Lancaster University

SCC203 Computer Networks: Introduction

Dr Andreas Mauthe (a.mauthe@lancaster.ac.uk)
Dr Andrew Scott (a.scott@lancaster.ac.uk)

Lancaster University

Course Aims

- Provide students with a detailed understanding of data transmission techniques in computer networks
  - Introduce students to the **key features of network protocols**, and the steps and relevant protocols needed for establishing and maintaining reliable data communication between network endpoints
  - Enable students to **understand** the key **building blocks** of the **Internet**
  - Familiarise students with **practical aspects** of networks and network elements required to support network communication, from both a network protocol and application level perspective

- The module aims to enable you to:
  - Critically evaluate technical ideas and mechanisms in computer networking
  - Read and follow detailed technical documentation
  - Learn about design and implementation principles
  - Enhance your independent research skills
  - Solve network related problems and learn techniques relevant in this context

How this course is taught

Weeks	Mon. 9:00-10:00, 15:00-16:00	Tue. 14:00-15:00, Thu, 11:00-12:00	Lab (B76) (2 hours)
11	Introduction into Networking	Reference Models	X
12	Physical & Data Link Layer	LAN	Introduction to Lab Infrastructure
13	Internet Protocol (IP)	ICMP, ARP	Host Configuration, Server Communication
14	Transport Layer Basics, UDP	Naming & Addressing	Subnets & Subnet Configuration
15	Internet Forwarding Routing	Routing	Switch Configuration
16	RIP	Reliable Transport	Daisy Chaining Routers
17	Applications	Home Networks	LAB Test
18	Domain Name Service (DNS)	IPv6	VLAN Router Configuration
19	TCP	TCP (cont.)	LAB Coursework
20	Advance Routing (OSPF & BGP)	Revision Lecture	Marking Session

How this course is assessed

Title	Deadline Wk.	Weight %	Feedback Wk.	Feedback
Individual	17	20	19	Moodle
Group	19	20	20	In Lab
Exam		60		

- Coursework is submitted online and checked for plagiarism automatically.

Feedback

- How is feedback provided?
  - **Moodle Test:** Feedback in class and through moodle by providing expected answer content after marking
  - **Group CW:**
    - **Overall/class:** by the academics in lectures (and/or via the class announcements). We will give updates on how the class is performing overall, and any common mistakes/areas to improve.
    - **Detailed/personal:** by the academics and TAs in the lab sessions. We will mark your work and provide on-the-spot verbal feedback.

Attendance Check-in using iLancaster

Lancaster University

- Ensure you have the most up-to-date version of iLancaster
  - Then, to check in automatically:
    - **Turn Bluetooth** and **Location Services** on for your device.
    - If you want to know when you're checked in, **turn Notifications on** for your device.

Lancaster University

What is plagiarism

- Passing off someone else's work as your own, including:
  - Submitting (e.g.) code that someone else wrote
  - Paying for someone else to do it for you
  - Working on one piece of non-group work together as a group, and submitting it as individual work
  - Sharing of code that you then possibly adapt/ obfuscate

What we expect from you

- **Integrity** (no plagiarism, no faking results) and **effort** (active learning):
  - i.e. come to lectures, even early ones (it helps!)
  - go to labs (these are compulsory!)
  - get the textbook (if there is one) and use our/the world's resources effectively
  - take notes (evidence hand written are better!)
  - read around the subject/try things for yourself
  - ask us questions in lectures/labs
  - take notes (again, because the slides are not enough when you try to revise, really…!)
  - plan your time and coursework carefully

Lancaster University

What you can expect from us

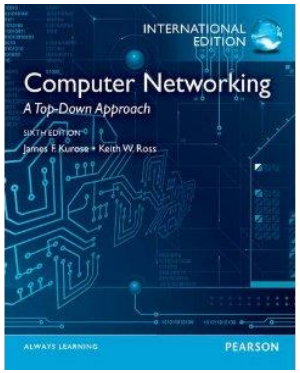
– We'll do our best
  - to make all our lecture notes available on moodle
  - to give you references to follow up
  - to personally check the labs are running smoothly and the TAs are offering support
  - to arrange extra support if you've already tried the normal routes (books, web, forum, TAs)
  - to offer feedback on formative coursework promptly
  - to respond to email (ideally as a last resort! - note: we get more email than we can handle, and have a lot of teaching/research/admin commitments, so are often not in our offices!)

Lancaster University

How do I get help?

- Please use the labs to ask for help
- Please use the course forum on moodle:
- Set up a meeting via *email*,
  - we expect you to have a) tried, b) tried the literature, c) looked for solutions yourself, d) asked the TAs first...

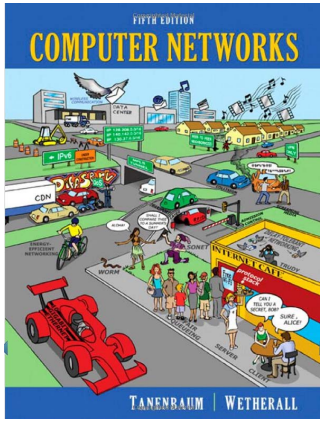
Recommended Textbooks



James F. Kurose, Keith W. Ross:

Computer Networking: A Top-down Approach (Sixth Edition)

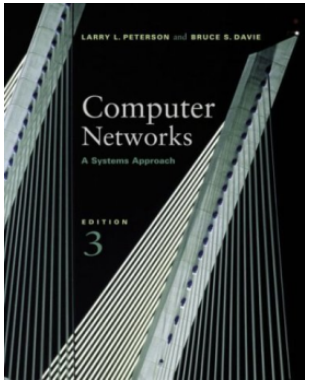
Pearson, 3 May 2012, Hardback/ Paperback



Andrew S. Tanenbaum:

Computer Networks

5.th edition, Prentice Hall, 2011



Larry L. Peterson, Bruce S. Davie:

Computer Networks: A Systems Approach (5th Edition)

Morgan Kaufmann, 20 April 2011, Hardback/ Paperback/ Kindle: 920 pages

Today's Lecture

Lancaster University

- Outline
  - Short History of Networking
    - Telegraphy, Telephony, Internet
  - Basic Networking Concepts
    - Network structure
    - Time Sequence Diagrams
    - Protocols
    - Addressing
  - Communication Principles
    - Communication states
    - Packets
    - Network types

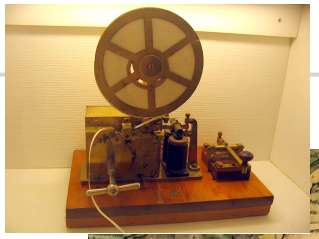
- Objectives
  - To get familiar communication abstractions and the motivation behind
    - You should be know about network elements, layering, reference models and their purposes in communication
  - To learn about the basic principles underpinning networking and communication
    - You should understand some of the fundamental concepts used in networking and communication systems

Lancaster University

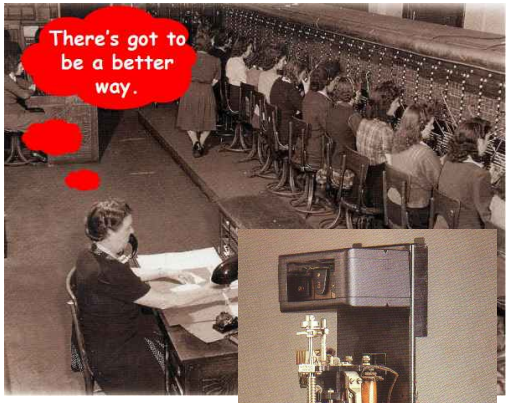
What are communication networks and why are they important?

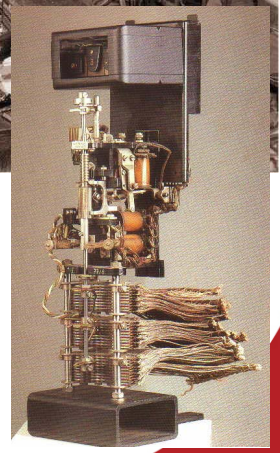
A little History of Networking

How did it all Start



- Telegraphy
  - ~1750: first experiments with electrostatic telegraphs
  - 1844: first Morse telegraph line between Washington and Baltimore
  - 1866: first operational transatlantic telegraph cable





- Telephony
  - ~1860: first successful electronic sound (nearly voice) transmission
  - 1877: first manually switched phone exchange in US
  - 1892: first automatic telephone exchange patented in US (Strowger)

Telegraphy vs. Telephony

Lancaster University

- Telegraphy
  - Message switching
    - Telegram as discrete unit forwarded from sender to receiver via relay stations

    ➔ no dedicated line between sender and receiver
  - **Connectionless**
    - Subsequent telegrams can use different line
- Telephony
  - Circuit Switching
    - Dedicated line between sender and receiver
  - **Connection-oriented** service

The Internet

Lancaster University

- History
  - 1961: Packet switching theory described by Kleinrock at MIT
  - 1965: First computer network by Roberts and Marill
    - Through dial-up telephone line
  - 1967: ARPANET concept published by Roberts
  - 1970-1972: First communication protocol implemented in ARPANET
    - Network Control Protocol (NCP)
  - 1973: Initial TCP/IP idea presented by Cerf and Kahn at Stanford and BBN
  - 1983: Cutover from NCP to TCP/IP in ARPANET
  - ~1990 World Wide Web
- Who defines the Internet
  - Internet Engineering Task Force (IETF)
  - Internet Research Task Force (IRTF)
  - Internet Architecture Board (IAB)
    - Coordinates task forces
  - World Wide Web Consortium (W3C)
- How is it being defined
  - Request For Comments (RFCs)
    - http://www.rfc-editor.org/rfc-index.html

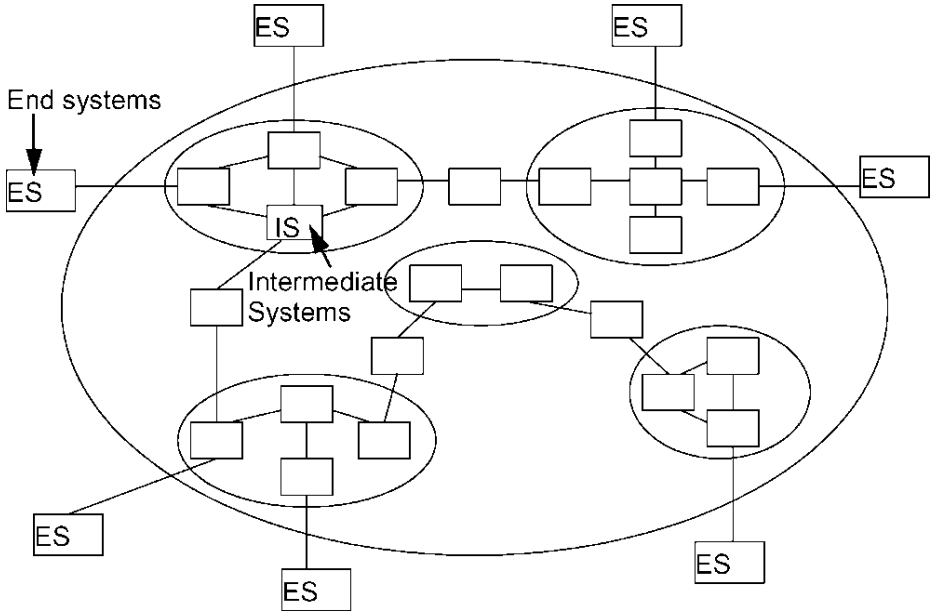
Lancaster University

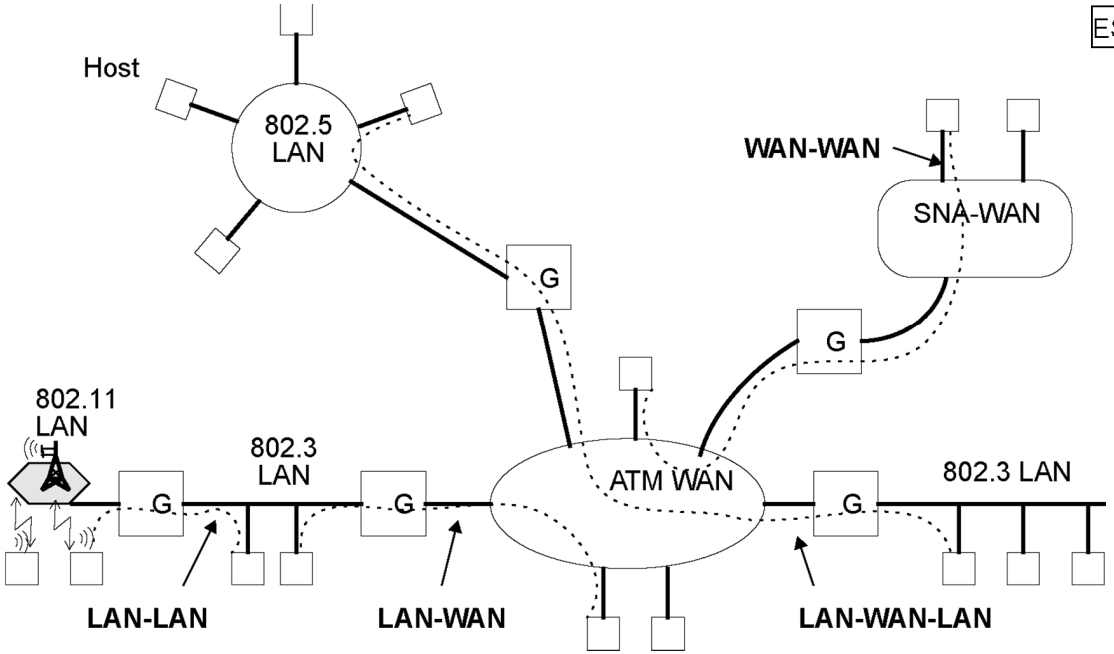
What does a Network look like?

