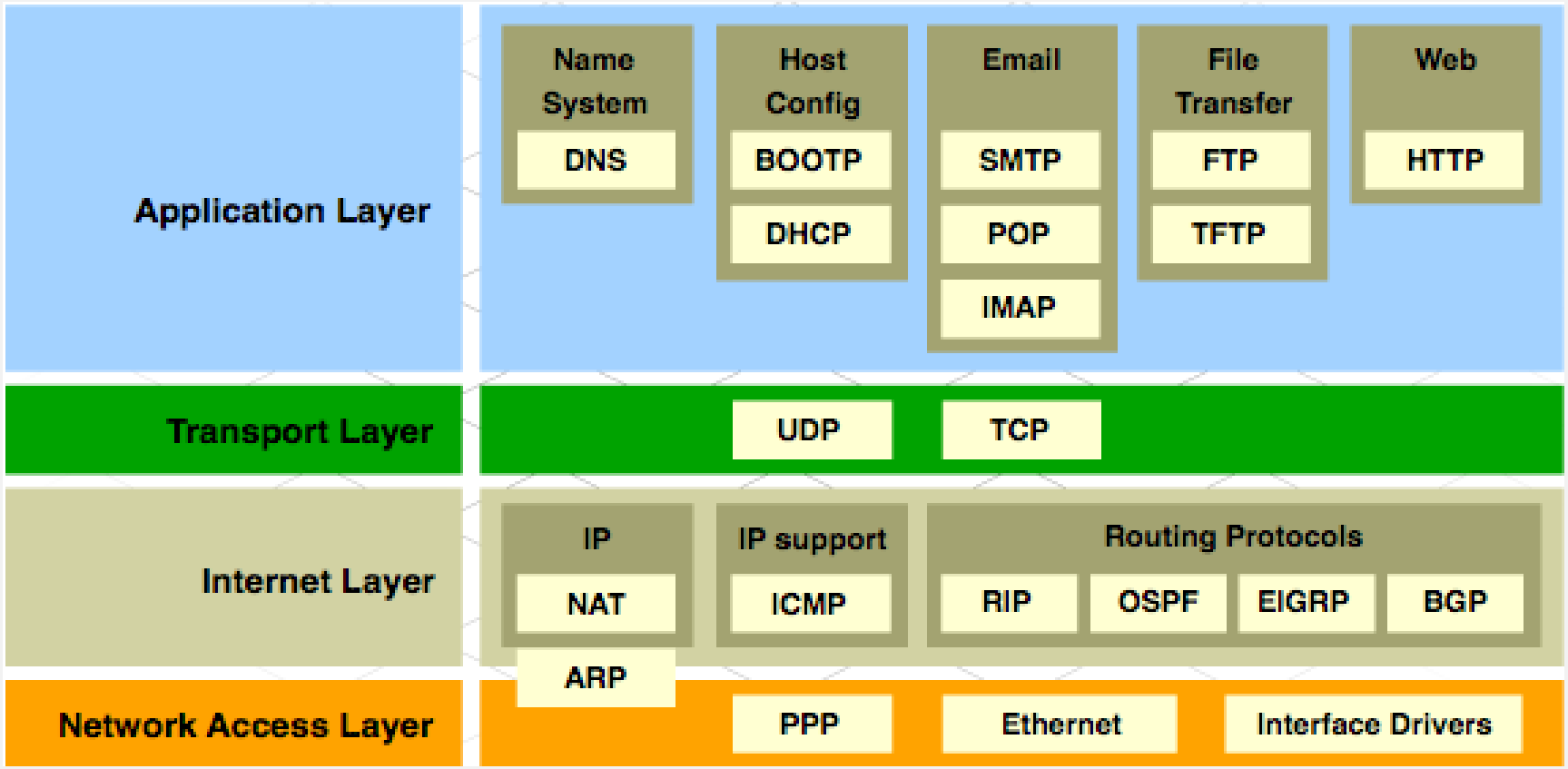
- The layers do not “understand” each other
- At each layer of encapsulation, the layer “sees” only bits and bytes
- A pdu from layer 5/6/7 has transport headers attached at layer 4, layer 4 only sees data from above
- The pdu from layer 4 has IP address headers attached at layer 3, layer 3 only sees data from above
- Provided that we know which layer we are at we do not need to show the content of a pdu from a layer above

