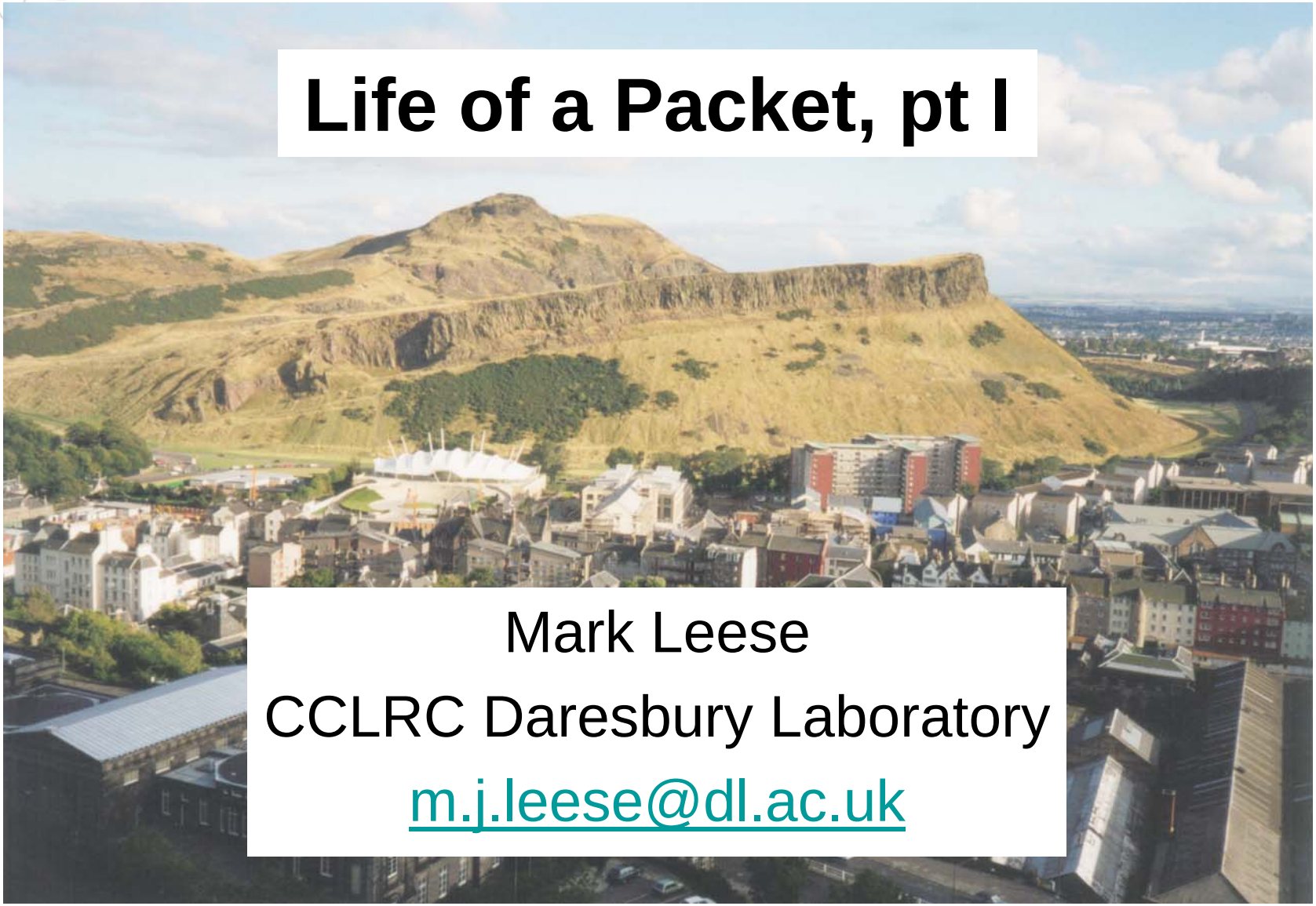


Life of a Packet, pt I

Mark Leese

CCLRC Daresbury Laboratory

m.j.leese@dl.ac.uk

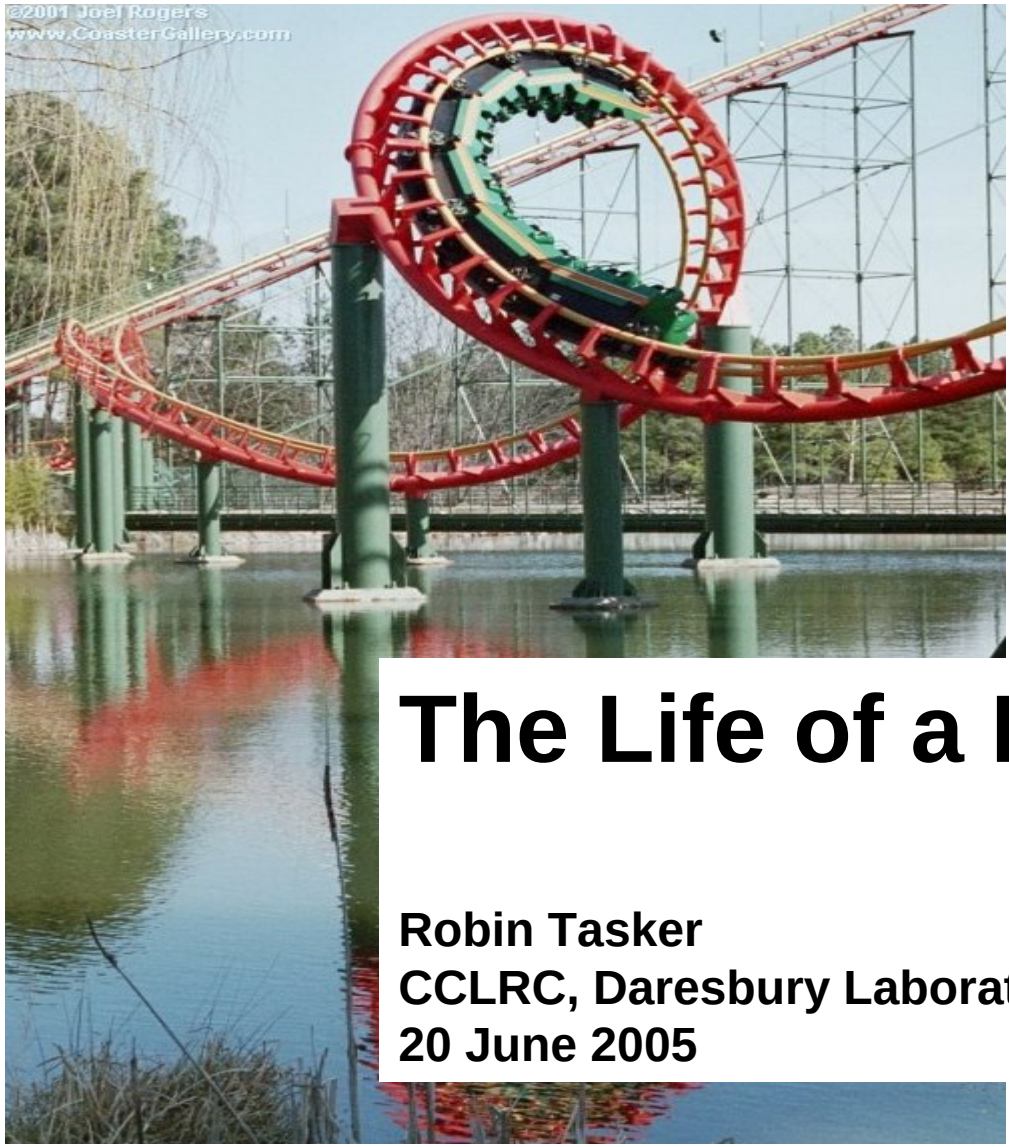
Disclaimers & Contents

Disclaimers

- Agenda showed Robin as giving this talk
- Very unfortunately, he can't be here
- Be patient - this is not my area of expertise, but I like the sound of my own voice

Contents

- Robin's talk: designed to demonstrate how far-reaching/ubiquitous **AND** complicated networks are
- Applause
- Mark's talk: introduction to networks
- Jeering

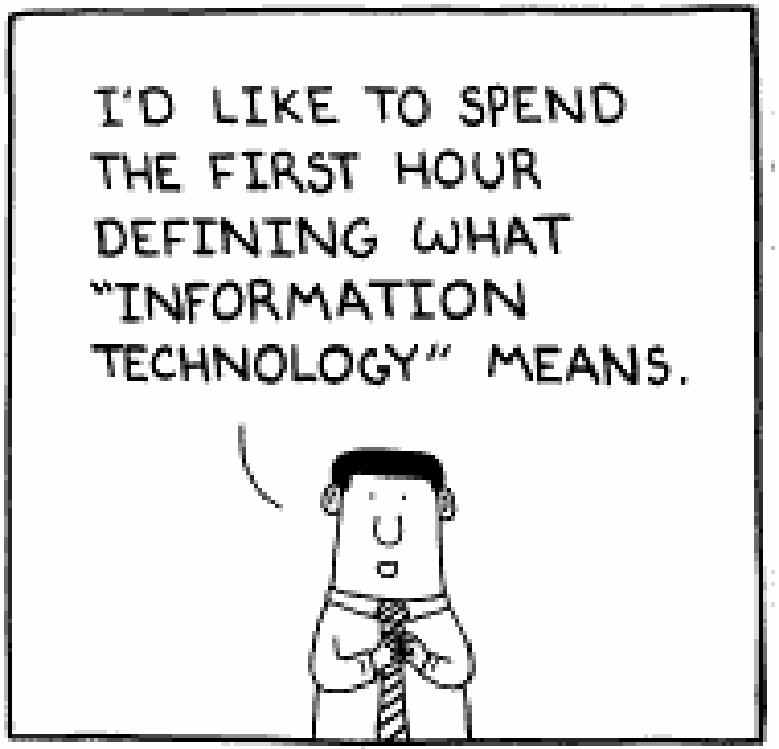


The Life of a Packet

Robin Tasker
CCLRC, Daresbury Laboratory
20 June 2005

CCLRC Daresbury Laboratory

Contents


I'D LIKE TO SPEND THE FIRST HOUR DEFINING WHAT "INFORMATION TECHNOLOGY" MEANS.

Setting the Scene

Q1. A packet is just the means of transferring your data end-to-end across the Internet?