Network Structures

Network Elements

- Networks
  - Cables (or wireless)
  - Routers & Switches to forward messages
  - Intermediate Systems (IS) to connect networks
- End-Systems

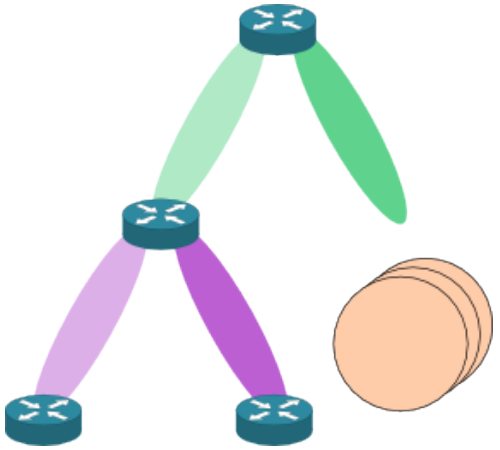


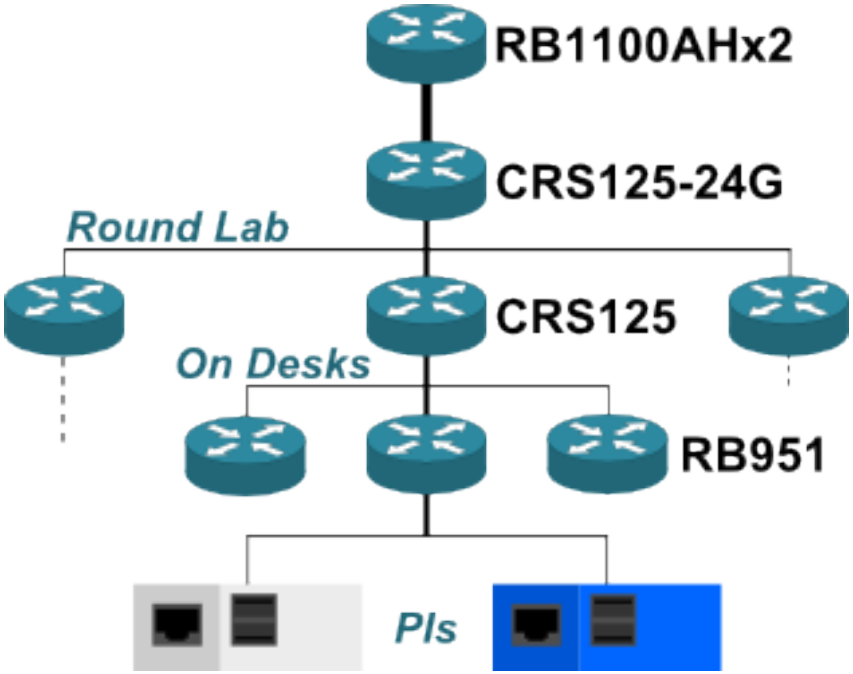


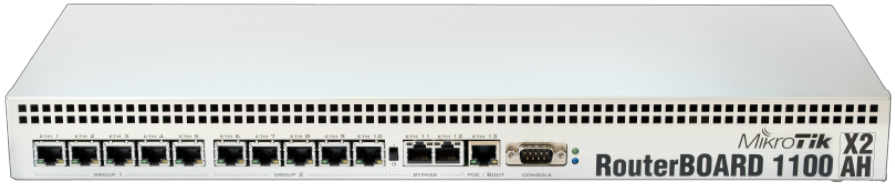
- Network Types
  - Local Area Networks (LAN)
    - E.g. Ethernet (802.3), WLAN (802.11)
  - Wide Are Networks (WAN)
    - Provide the backbone
    - Operated by major providers

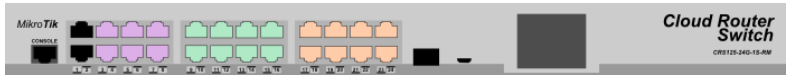
Lab Network

- Logical Network Hierarchy
  - Routers connected through network cables
  - Top router is central element in Comms room
  - Middle router is CRS125 Router Switch at the end of each row of desks
  - Bottom routers are RB951 in the individual network stations









Lancaster University

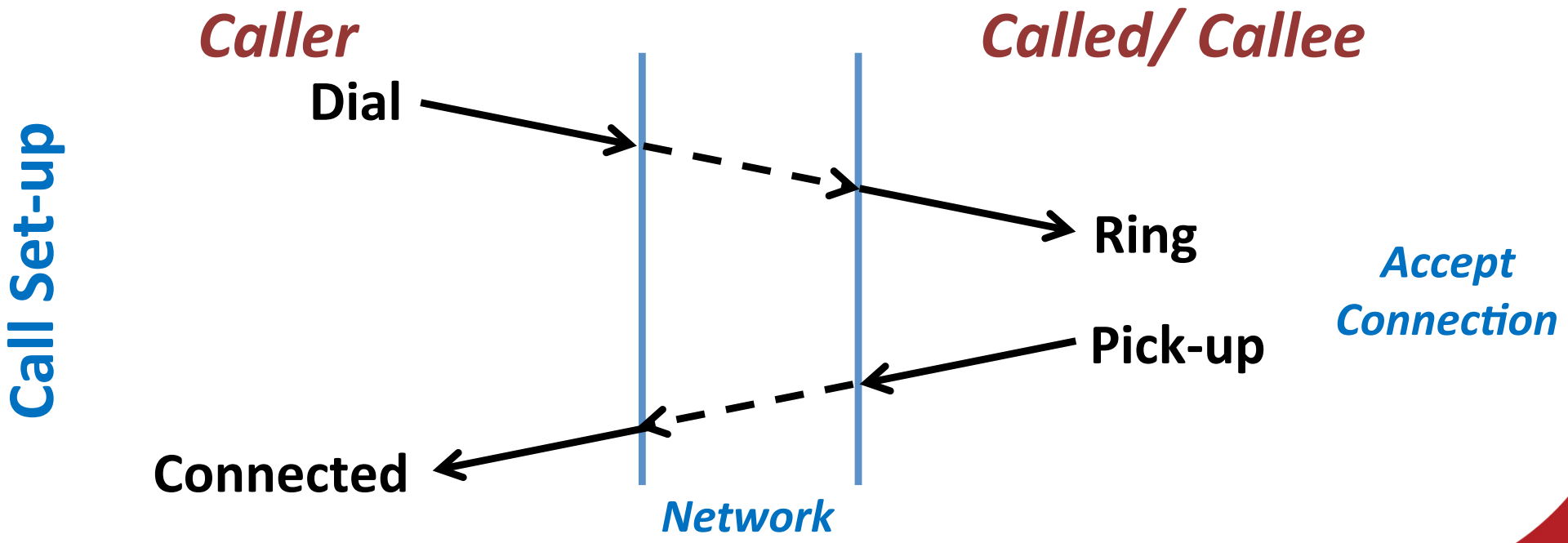
How does Communication work?

Switching, Phases of Communication, etc.

Circuit Switched Systems

Lancaster University

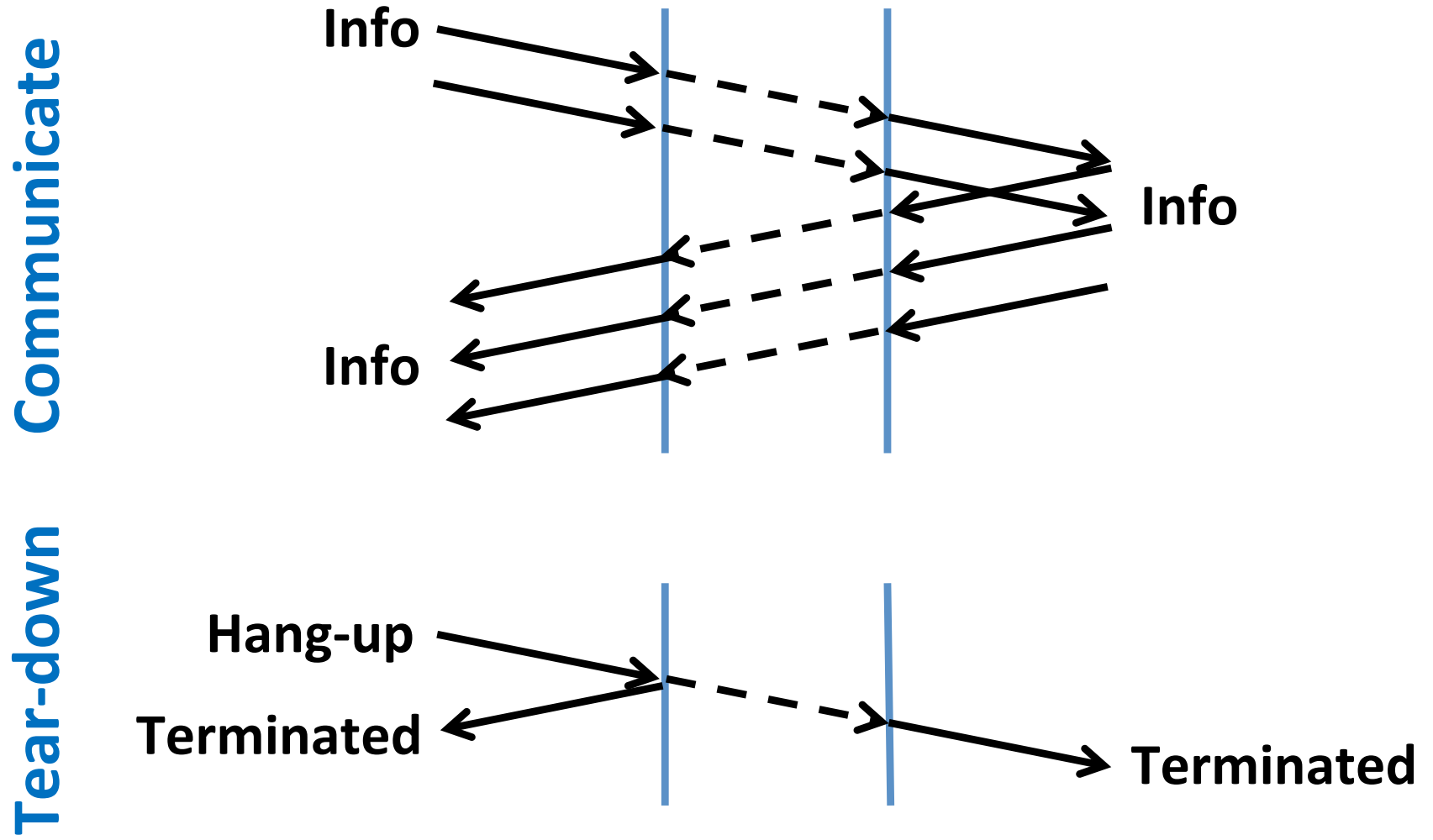
- Connection has three distinct phases
  - Call set-up, communicate, tear-down
- In the case of a phone,
  dialling initiates a connection to remote phone



Remaining Phases

Lancaster University

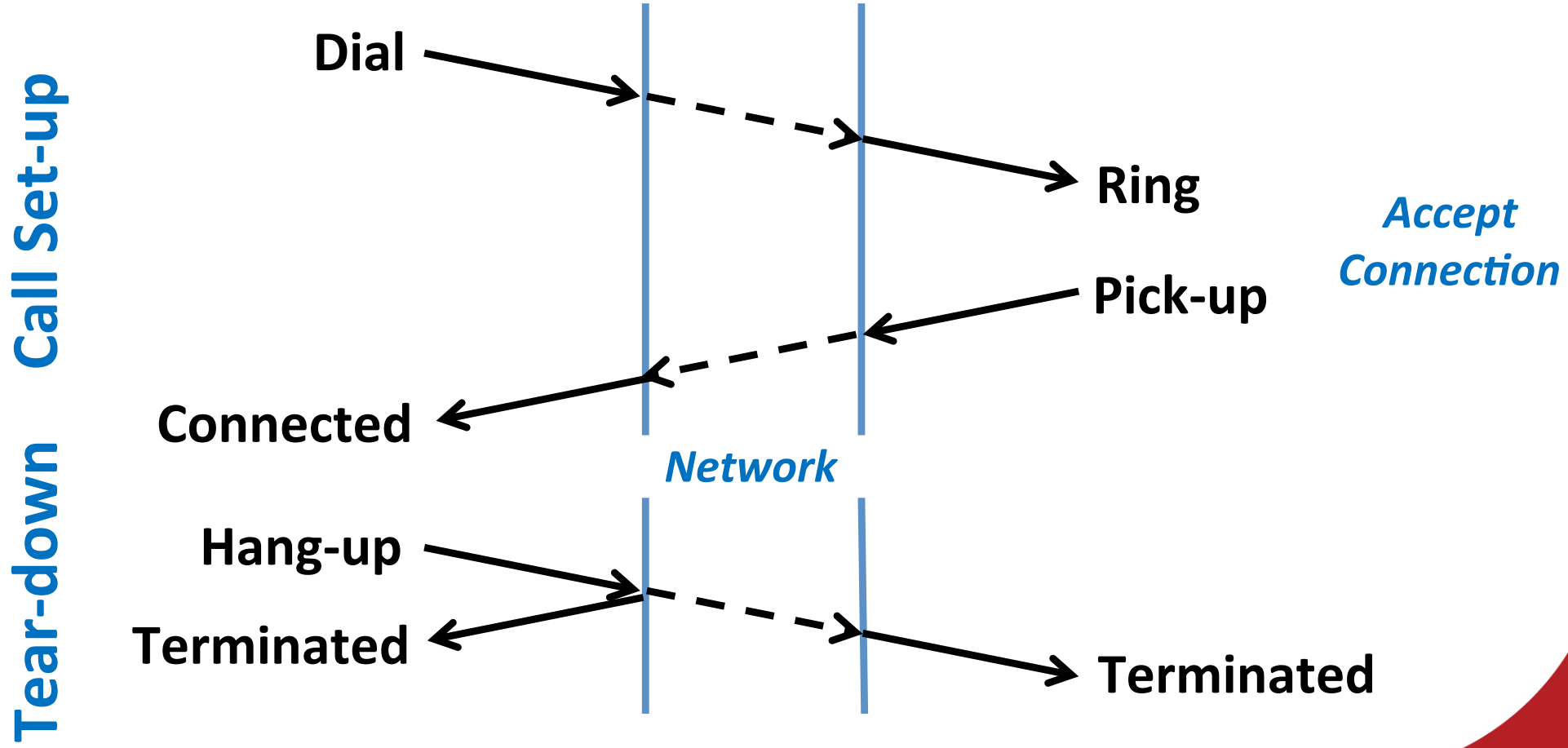
- We can show the other phases in the same way