Protocol Data Units (PDUs)

- Data
- Segment
- Packet
- Frame
- Bits



TCP/IP Protocol Suite



Port Numbers

- Mentioned before, each Application Layer process has an identifier known as a Port Number
- This is a logical numeric descriptor that directs the data to the application that will understand it
  - This is normally one of many applications (services) that are running in the background of the “server device”

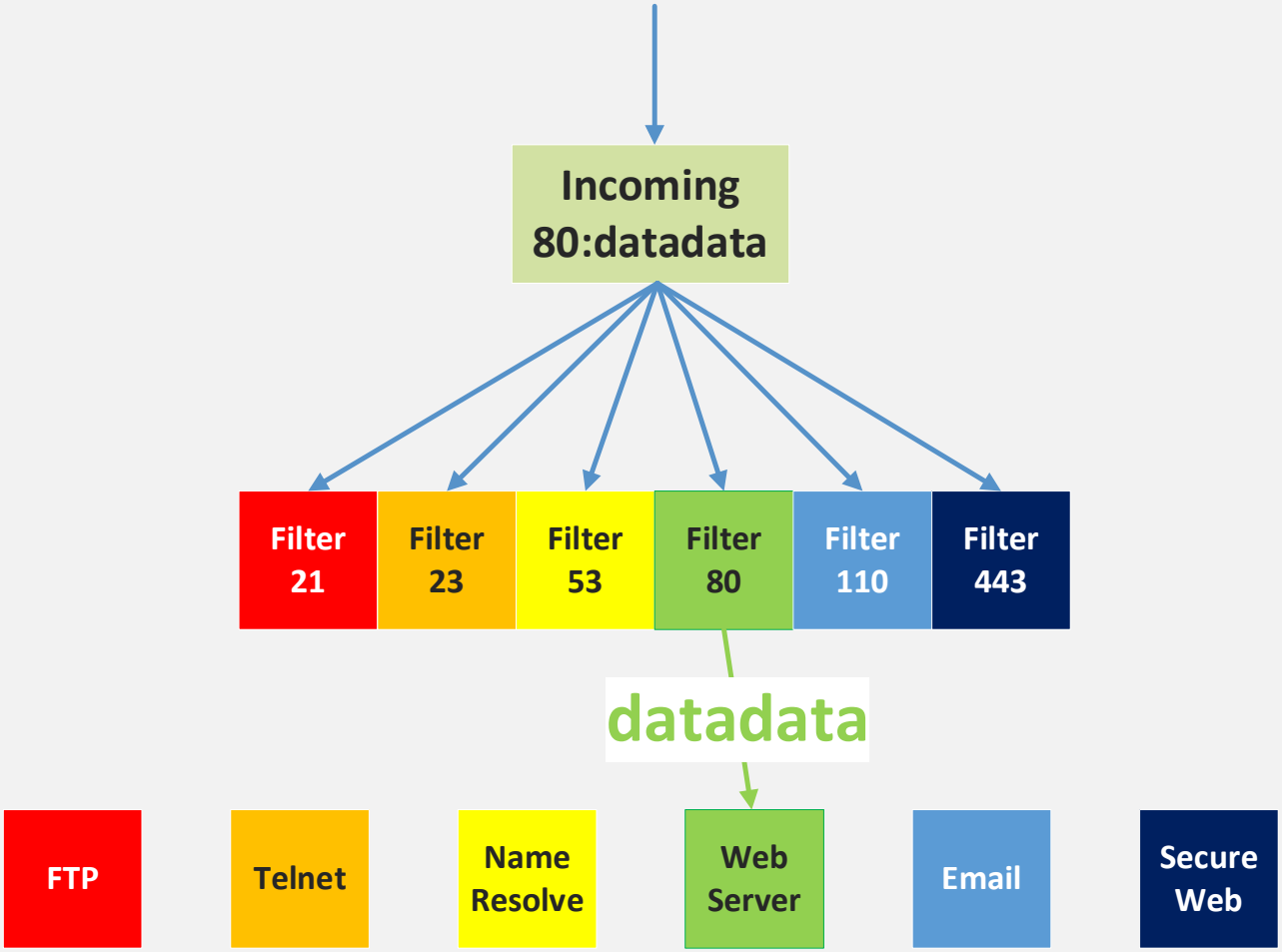
Port Numbers

- Example: we send http:// requests with a descriptor of “80” (port 80)
- The receiving device looks at port numbers that it has open – applications that are running
- If there is a match (80 required, 80 available) then the port number gets stripped away and the information / data is passed to the application
  - The pdu is decapsulated to reveal data that “makes sense” to a web server

Port addressing

- This is addressing at the simplest (highest) level
- Along comes some data “addressed” to a service at number 80 and that is where it gets delivered
  - There is a return port address that the “filter” needs to remember but we can see that another day
- We are sending data to the correct service

Port addressing



IP addressing

- Next level addressing
- This is used to make sure that the data is being sent to the correct device (the one running the service that we want to use)
- IP addresses are logically assigned – they match (define) the network that you are in
- If you move the device to another LAN it will normally need another address – we can change it whenever we want to – it is not fixed