A1. True. So we need to understand how this works; and more importantly why it doesn't always work the way you might expect.
Also need to understand how your data got tangled up with a packet.

Q2. A packet is carried in a series of frames on each of the point-to-point links that together comprise the Internet?

A2. True. So we need to understand just how your packet got itself into a frame and what are the issues making a point-to-point link work for you.
Actually it's why the link doesn't work for you that matters!

Q3. A packet "emerges" at intermediate points across the Internet – routers - and is subject to the vagaries of traffic management and other evil?

A3. True. Just what is "traffic management and other evil" that will scupper your packet's performance, or worse...
That's to say, your data's chances of making it to the other side

It Just Gets Easier and Easier…

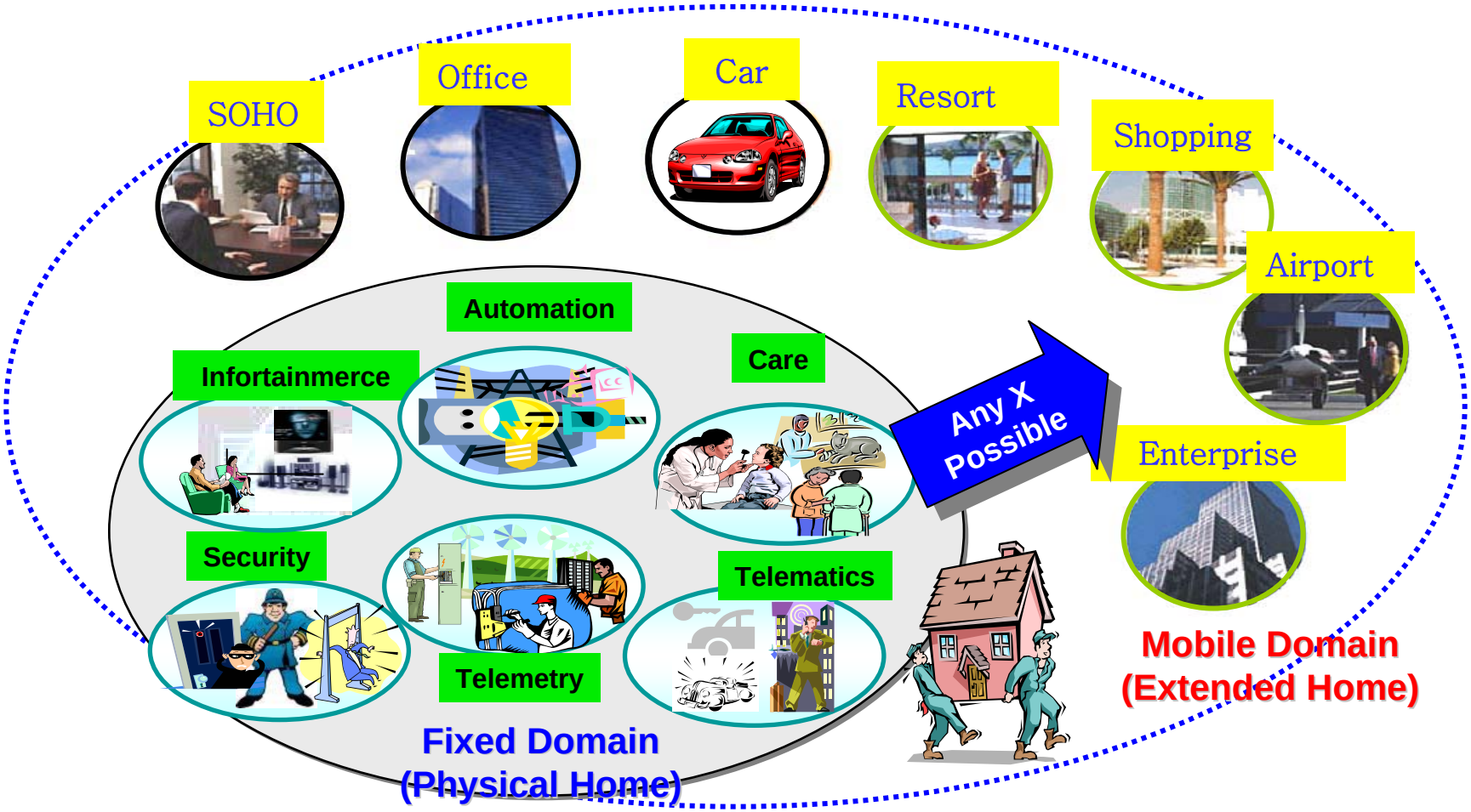
So if you understand the following you're home and dry…