Time-Sequence Diagrams

Lancaster University

- Show ordering of events
- What should respond to each event and how



Common Alternatives to Time Sequence Diagrams

Lancaster University

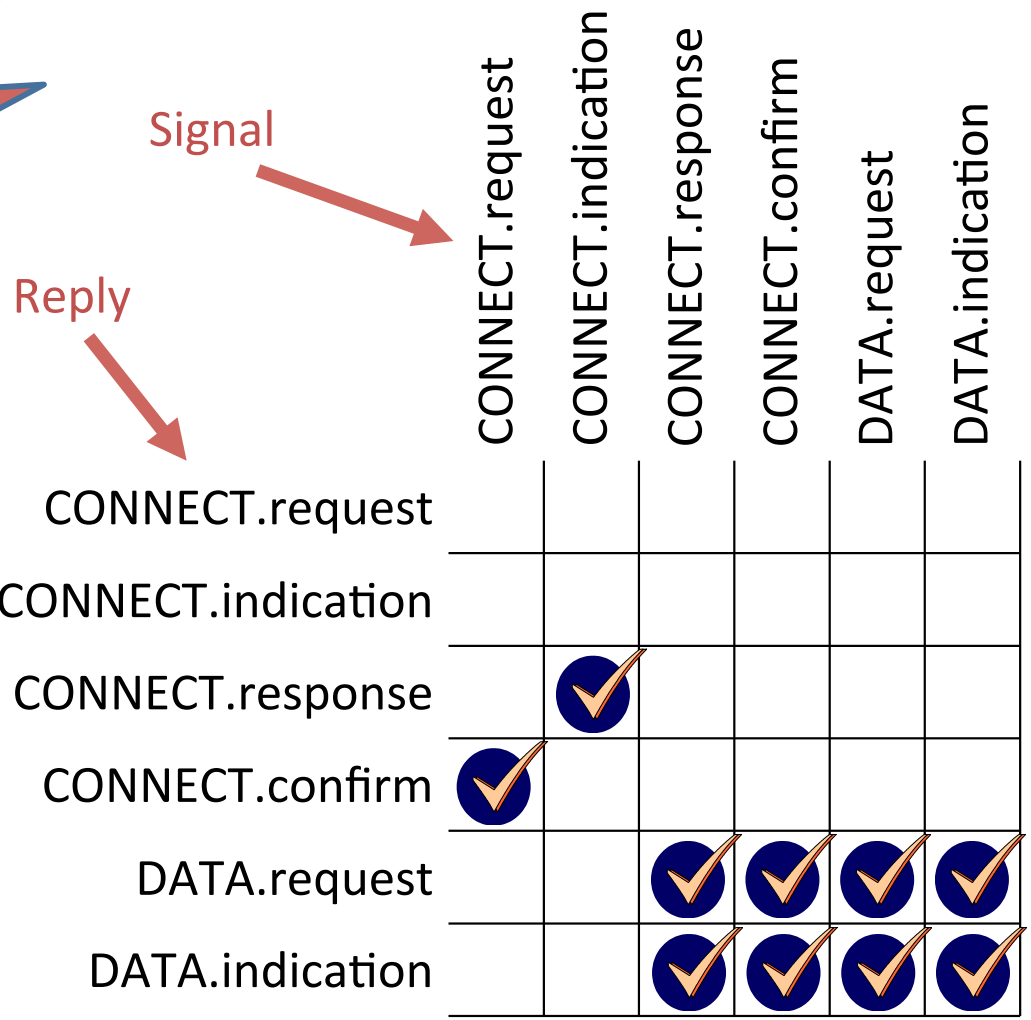
State transition tables

State machines

State Transition Tables

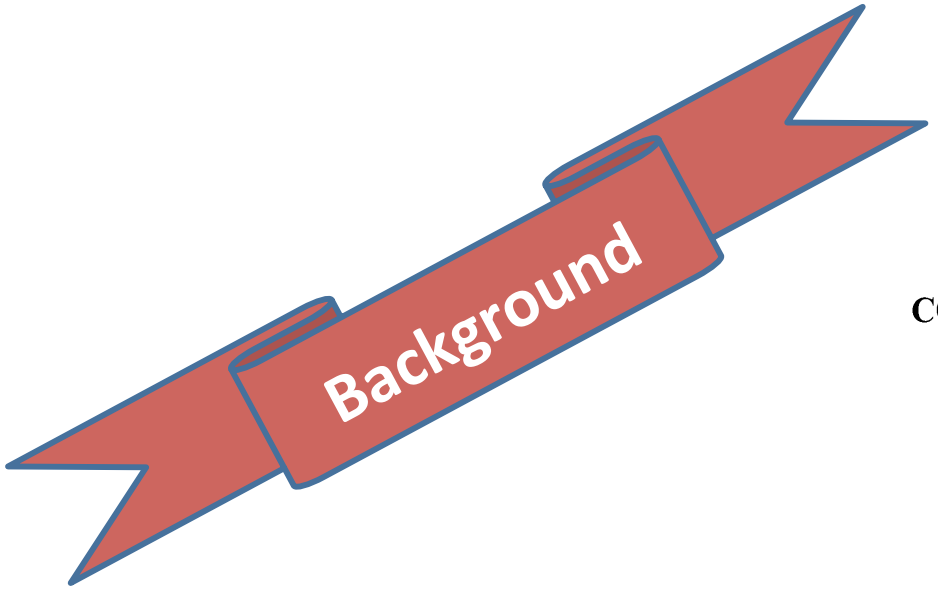
Lancaster University

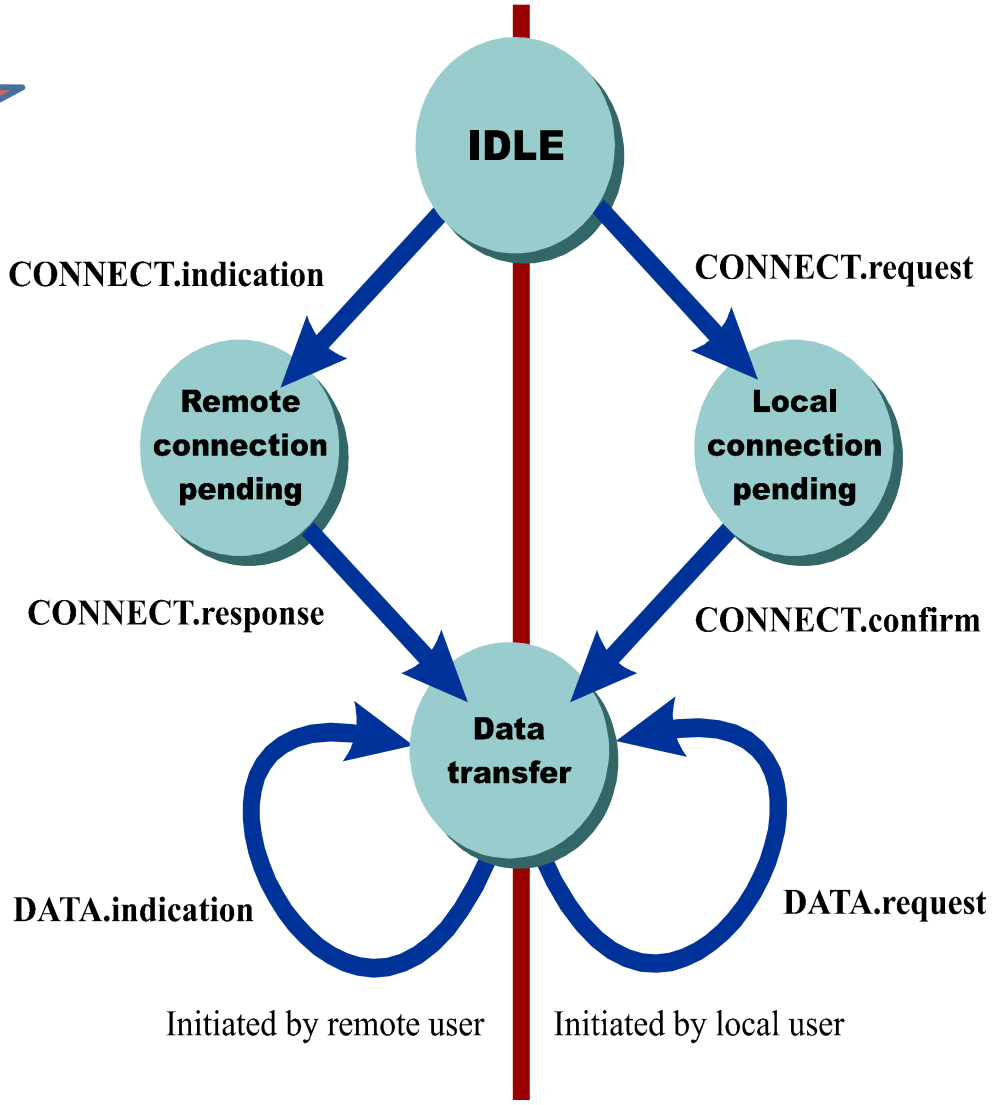
Background



State Machines

Lancaster University





Protocols

Lancaster University

- Time-Sequence diagrams one way of specifying a protocol
- Protocol is the specification of:
  - ALL possible events and actions in system
  - Expected/ required response to every action or event
  - Exact detail of every message sent
- Remember dealing with independent systems
  - Probably using different implementations
  - Protocol must be complete and unambiguous
    - Seemingly minor errors/ omissions can lead to total failure

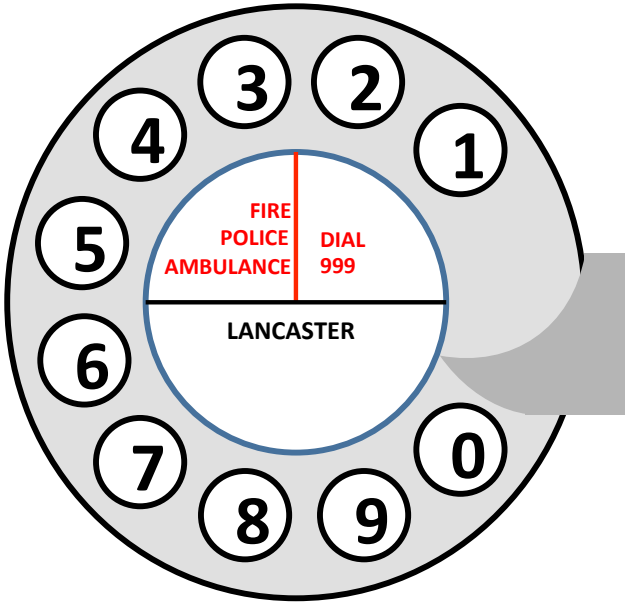
Lancaster University

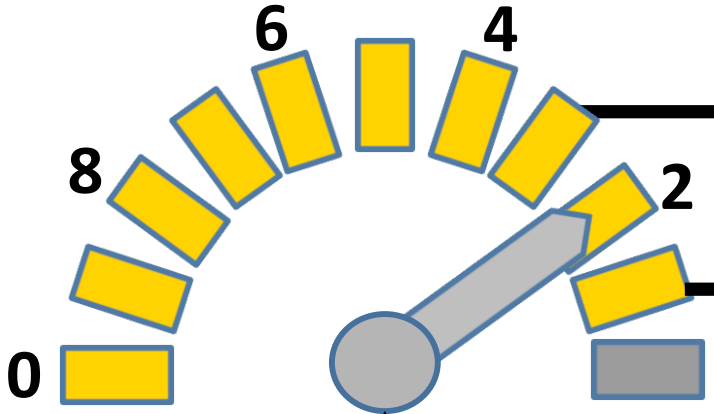
Addressing

Or how to make sure the communication ends up at the right place

Early Telephone System

- System automated using pulses from dial
  - Pulses step armature round using a relay