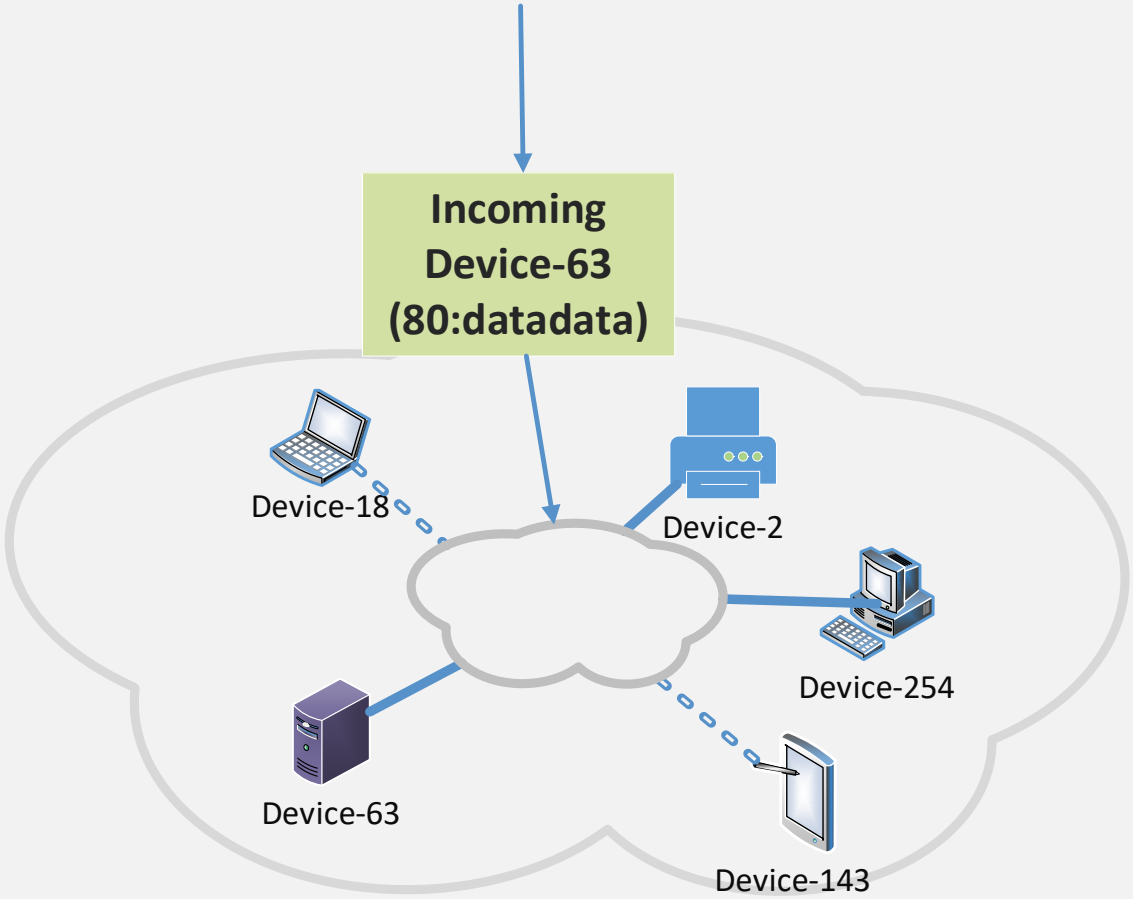
Delivery options

- We have options about finding the right device
- Broadcast: Send the message to everyone and hope that one device is running the service that we want
- Unicast: Send the message to one device knowing that this is the correct one – but how?
- Multicast: Send the message to some but not all devices – sounds complicated!

IP addressing

- Every device on the network has another logical descriptor to identify that it is part of this network – the Host part of the IP address
- This is used to make sure that the data is being sent to the correct device (the one running the service that we want to use)
- The **segment** with the port addressing had another header that contained this address
  - And a return address, something for another day

IP Addressing



IP addressing

- In this example, “device-63” is the server (?) at the bottom left
- When Device-63 reads the incoming **packet**, it strips the ip address and passes on the information 80:datadata
  - we already know what to do with that
- We know that we are sending the data to the correct Device!

Ethernet (hardware) addressing

- The diagram implies that every Device will see this data
- This causes problems:
  - They can sniff it with wireshark – not very secure
  - Only one device can send a message at any time – slow network
- The network switch is able to read the fixed hardware (MAC) address of the connected device

Ethernet (hardware) addressing

- The switch can send the information to the single *physical* port that the device is connected to
- To do this there is yet another header
- Take the IP **Packet** and add the ethernet (MAC) header to create an Ethernet **Frame**

Ethernet (hardware) addressing

- The switch sees the frame header and sends the information to only one port (normally)
  - Sometimes more, if it doesn't know what to do