Your Application	*The Attached Network*	*TCP/IP or UDP/IP or ????*
Your Operating System	*Switches or Hubs or Both?*	*Routing*
Your System Hardware	*VLANs*	*NAT*
The TCP Stack	*Other Network Users*	*VPN*
Your System's Kernel	*The LAN Capacity*	*Your Firewall :-)*
Your NIC	*Network Usage Policy?*	*Your Sites Connectivity*
		The Regional Network
Remote "All the Above"	*Remote "All the Above"*	*The Core Network*
		Congestion on the path

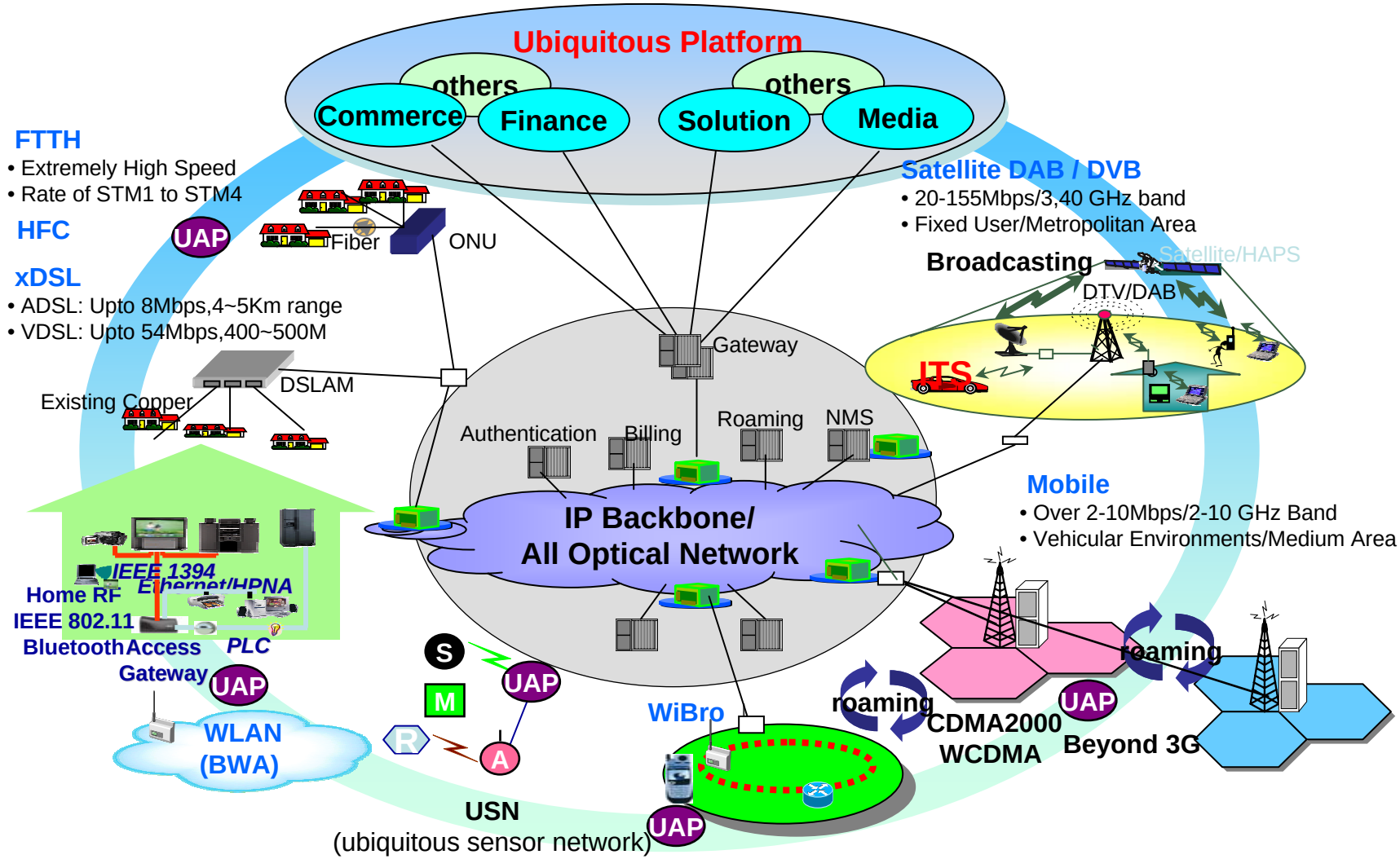
Easy really, but hang on! That's just your end-to-end worries

But isn't there an Internet out there???