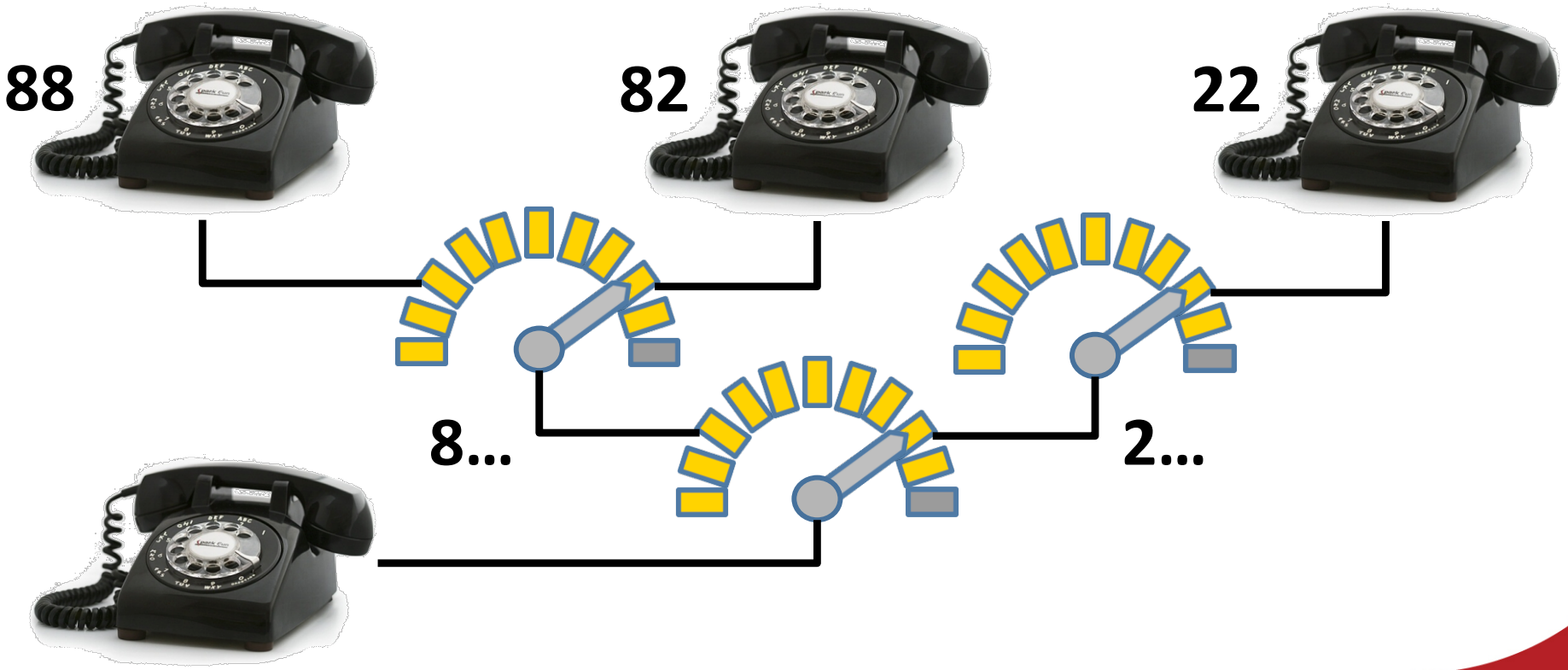
3

1

Adding Digits

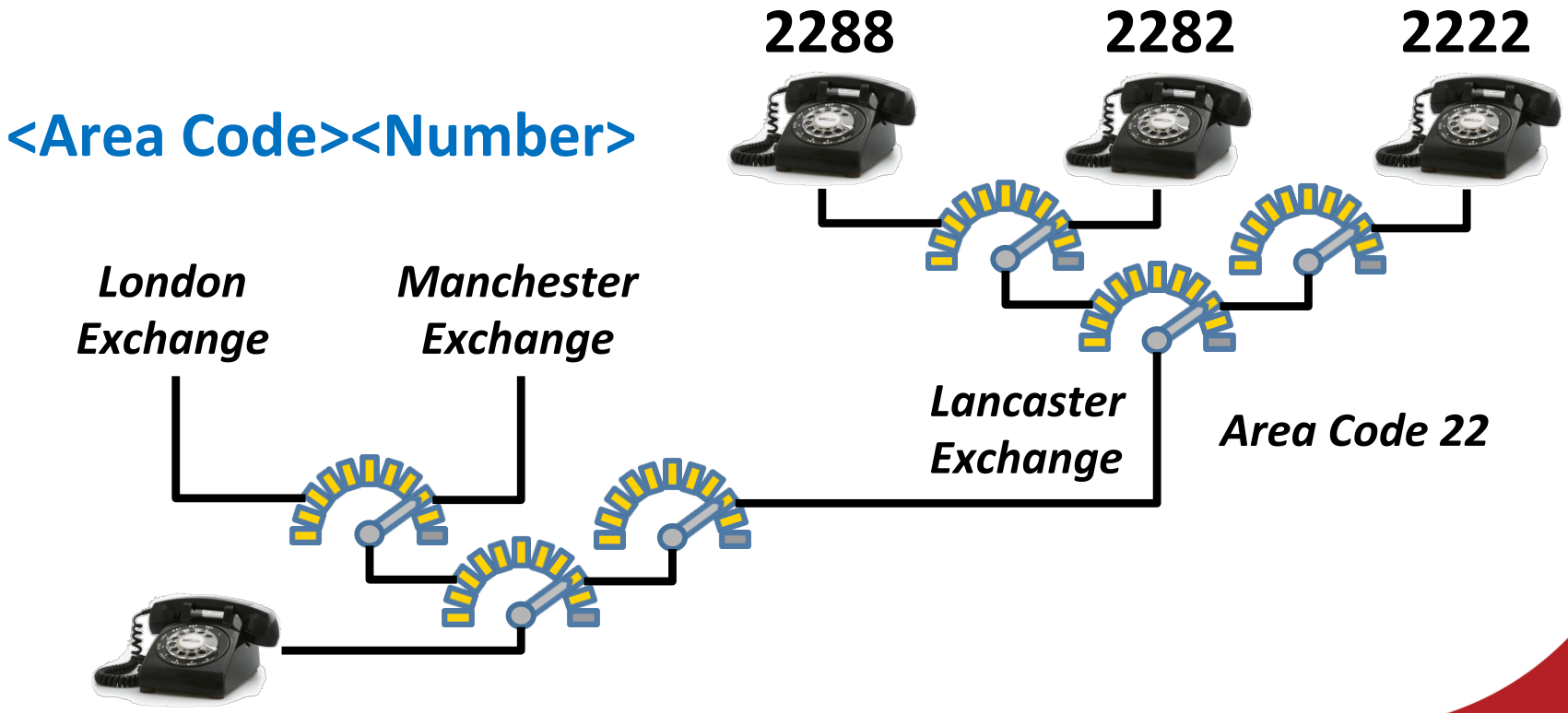
Lancaster University

- Rather than connecting a phone, we can connect other selectors
  - Successive digits step through number hierarchy



Hierarchical Numbers/ Addresses

- This can be further extended
  - Notice we have a hierarchical number
  - Number of distinct groups of digits



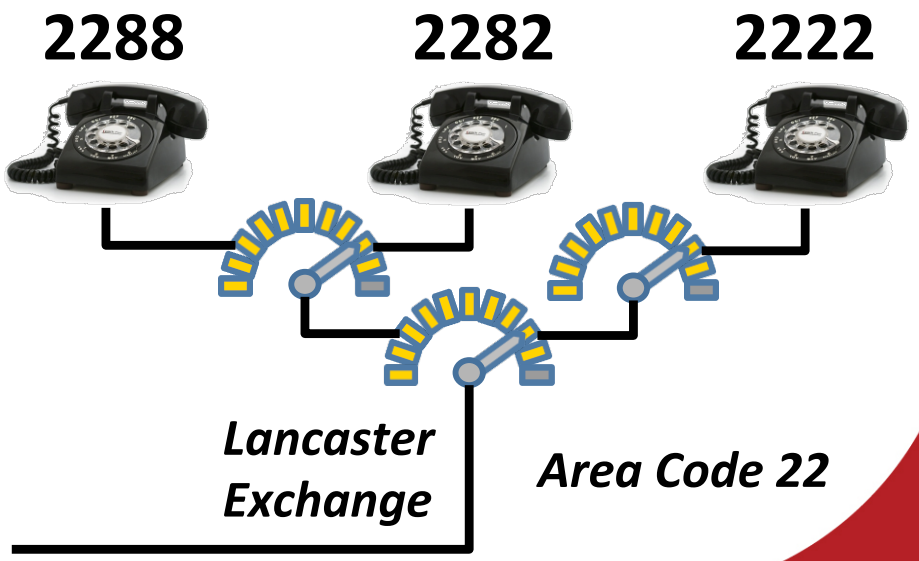
Hierarchical Addresses

Lancaster University

- ***Prefixes*** allow us to infer location from an address
  - Given a number we 'immediately' know where phone is
  - Any number with a 22 prefix must be in Lancaster
    - Well, 0154, actually

- Note it is a topological location
  - Determined by internal network structure
  - Not necessarily geographic

- Scheme used in Internet



Flat Addresses

Lancaster University

- ***More efficient address allocation possible***
  - E.g. easier to balance needs of a city and a village
  - 43% countries have fewer Internet addresses than Lanc. Uni. *
- ***Ethernet uses flat addressing scheme***
  - Blocks of addresses (OUIs) allocated to vendors by IEEE
  - Vendors program addresses into Network Interface Cards (NICs)
  - NICs then 'randomly' distributed around world
- ***Given an address we have no idea where device is***
  - Unless we've seen it recently

Organisation Unique Identifiers (OUIs) :

http://standards.ieee.org/develop/regauth/oui/oui.txt

* based on UN recognised countries/ territories (2012); Internet hierarchical

Example: Lenovo X230 Laptop

Lancaster University

Windows: *ipconfig /all* Linux: *ifconfig*

```
Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel 82579LM
   Physical Address. . . . . . . . . : 3C-97-0E-37-10-69
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
```

3C-97-0E (hex) Wistron InfoComm(Kunshan) Co. Ltd.
168 First Avence
Kunshan Export Processing Zone
Kunshan JiangSu 215300
CHINA

Example: Lenovo X230 Laptop

Lancaster University

Works with other types of physical network

```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel Ultimate-N 6300 AGN
   Physical Address. . . . . . . . . : 24-77-03-71-39-80
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
```

24-77-03 (hex)

Intel Corporate
Lot 8, Jalan Hi-Tech 2/3
Kulim Hi-Tech Park
Kulim Kedah 09000
MALAYSIA

Note: this is not a prefix
just one of blocks allocated to Intel
…can't use to deduce anything of
protocol/ network structure

Lancaster University

Connecting Communication Instances

Connection Modes

Connection Oriented Systems

Lancaster University

- Overhead of connection setup/ teardown
- But clear relationship between transmissions
  - Identifiable connection linking traffic
    - Physical connection: wire, fibre, …
    - Virtual connection (see later): connection identifier
  - Resource use more easily tracked
- May not be possible to continue on failure
  - Need mechanism to re-establish call/ connection

Connection Oriented Systems

Lancaster University

- Resources held for connection end-to-end
  - For full period connection established
- Resources still tied up even if connection idle
- Bandwidth (link capacity) partitioned
  - Connection has exclusive use of negotiated capacity

    …even if it's not actively using link

Routing Packets to their Destination: The Connection-less Alternative

Lancaster University

- Packet Switching
  - Developed independently by three groups c1961-64
    - National Physical Laboratory (NPL), UK
      - Donald Davies, Roger Scantlebury
    - MIT, USA
      - Leonard Kleinrock
      - Later: JCR Licklider, Lawrence Roberts at MIT and ARPA
    - Rand Institute, USA
      - Paul Baran
  - First 'Internet' packet switch installed 1969

ARPA: US Advanced Research Projects Agency

Packet Switching

Lancaster University

- Each transmission
  - Independent packet of data
  - A set of bits/ bytes in defined format

- Each packet like an addressed envelope
  - Header holds address or ID for path selection*
  - Payload holds content/ application data

** Used to determine how we get from sender to receiver*

Lancaster University

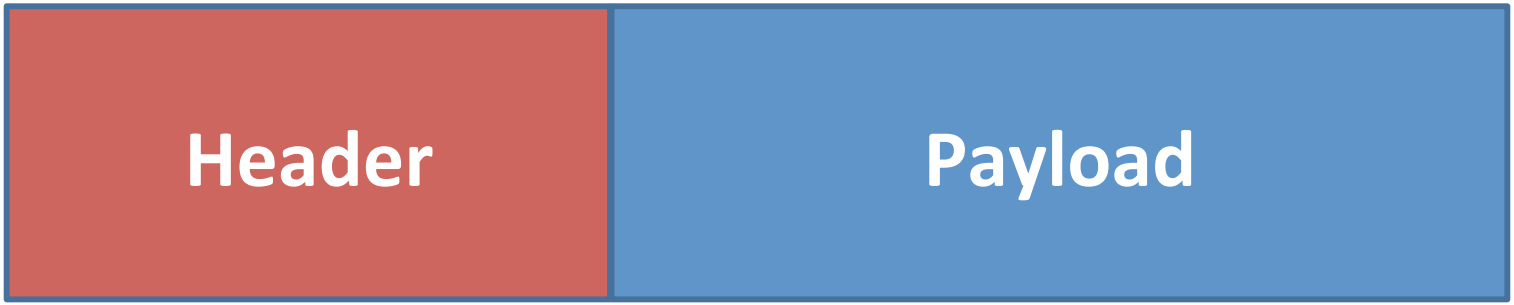
Packet Formats & Communication Modes

What do electronic packets look like and how do they get to their destinations?