- The receiving device checks the **frame** to confirm its hardware address, then strips the MAC header to reveal the IP address
  - We already know what to do with that!
  - (yes there is a return MAC address, but another day…)

Message Size - Segmenting

- Infinite length messages cannot be sent otherwise there could be only one message sent (ever)
- Long messages are broken into individual pieces
- This is known as segmenting.
- There are minimum and maximum data size requirements for the network
  - Too small means too many packets, busy network
  - Too large means limited user access

Message Size - Segmenting

- Each segment is encapsulated in a separate packet with address information and is sent over the network.
- At the receiving host, the messages are de-encapsulated and reassembled

- Data can be any size but network packets have a maximum size, guess which layer segmentation happens??
  - Another clue, look at the pdu names at each layer!

Message Size - Segmenting

- Segmenting message benefits:
  - Different conversations can be interleaved (do not need for one device to say everything…)
  - Increased reliability of network communications
- Segmenting message disadvantage:
  - Increased level of complexity

Summary

- Network devices must comply with communication rules and protocols.
- The most widely-used networking models are the OSI and TCP/IP models
- Data that passes down the stack of the OSI model is segmented into pieces and encapsulated with addresses
- The recipient recovers the data in the reverse process
- The OSI model describes the processes of encoding, formatting, segmenting, and encapsulating data for transmission over the network

www.derby.ac.uk/engtech

What did we cover?

- Layers concept
- Encapsulation
- Data units
- Logical port numbers
- Logical addressing
- Physical addressing
- Segmenting

Email : j.hardy@derby.ac.uk
visit : MS310
phone : 01332 591732