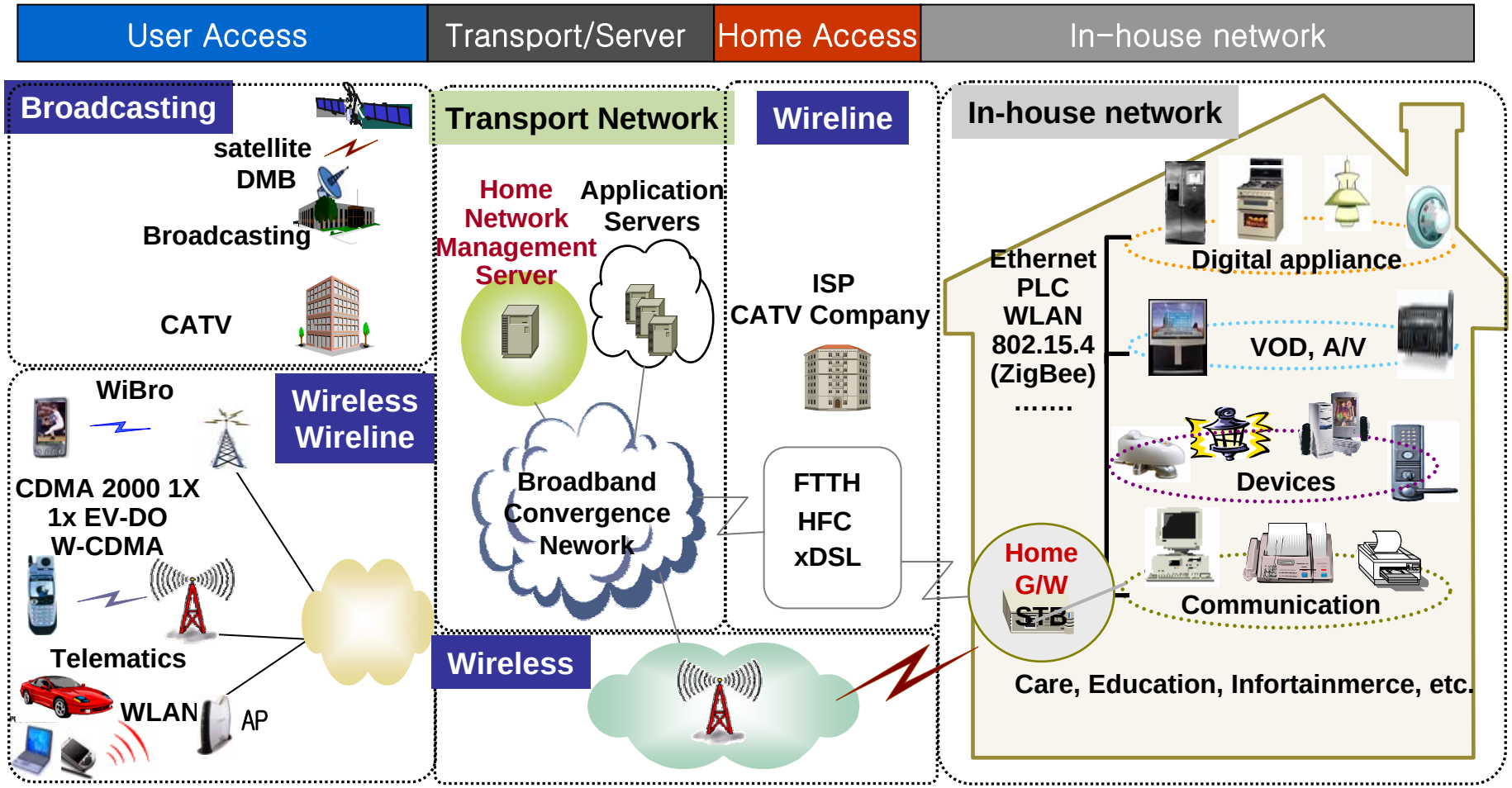
Certainly Is!! "The Big Picture"!



And it gets bigger and more complex!



And there's no hiding at home…



But hey! We're here to help…

Life of a Packet, pt II

Mark Leese
CCLRC Daresbury Laboratory
mindyourownbusiness@dl.ac.uk

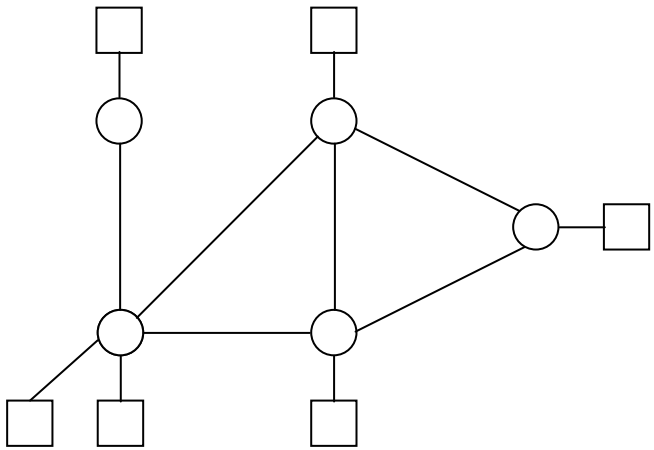
Contents

A basic introduction to networks:

- What is a network?
- Network classification: broadcast bus-based LAN?
- Connectionless vrs connection-orientated services
- Protocol stacks
- TCP and UDP
- Packets
- IP and Ethernet
- And finally: the simplifed life of a packet

What is a network?