A Simple Packet

Lancaster University

Transmitted byte-by-byte across network



Packet Switching

Lancaster University

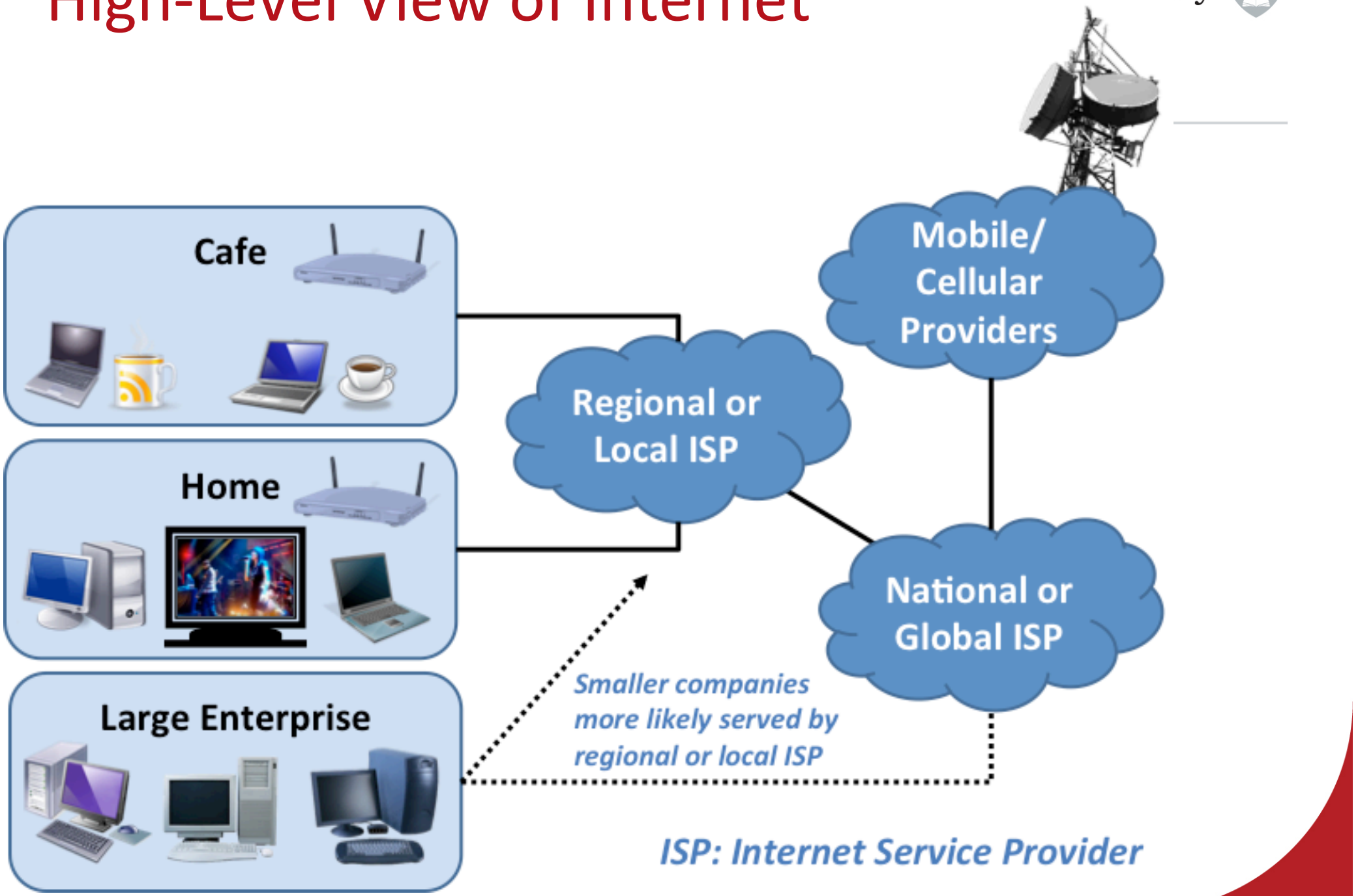
- Users share capacity
  - Packets use full link bandwidth for short period
    - Was a novel idea; traditionally bandwidth partitioned
- Resources only used as and when required
  - Statistical multiplexing
    - Allow resource demand to exceed supply
    - Queue packets until available link capacity
    - Congestion can lead to delay or packet loss

Packet Switching

Lancaster University

- Two modes
  - Datagram (connection-less)
    - Applications **aware** of packet structure
  - Virtual Circuit (non-physical connection)
    - Applications **unaware** of packet structure
- Modes determine
  - Style of communication
  - How addressing information used for Path Selection (how we get from A to B)

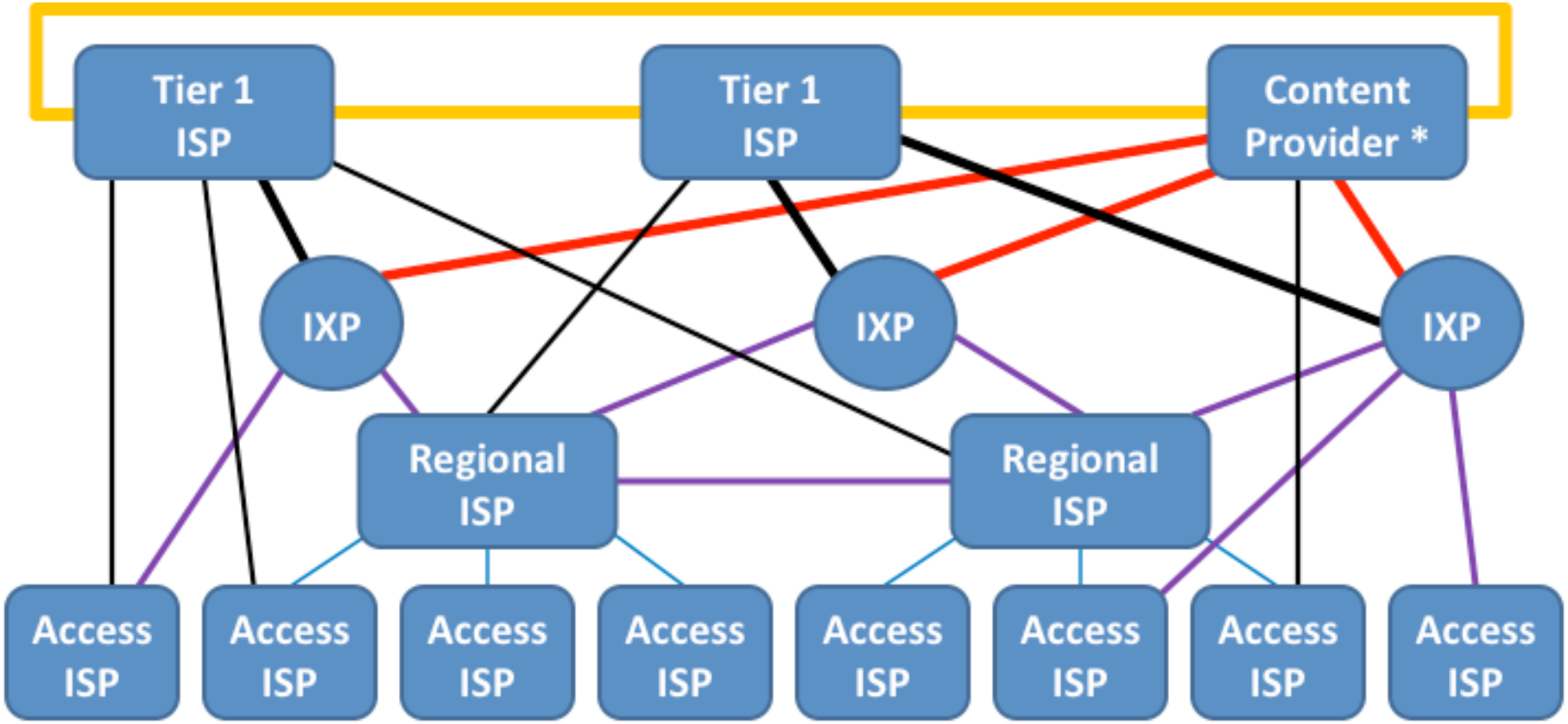
High-Level View of Internet



Connections Pragmatic

Lancaster University

- Volume of traffic, link speed, cost of provision, …



IXP: Internet eXchange Point

** With global reach: Google, …*

Getting from A to B: *Path Selection*

Lancaster University

- Datagrams
  - Each packet considered as separate message
  - Datagrams like postcards
    - Every short message fully addressed
  - Header holds address
  - Payload holds application data

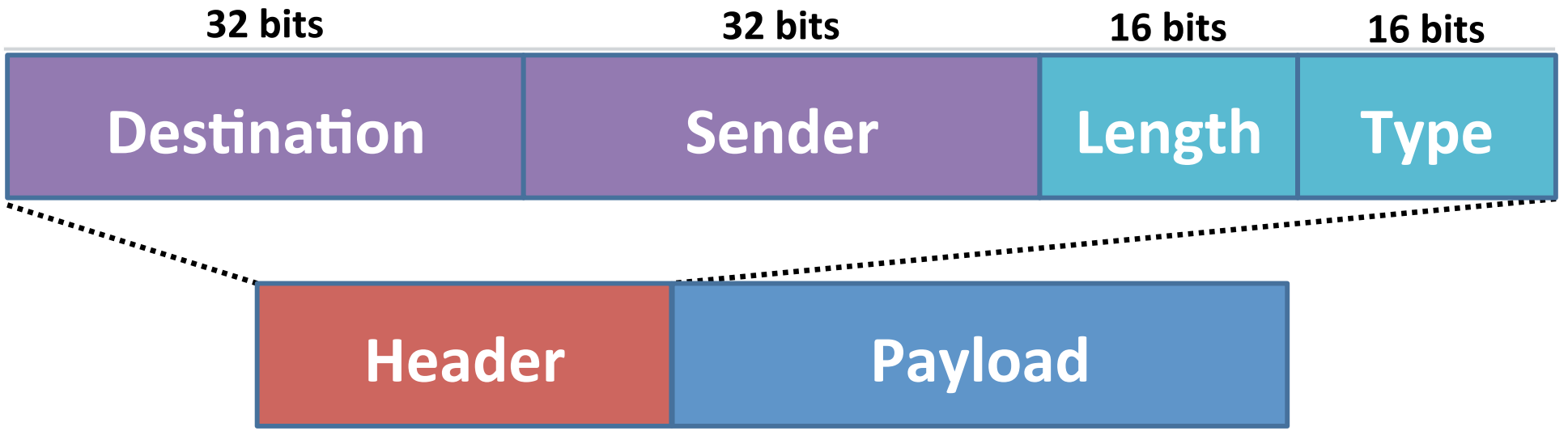
A Simple Datagram

Lancaster University

- *Payload type* tells receiver what content is
- *Sender* needed so we know where to reply to


```
struct datagram_pkt {
    struct {
        struct address  destination;
        struct address  sender;
        unsigned int16  dgram_length;
        unsigned int16  payload_type;
    } header;
    char payload [ ];
};
```

A Simple Datagram




```
struct datagram_pkt {
    struct {
        struct address  destination;
        struct address  sender;
        unsigned int16  dgram_length;
        unsigned int16  payload_type;
    } header;
    char payload [ ];
};
```

Path Selection

Lancaster University

- Virtual Circuits
  - Emulate connection oriented system …over connectionless (packet) network
  - Phone-like connection set-up phase (message exchange)
    - Connection state must be retained for duration
    - Fixed path maintained for duration of connection
  - Each packet contains connection tag
    - Tag is Virtual Circuit identifier used for path selection
  - Payloads holds application data

A Simple Virtual Circuit Packet

Lancaster University

- Tag identifies connection and thus sender and receiver


```
struct vc_packet {
   struct {
      unsigned int16  tag;
      unsigned int16  packet_length;
      unsigned int16  payload_type;
   } header;
   char payload [ ];
};
```

Notice we have (at least) two levels

Lancaster University

- Application view (service provided to applications)
  - Datagrams Connection Less
  - Virtual Circuits Connection Oriented
- Physical view (how underlying network operates)
  - Packet switching Connection Less
  - Circuit switching Connection Oriented
- So we can provide either type of application service over either type of physical network

Store and Forward Networks

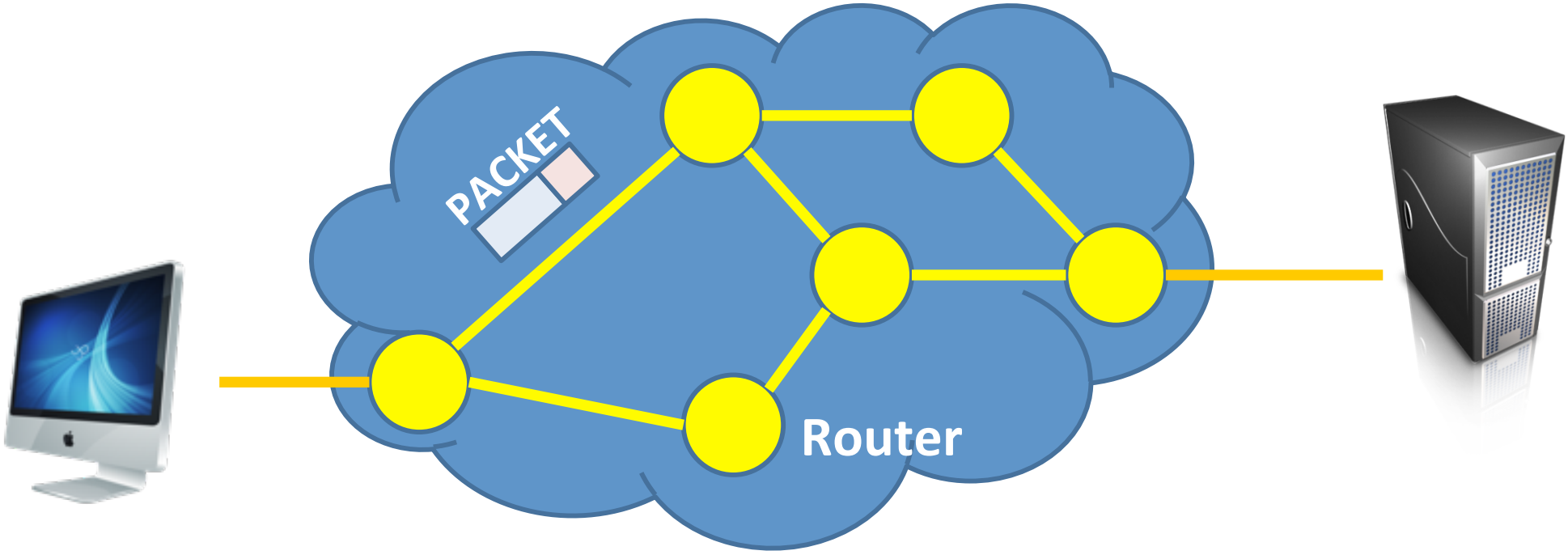
Lancaster University

- Packet Switches
  - Aka Routers
  - Receive packet
  - Read destination or ID
  - Find next step in path
  - Perform housekeeping
  - Send packet toward destination

Store and Forward Networks

Lancaster University

- Named as such because whole packet…
  - Received into local storage, then forwarded