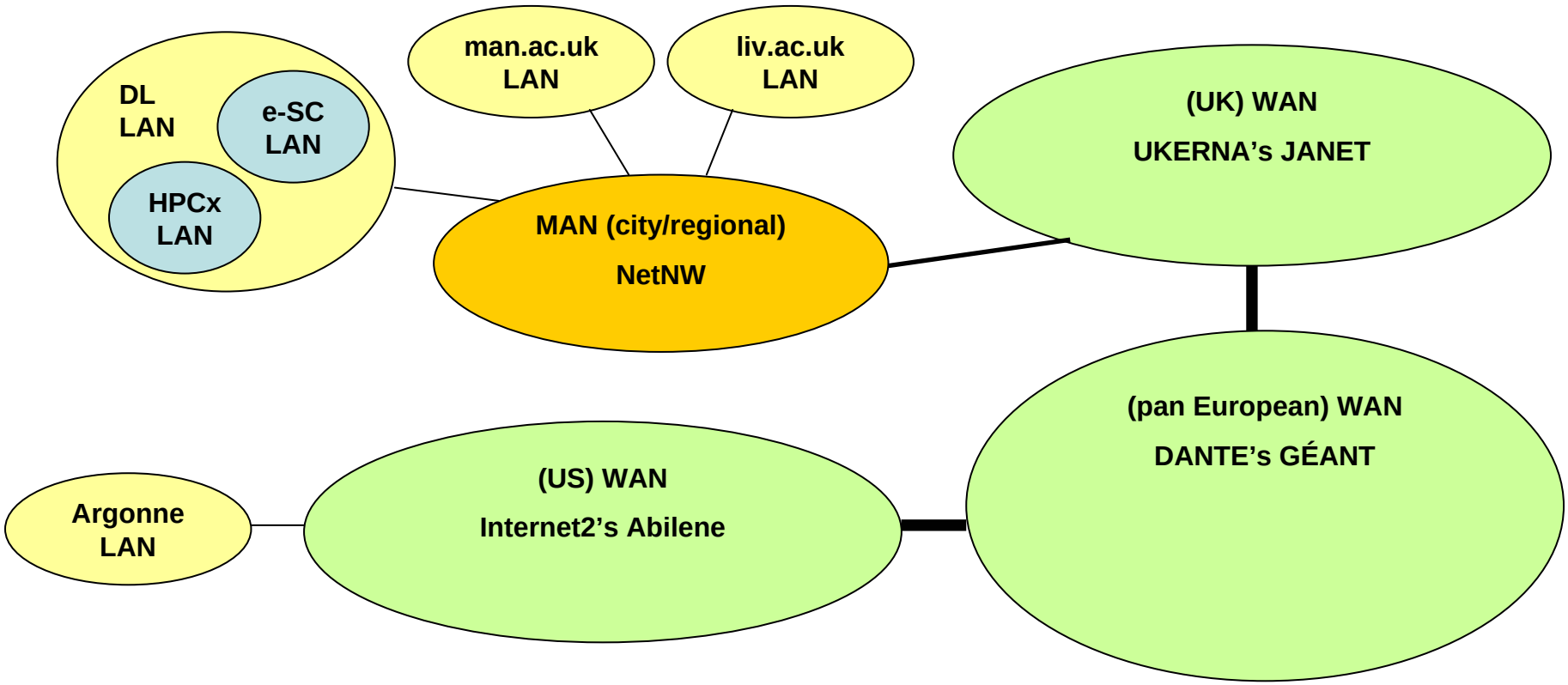
- Lots of definitions. Personally, I like:
  - a set of transmission paths, interconnected at nodes



- In practical terms: an infrastructure that lets you access shared resources (data, processing power *etc.*).

Network Classification 1

- Geographical/scale: LocalAreaNetwork→MetropolitanAN→WideAN

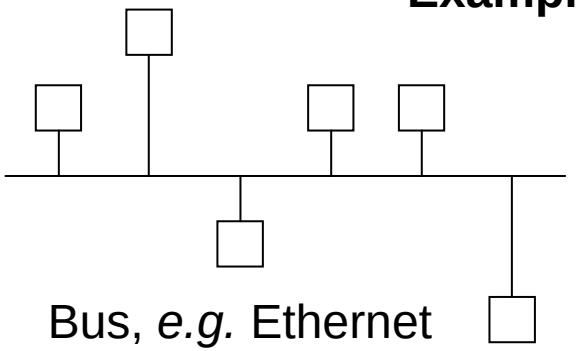


- Generally analagous to road system, e.g. a house to house journey = driveway→Acacia Ave→B5178→A5300→M62→A road→B road→quiet cul-de-sac→driveway

Network Classification 2a

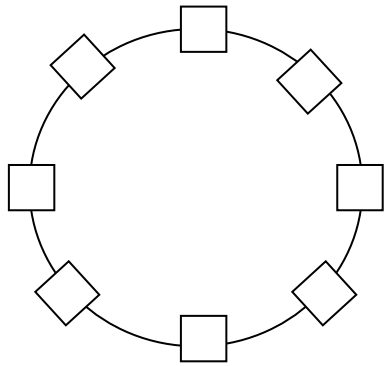
- Transmission type: broadcast or switched
- **Broadcast:**
  - all the machines on the network are connected via a single, continuous communications channel
  - Data transmitted by any of the machines is received by every machine on the network
  - Receiving machines ignore all data unless its destination address matches their own address

Example Topologies



Bus, *e.g.* Ethernet

Ring, *e.g.* Token Ring

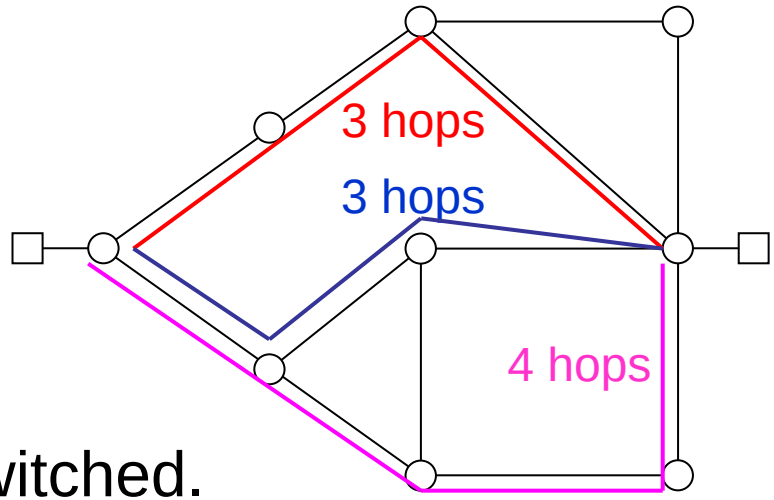


- Cons:
  - Broadcast networks are essentially a shared channel, meaning only one device can transmit at a time, and so only suitable for light traffic loads.
- Pros:
  - routing very simple, in that there isn't any :-)
  - very low wiring costs

Network Classification 2b

- Transmission type: broadcast or switched
- **Switched:** data is **routed** to its destination via a sequence of **point-to-point** (node-to-node) links – so data only seen by destination and nodes along route
- Most common topology = incomplete mesh
  - Each node linked to some, but not all, other nodes
  - Provides multiple routes from the source to destination
    - Efficient routing becomes important, but gives redundancy

Incomplete Mesh



- Two types: circuit and packet-switched.

Network Classification 2c

- **Circuit-Switched:** a dedicated path is established across the network, and exists for the duration of the transmission
- The PSTN (public **switched** telephone network) is a good example
- **Packet-Switched:** your data (whatever it is) is chopped up into small chunks (packets) which are then routed through the network taking the same or different routes, i.e. there is no dedicated path
- Each node will store-and-forward packets
- The Internet is a common example ;-)

Network Classification 2d

Packet-Switched Network Advantage:

- no dedicated connection
  - when you're not transmitting, someone else can
  - packets from different senders (end hosts and nodes) can be interleaved, achieving better utilisation of network capacity than dedicated circuit-switched connections
  - This is an important concept
  - Computer network activity tends to be very bursty
    - On the micro-scale, you might send some data, and then go an access the hard disc. While this is happening, you're tying up the infrastructure that could be used by someone else.
    - On a large scale, you might load a web page, the read it before loading another one. Until then you're not using the network other than for general housekeeping stuff e.g. "Yes, I'm still here".