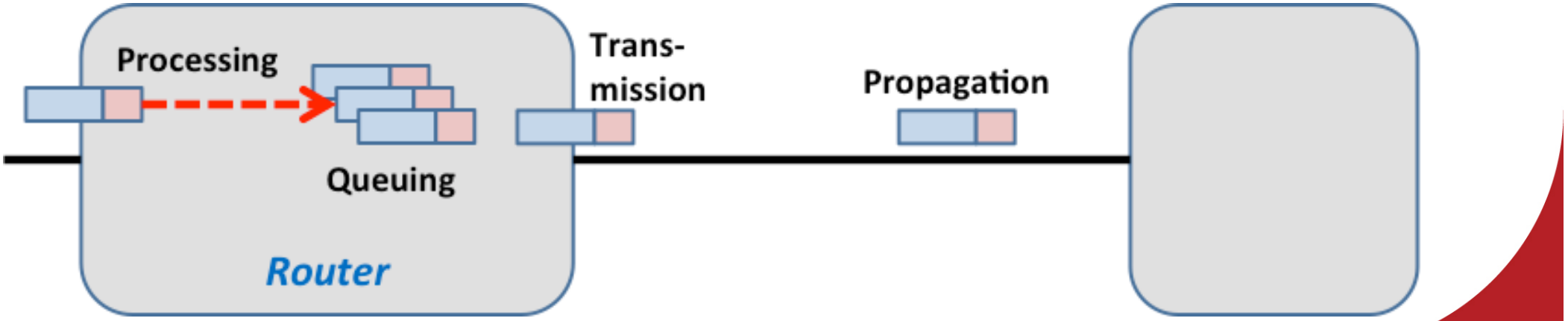


Packet Switching Delays

Lancaster University

- Transmission delay
  - Length of packet / bits per second
- Propagation delay
  - Length of link / propagation speed in material*

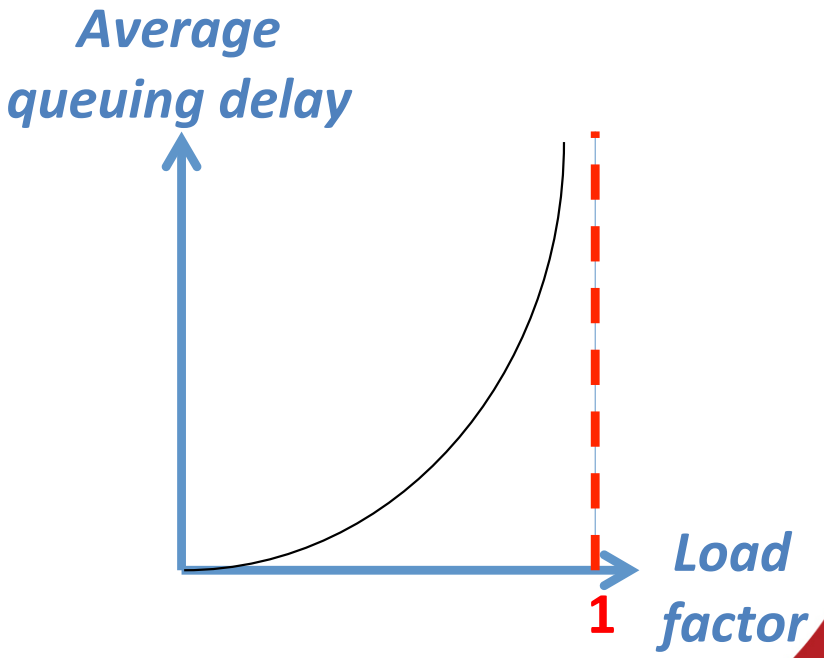
** Some % of speed of light*



Congestion

Lancaster University

- Load factor
  - Attempted transmission rate / outgoing link rate
- Delay rapidly increases as load factor tends to 1
  - Above 1 delay is unbounded
    …exceeding link capacity
- Use queuing theory to model this
    See later modules…



Complexity

Lancaster University

- We have a **distributed system**
- Dealing with many communicating systems
  - Each independent
  - Each doing what it wants, when it wants
- Communication protocols must
  - Sequence actions
  - Ensure every case covered unambiguously

Summary

Lancaster University

- Relevant networking principle and concepts
  - Network structures, addressing, packet formats, communication modes

  ➔ These are abstractions and concepts on which computer communication is built!!
- Communication controlled by protocols
  - Tightly and completely defined
  - Control exactly what's transmitted, how, and when
    - Detail all states sender and receiver can be in
    - When and how to transition between states

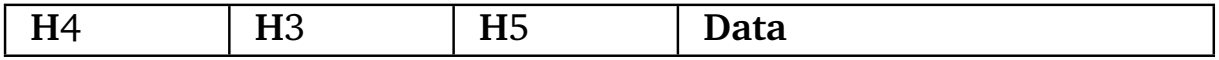
Questions?

Lancaster University

Questions!!

Lancaster University

- Explain the terms connection-oriented service and connectionless service with examples
- Consider the following illustration of a packet. What do you notice?



H: Header-Data

- A system has a n-layer protocol hierarchy. Applications generate messages of length M bytes. What fraction of the network bandwidth is filled with headers, if at each of the layers a h-byte header is added?

Lancaster University

SCC203 Computer Networks: Layering & Reference Models

Dr Andreas Mauthe (a.mauthe@lancaster.ac.uk)
Dr Andrew Scott (a.scott@lancaster.ac.uk)

Today's Lecture

Lancaster University

- Outline
  - Basic Concepts
    - Communication modes
    - Packets
  - Layered Communication
    - What is layering
    - Reference models
  - Communication Stacks
    - Hour Glass model
  - Communication Layers
    - Physical, Data Link, Network, Transport,
    - Upper Layer Architecture
      - Session, Presentation, Application Layer
  - Internet Basics
    - Internet Standards & core elements
    - End-to-end argument

- Objectives
  - To give an overview of the communication different layers and their purpose in the communication stack
    - You should know the basic functionality of the different layers
  - To get familiar with central concept of the Internet
    - You should understand what the different layers of the Internet do and be able to distinguish them from the OSI reference model

Lancaster University

From yesterday's lecture

Communication Modes, Headers, etc.

Packet Switching

Lancaster University

- Two modes
  - Datagram (connection-less)
    - Applications **aware** of packet structure
  - Virtual Circuit (non-physical connection)
    - Applications **unaware** of packet structure
- Modes determine
  - Style of communication
  - How addressing information used for Path Selection (how we get from A to B)

A Simple Datagram

Lancaster University

- *Payload type* tells receiver what content is
- *Sender* needed so we know where to reply to


```
struct datagram_pkt {
    struct {
        struct address  destination;
        struct address  sender;
        unsigned int16  dgram_length;
        unsigned int16  payload_type;
    } header;
    char payload [ ];
};
```

A Simple Virtual Circuit Packet

Lancaster University

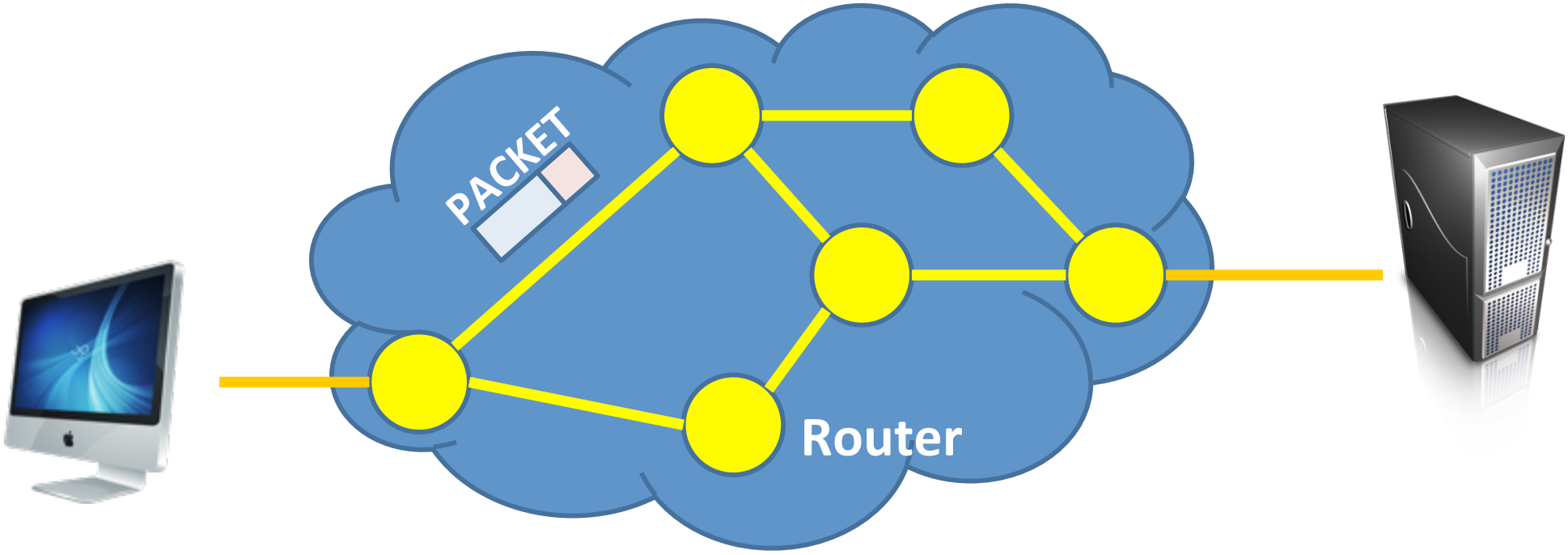
- Tag identifies connection and thus sender and receiver


```
struct vc_packet {
   struct {
      unsigned int16  tag;
      unsigned int16  packet_length;
      unsigned int16  payload_type;
   } header;
   char payload [ ];
};
```

Store and Forward Networks

Lancaster University

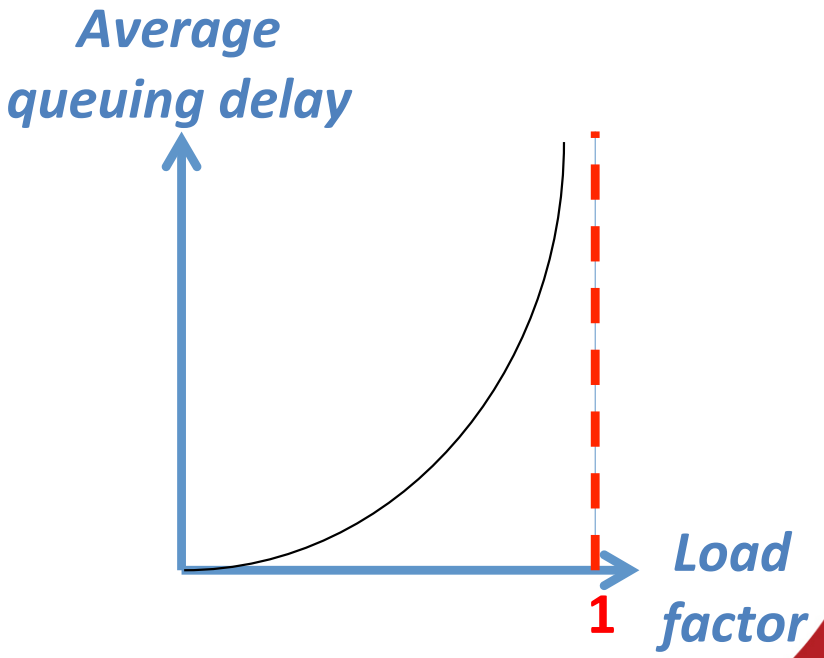
- Named as such because whole packet…
  - Received into local storage, then forwarded



Congestion

Lancaster University

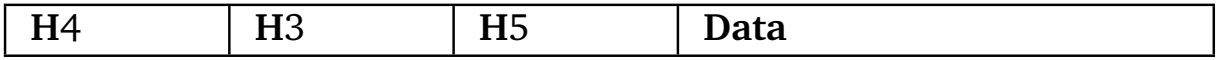
- Load factor
  - Attempted transmission rate / outgoing link rate
- Delay rapidly increases as load factor tends to 1
  - Above 1 delay is unbounded
    …exceeding link capacity
- Use queuing theory to model this
    See later modules…



Questions!!

Lancaster University

- Explain the terms connection-oriented service and connectionless service with examples
- Consider the following illustration of a packet. What do you notice?



H: Header-Data

- A system has a n-layer protocol hierarchy. Applications generate messages of length M bytes. What fraction of the network bandwidth is filled with headers, if at each of the layers a h-byte header is added?

Lancaster University

Communication Protocols & Layered Communication

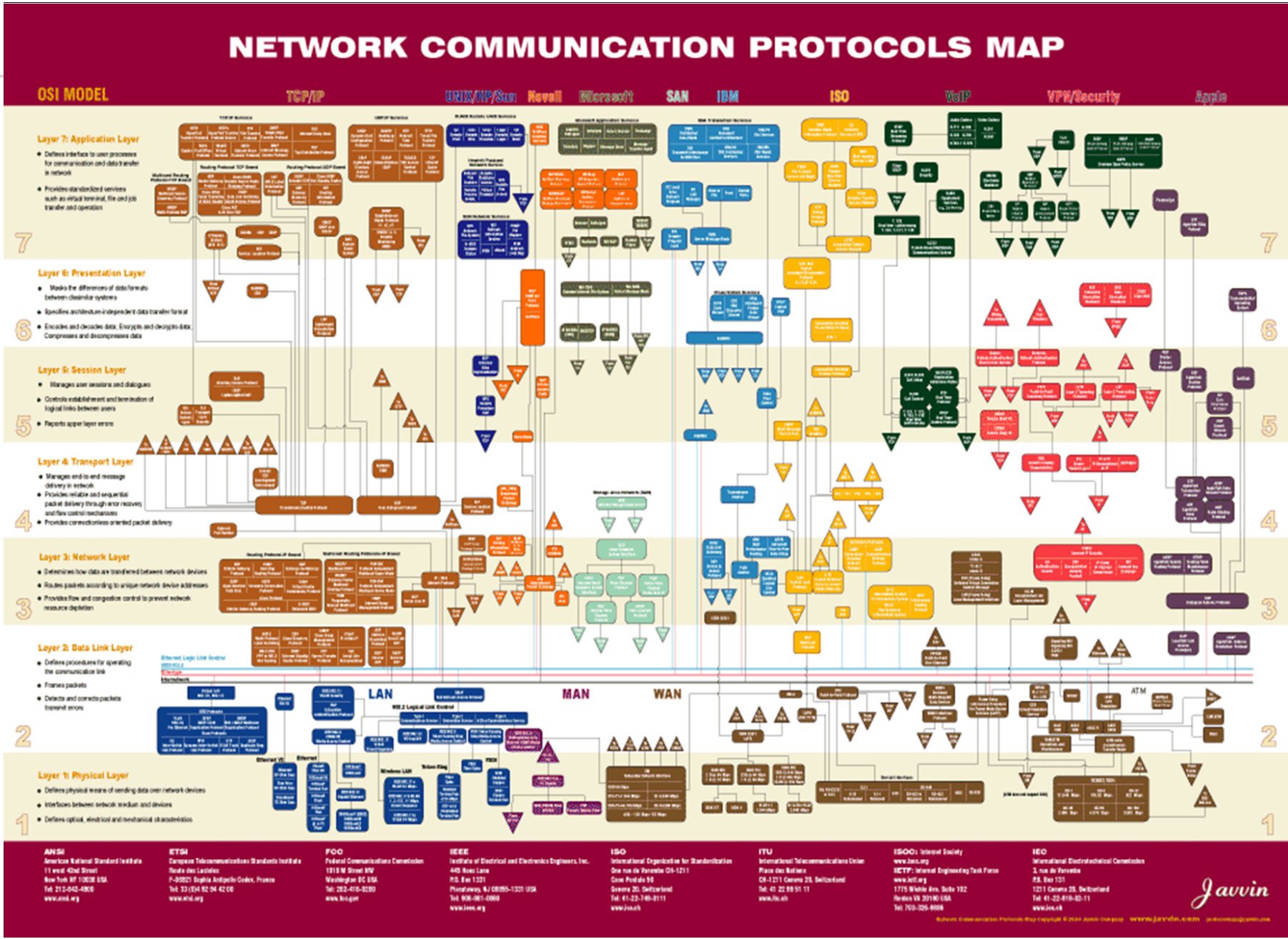
Cornerstones of Communication Infrastructures