Packet-Switched Network Disadvantages:

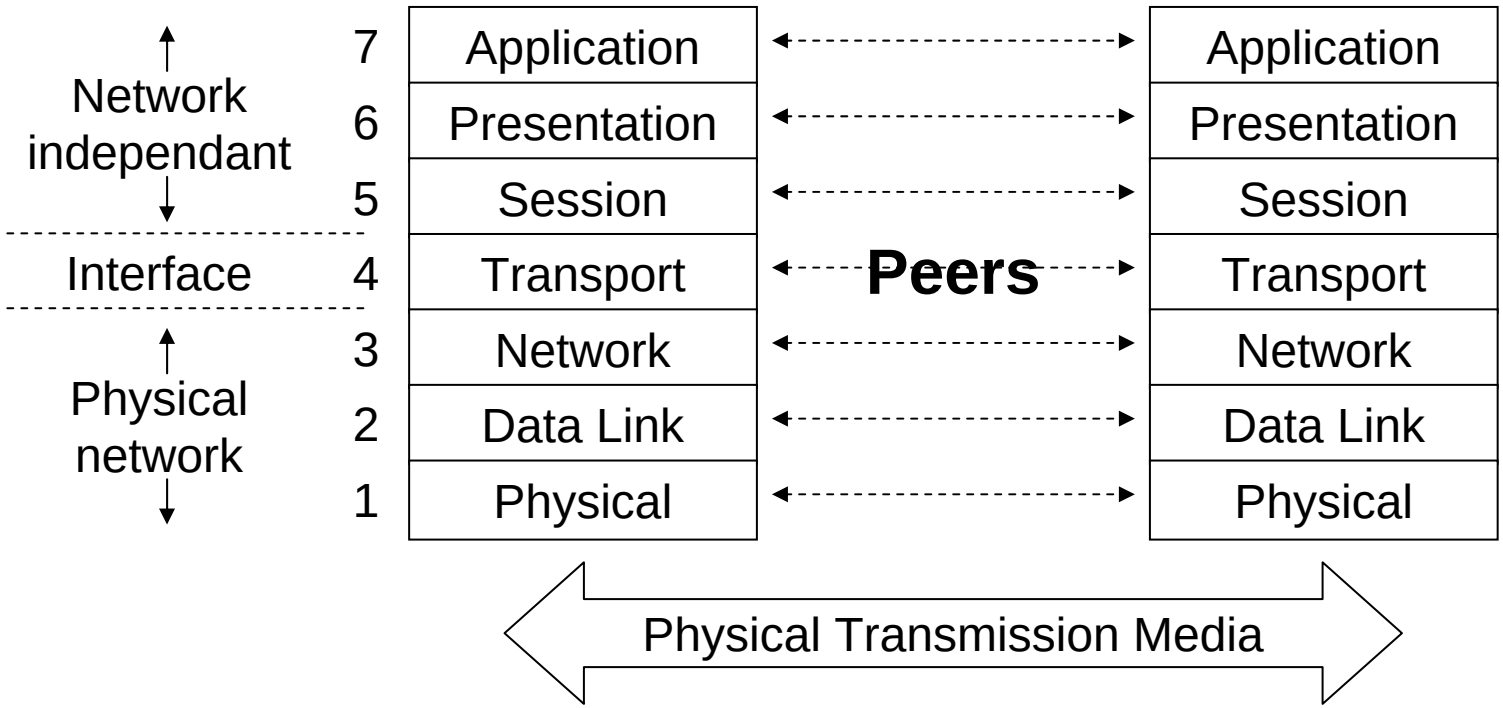
- multiple routes significantly increases the importance of routing strategy
- Chopping data into packets (segmentation and later re-assembly), routing *etc.* all mean networking software is complex
- Delays:
  - **Node delay:** When a packet is received by a node its stored, node then must decide which outgoing link to forward the packet on (based on routing information) then packet must be retrieved for retransmission
  - **Queuing delay:** exacerbated by the need to place packets in queues during periods of heavy loading

Connection Type

- Connectionless or connection-orientated services
- **Connectionless:** packet-switched networks are by their very nature connectionless.
  - there is no direct, dedicated connection between the source and destination
  - each packet travels independantly of any others
- A ***layer*** (see later) making use of a *connectionless service* has to perform its own checks to ensure that packets arrived, and arrive correctly.
- **Connection-orientated:** A ***layer*** can also offer a connection-oriented (aka virtual circuit) service
  - the layer implements a **virtual** direct connection (like a phone call)
  - must include code to check that transmitted packets have arrived (and correctly), resends lost packets *etc.*
- The TCP/IP protocol (again, see later) offers both

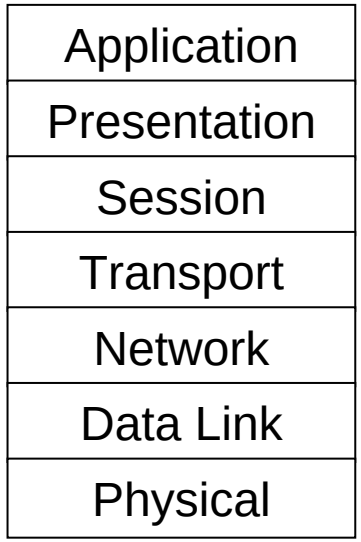
Protocol Stacks 1

- The ISO OSI (Open Systems Interconnection) Seven Layer Model was developed in the 80's to as a well-defined structure against which real network protocols could be developed



Protocol Stacks 2

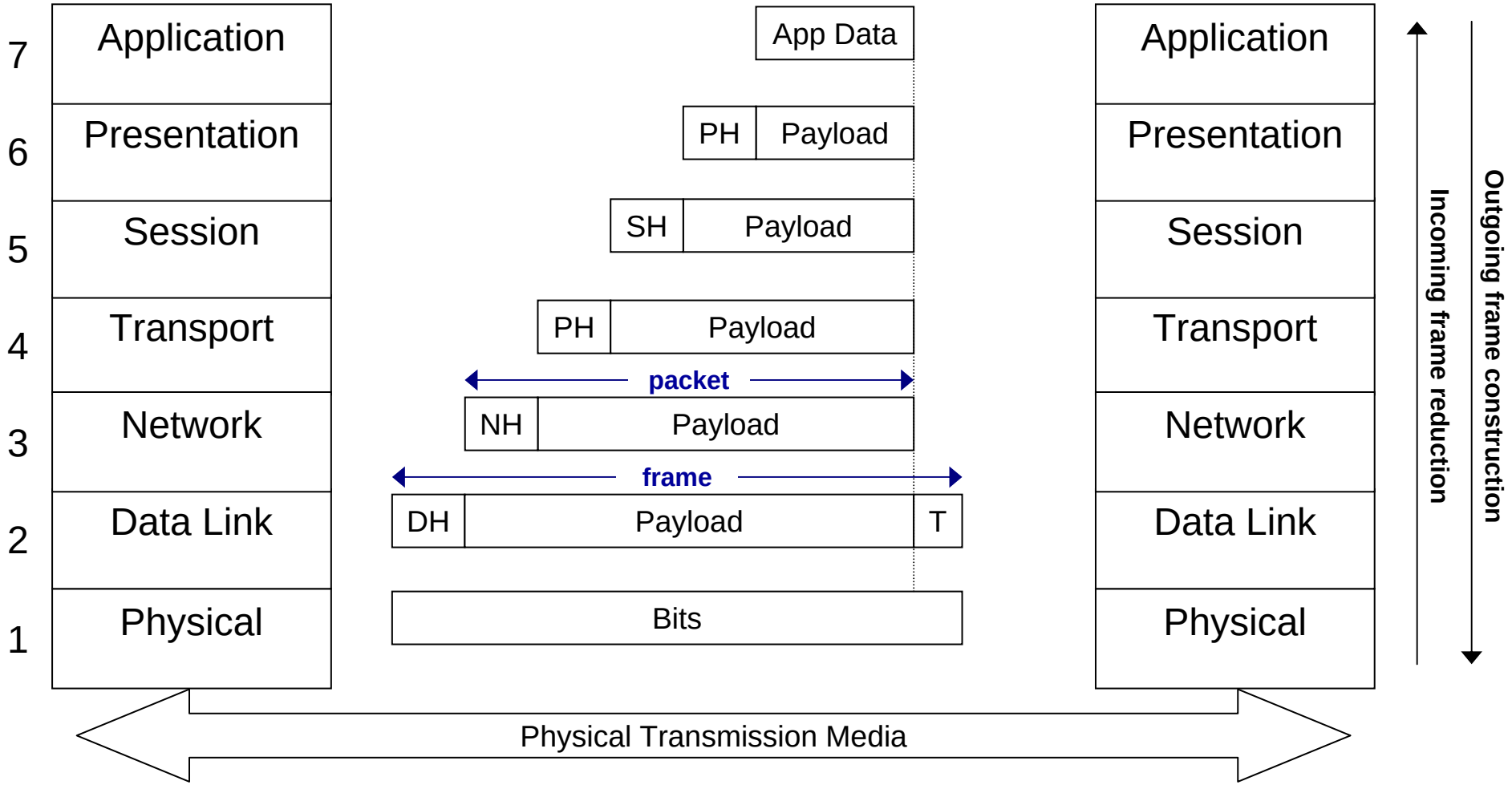
- Layering is key. A layer…
  - performs unique and specific tasks
  - only has knowledge of those layers immediately above and below
  - uses services of layer below, and provides servics to layer above
  - the services defined by a layer are implementation independent - it's a definition of how things work
  - conceptually communicates with its peer in the remote system



- **Transport Layer:** acts as a go-between for the user and network
  - Provides end-to-end control, with the level or reliability need for the application
  - Can ensures a reliable service (which network layer cannot), e.g. assigns sequence numbers to identify "lost" packets
- **Network Layer:** deals with the transmission of packets, inc. routing.
- **Data Link Layer:** provides the synchronisation and error control for the data transmitted over the physical link (ensures correct delivery of frames)
  - **Going down:** fits *packets* from the network layer above into *frames*.
  - **Going up:** Groups *bits* from the physical layer into *frames.*
- **Physical Layer:** concerned with the transmission of individual bits.

Protocol Stacks 3

- **Encapsulation:** data from layer n becomes payload of data at later n-1
- H = header, T = trailer



TCP and UDP

- TCP = reliable, connection-orientated service <<<< Brian

	OSI	TCP/IP
7	Application	Application
6	Presentation	Not present in TCP/IP model
5	Session	
4	Transport	Transport
3	Network	Internet
2	Data Link	Host-to-network
1	Physical	

- UDP = connectionless service
  - User Datagram (next slide) Protocol
  - essentially, fire and forget
  - used where guaranteed delivery and retransmission are not required, *e.g.* real-time apps like Video Conferencing – if some packets are lost, there is no point re-transmitting them – by the time they arrive the VC is two seconds further on

Packets and IP

Read this later:

- Packets are just the unit of data transmitted from a source to a destination on a packet-switched network
- Misconception that packets are an IP thing. IP packets **do** exist, but aren't the only ones, *e.g.* before TCP/IP JANET was a packet switched network using X.25

- **a datagram** is "a self-contained, independent entity of data carrying sufficient information to be routed from the source to the destination computer without reliance on earlier exchanges between this source and destination computer and the transporting network." [RFC 1594]
- Datagrams need to be self-contained without reliance on earlier exchanges because they are used by connectionless services
- The ubiquitous nature of the Internet means that datagram and packet are now largely interchangable terms for the message units dealt with by the IP protocol (layer 3) and transported by the Internet.

Packet Size

Read this later:

- Packets can vary in size.
- And there are various transmission technologies: ATM, Ethernet, GPRS, PoS (Packet on SONET) *etc.* that can all have different frame sizes.

- **Q:** So how big should a packet be?
  - Should it be small so it takes a very little time to send
  - Or should it be large so that it's efficient: the ratio of header and trailer to payload is small?

- **A:** Depends really on the transmission protocols.
  - If the packet you send down the stack is too big, then the IP layer will break it into smaller pieces called fragments.
  - MTU (Maximum Transmission Unit) discovery allows you to adjust packet size to what the path can support.