Communication Protocols

Lancaster University

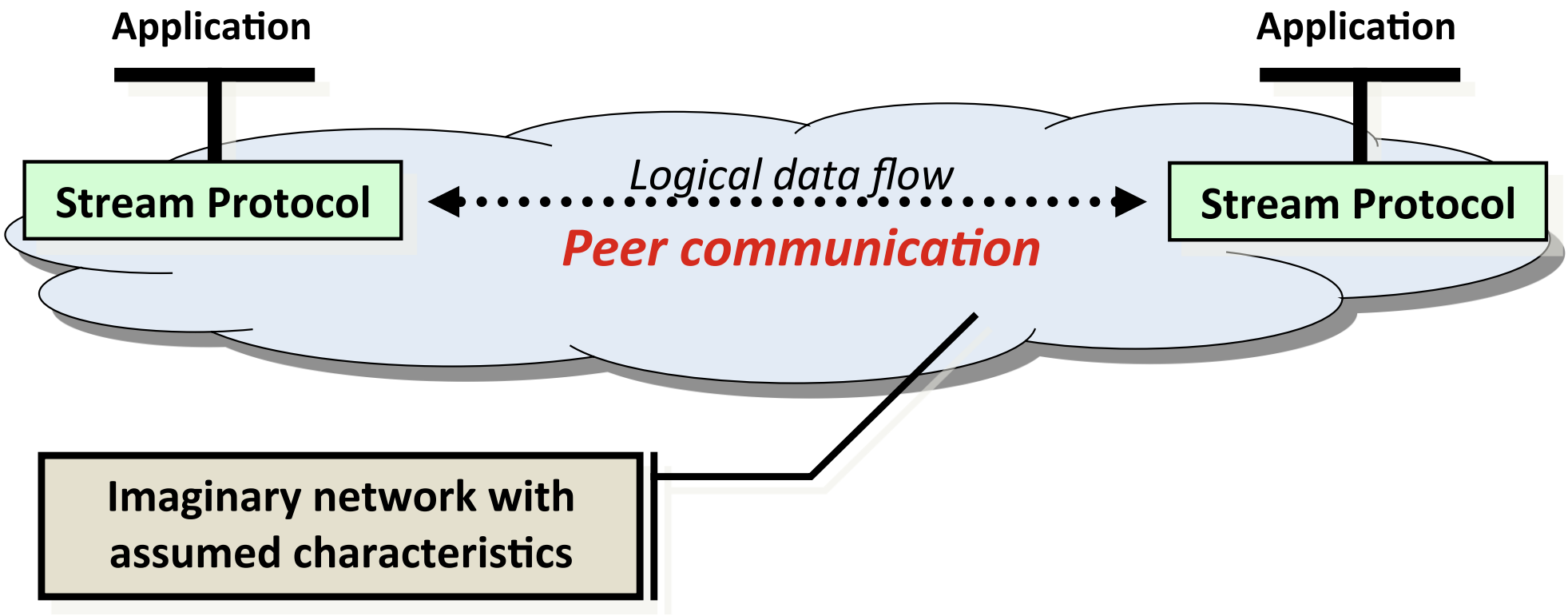
- Specification of
  - How and when data is transmitted and received
  - Allowed messages and expected/ required replies
  - Ordering and timing of transmissions
  - Exact format of transmitted data
- Protocols often make assumptions that demand lower-level protocols
  - Layering hides complexity
    - Easier to understand, implement and debug
  - We talk of communicating peers using the same protocol