IP and Ethernet 1

- IP is the network (layer 3) protocol that supports TCP and UDP (layer 4 protocols)
  - Below IP we have different data link (layer 2) protocols for actually moving the IP data around.
  - We have different protocols because each was designed with a specific purpose in mind, *e.g.* we might not want the same features/characteristics when transporting data around a LAN as in a WAN.
  - The network layer (and hence IP) is responsible for routing, essentially because it makes routing independent of the data link protocols. .
- IP devices (end hosts, nodes *etc.*) are identified by IP addresses:
  - a “dotted quad”, e.g. 193.62.119.20
  - easier to remember are the textual domain name (*e.g.* gridmon.dl.ac.uk) associated with IP addresses
- In the LAN, Ethernet is the prevalent data link technology
  - It has it’s own addressing, using MAC (Media Access Control) or physical addresses (next slide)
- Out in the WAN we may use SDH (Synchronous Digital Hierarchy) or SONET (the N.American equivalent) as the data link protocol

IP and Ethernet 2

- Remember, IP addresses are dotted quads, e.g. 193.62.119.20
  - With the exception of some addresses which are reserved for special purposes, each part of th quad can range from 0→255.
  - The first so many digits identify the network to which a particular address belongs
  - IP addresses are logical ones – assigned by sysadmins or DHCP
- An Ethernet address (a MAC/physical address) comprises 12 hexadecimal digits
- E.g. 01-2E-2D-59-BB-6B
  - Hard-coded into the device
  - Assigned by the manufacturer – each is allocated a certain block of addresses that they can use.

- **Q:** "Why do we use two addresses. Why can't we use the IP address OR the Ethernet address?"
- **A:** Two reasons:
  - One:
    - IP addresses are routable. Routers can use the network portion of the address to make routing decisions.
    - Ethernet addresses are not routable (unless every router knows how to reach every Ethernet address in the world) because they don't have the same concept of belonging to a particular network.
    - Two PCs in your network can have similar IP addresses (in the same network) but if their network cards are made by 3Com and Intel, they will have wildly different MAC addresses.
  - Two:
    - What is we're not using Ethernet as the data link (layer 2) protocol?
    - What if we're using ATM, which has a 6 digit address (VPI + VCI)?
    - The logical addressing (IP) should be independent of the physical addressing (Ethernet, Token Ring, ATM, SDH *etc.*) meaning you don't have to worry about what the underlying transmission technology is.

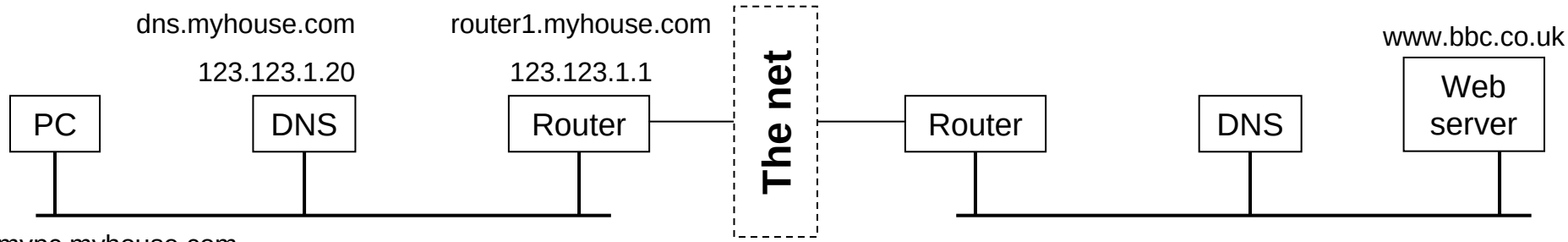
- ARP (Address Resolution Protocol) is the service used to convert IP addresses into physical ones
  - ARP request: broadcast the message "Who's got this IP address?"
  - Response should come back: "I have. My physical address is ab-cd-ef-gh-ij-kl"

Life of a Packet Example, pt 1

And finally an example to try and tie it all together:

- PC puts IP packet into Ethernet frame
- Site router will extract IP packet from Ethernet frame
- Maybe then put IP packet into SDH frame to go out into the big wide world
- Everytime your data reaches a router, your IP packet will be extracted from whatever frame it's in, analysed, and discarded or forwarded on in a new frame

UP AND DOWN THE PROTOCOL STACK



mypc.myhouse.com

Addr = 123.123.1.50

DNS = 123.123.1.20

Default Gateway = 123.123.1.1

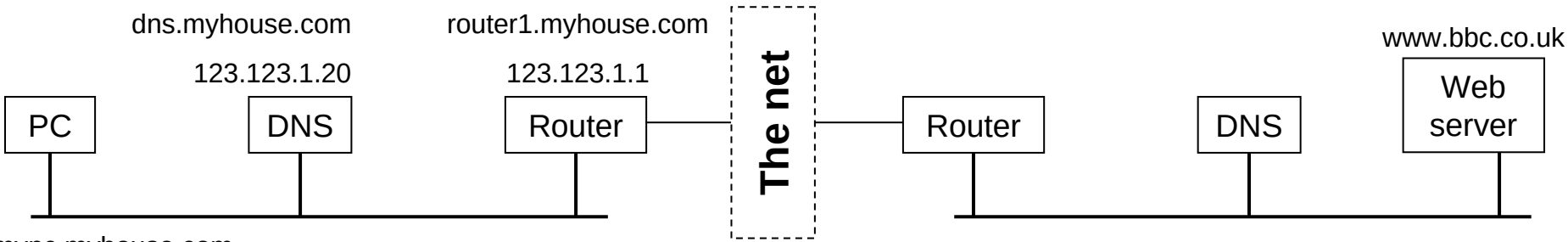
1. DNS query for www.bbc.co.uk
2. ARP for 123.123.1.20 (i.e. DNS); DNS machine replies with its MAC addr
3. DNS query sent to 123.123.1.20
4. IP addr of www.bbc.co.uk returned
5. mypc's routing says send to default gateway
6. ARP for 123.123.1.1 (i.e. router); router replies with its MAC addr
7. TCP conn request sent to www.bbc.co.uk via 123.123.1.1

Life of a Packet Example, pt 2

Routing:

- SysAdmins will have configured the default gateway (the default router) for your PC to talk to.
- Once your traffic reaches the default router, it must decide where the next step is, and so on.

- **Q:** How do routers know where to send packets to? How do routers know how to get to all the other networks in the world? The internate
- **A:** the Internet is a distributed, de-centralised organisation of routers which talk to each other all the time, buidling up a view/picture of the world



mypc.myhouse.com

Addr = 123.123.1.50

DNS = 123.123.1.20

Default Gateway = 123.123.1.1

Conclusion

I only want you to take two things forward:

- Most problems are not in the core of the network
  - If your application performance is poor, and IF the problem **really** is the network, 9 times of out 10 it will be your LAN
- Simple over provisioning is not enough
  - Throwing money at extra bandwidth is not the answer
  - A complete approach must be used:
    - better app design
    - better end systems and end system configuration
    - better LAN design
    - and so on…