Protocol Map

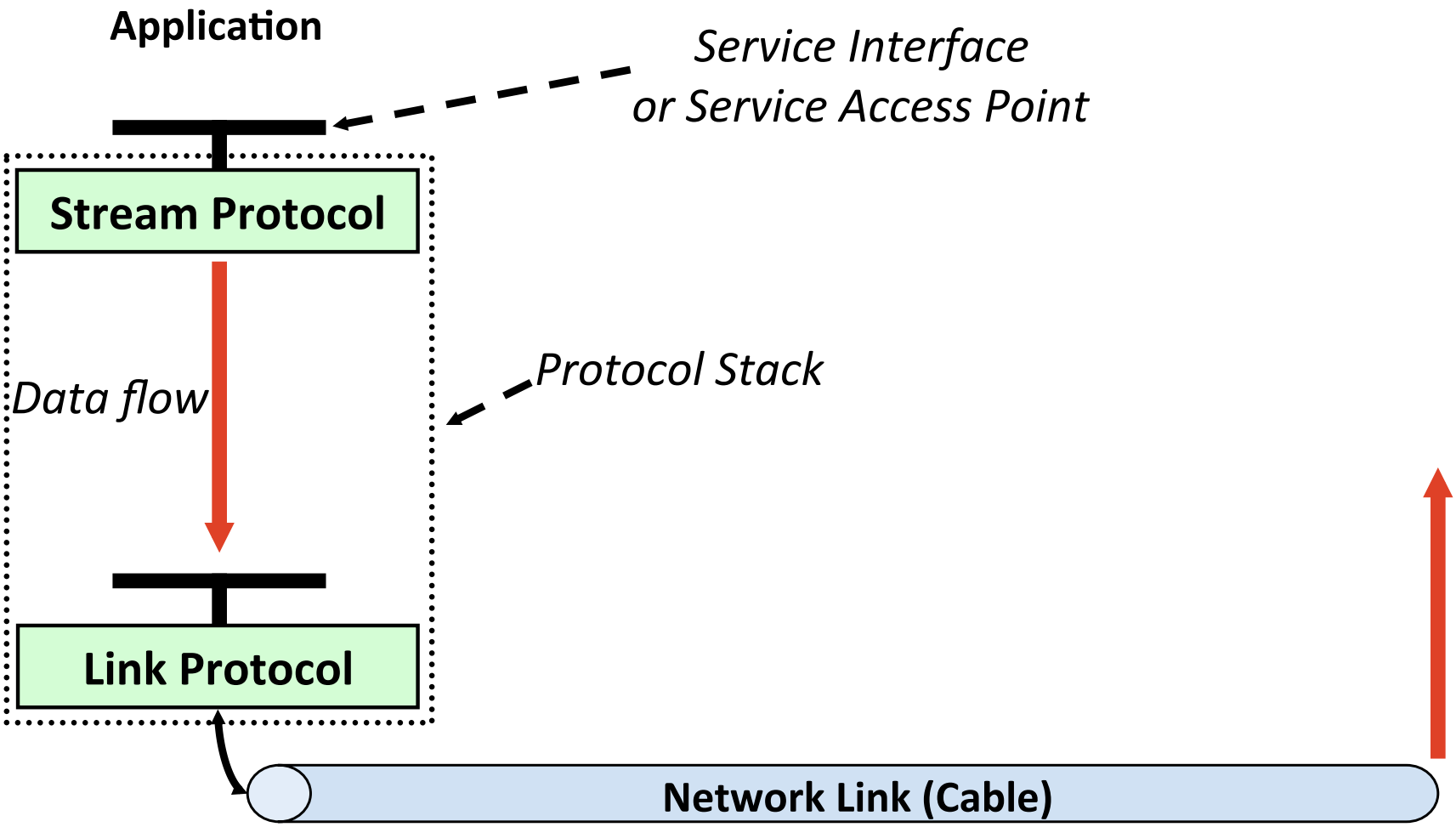


Using Protocols

Lancaster University

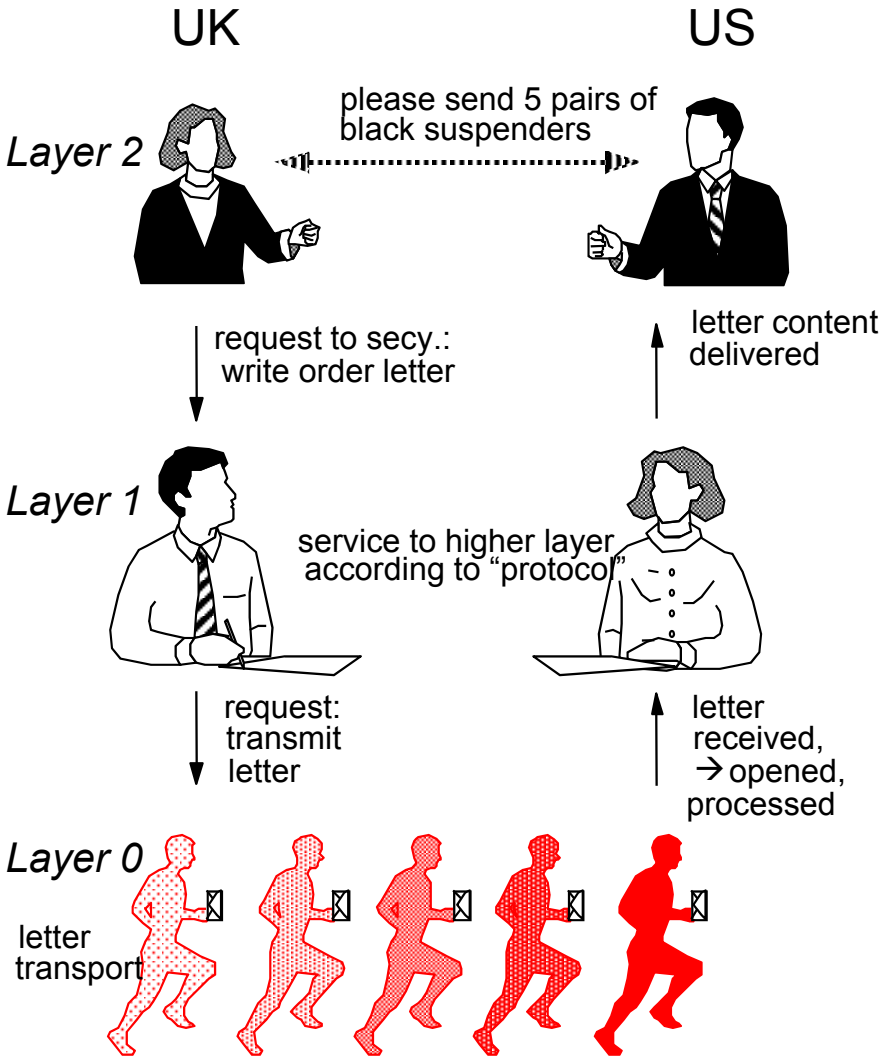


We must consider physical links…

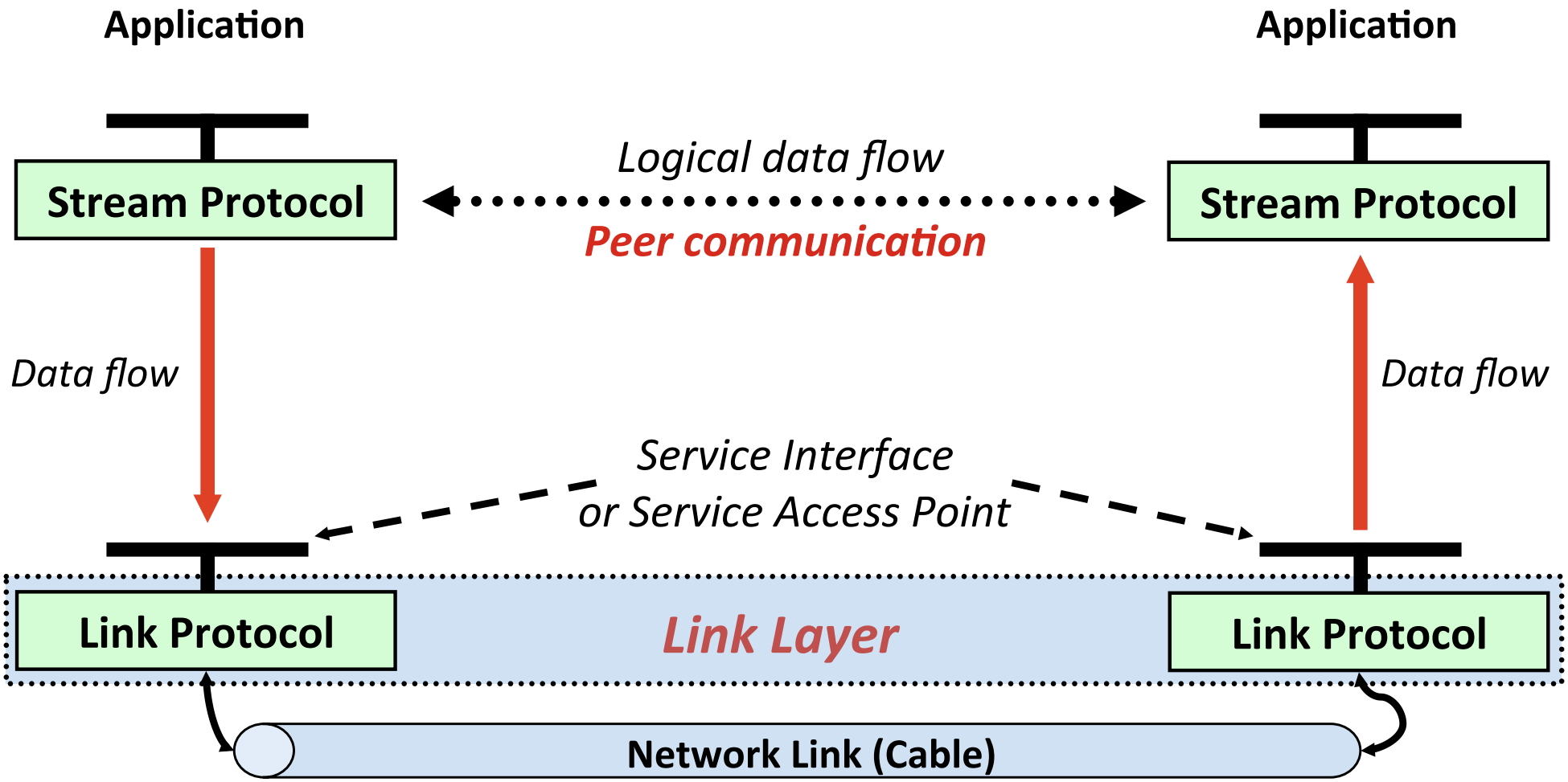


Using Layers

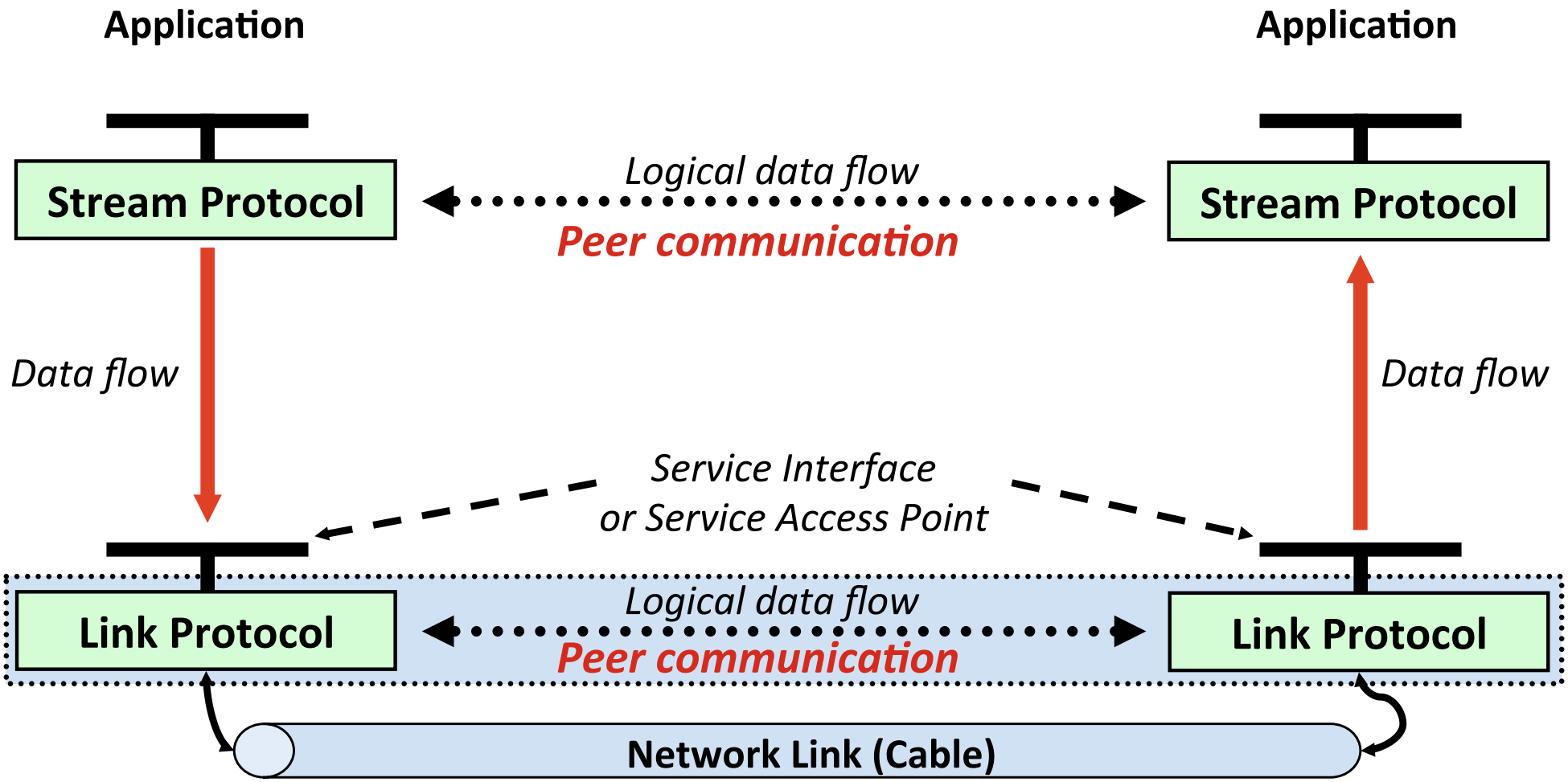
- Layer
  - Provides service to layer above
  - Content is exchanged at layers independently
- Protocol
  - Ensure peer entities interoperate correctly



Think of layers talking to their peers



Think of layers talking to their peers

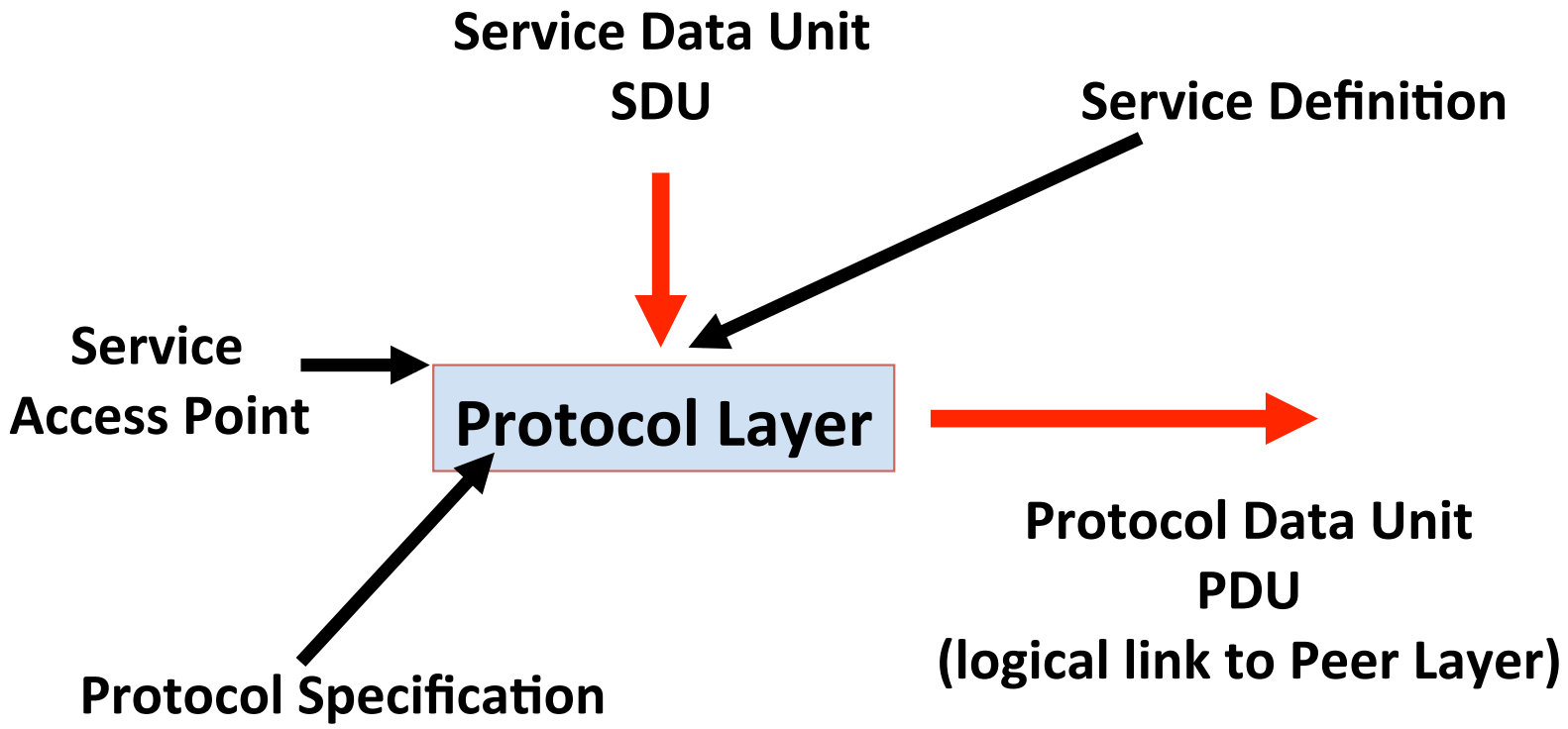


Layered Communication Models

Lancaster University

- We use layers to abstract out complexity
- So, applications communicate at Application Layer
  - Focus on end-points: sender and receiver
    - Can ignore problems of getting data between end-points
- Lower layers can deal with this complexity
  - But they, themselves, ignore detail
    - Which is, in turn, provided by lower layers
      etc.

Layered Communications Model



SDU = Higher layer PDU + Parameters and control information

Lancaster University

Reference Models

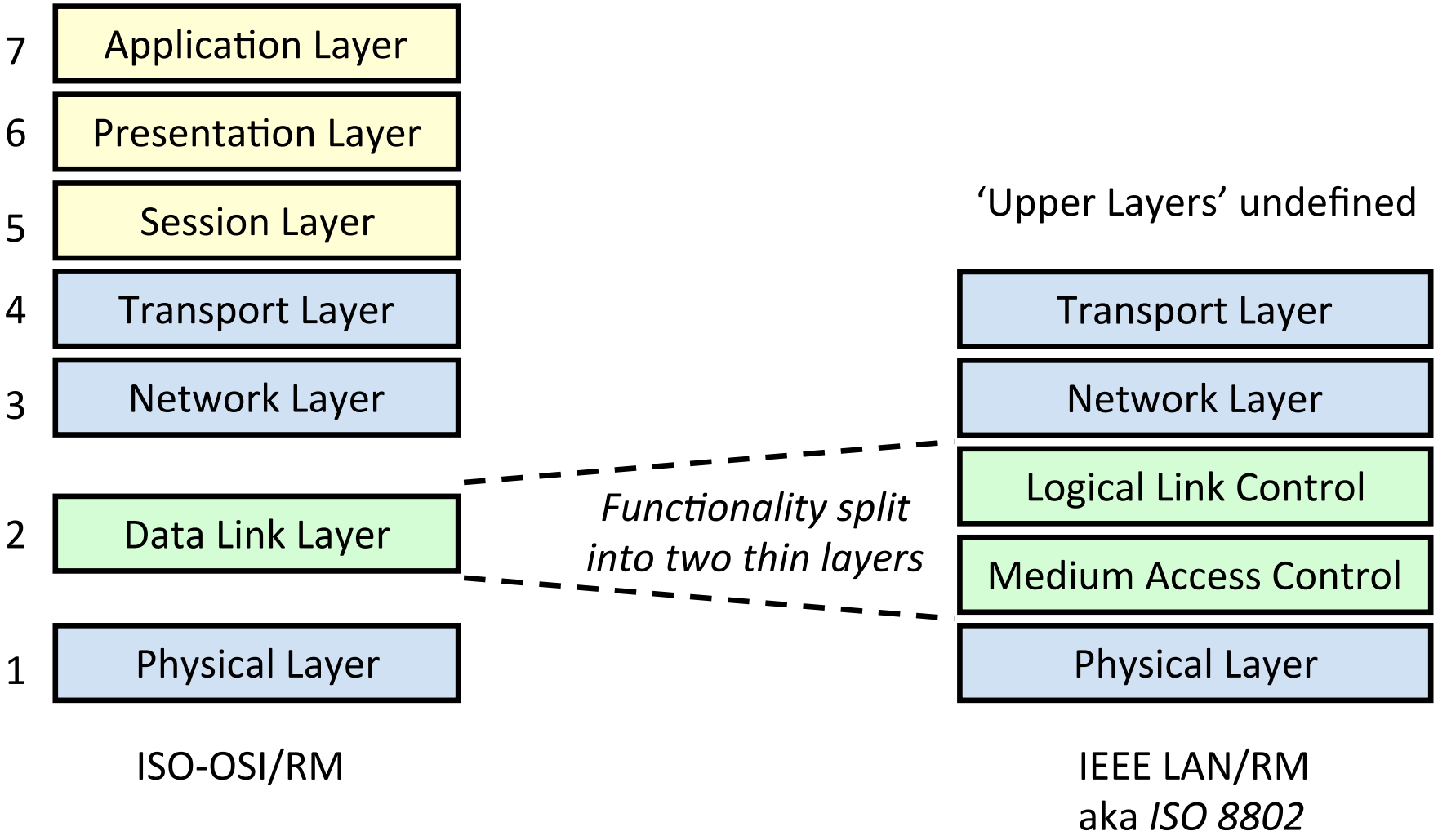
Ordering and Specifying Communication Functions across the Layers

Relevant Reference Models

Lancaster University

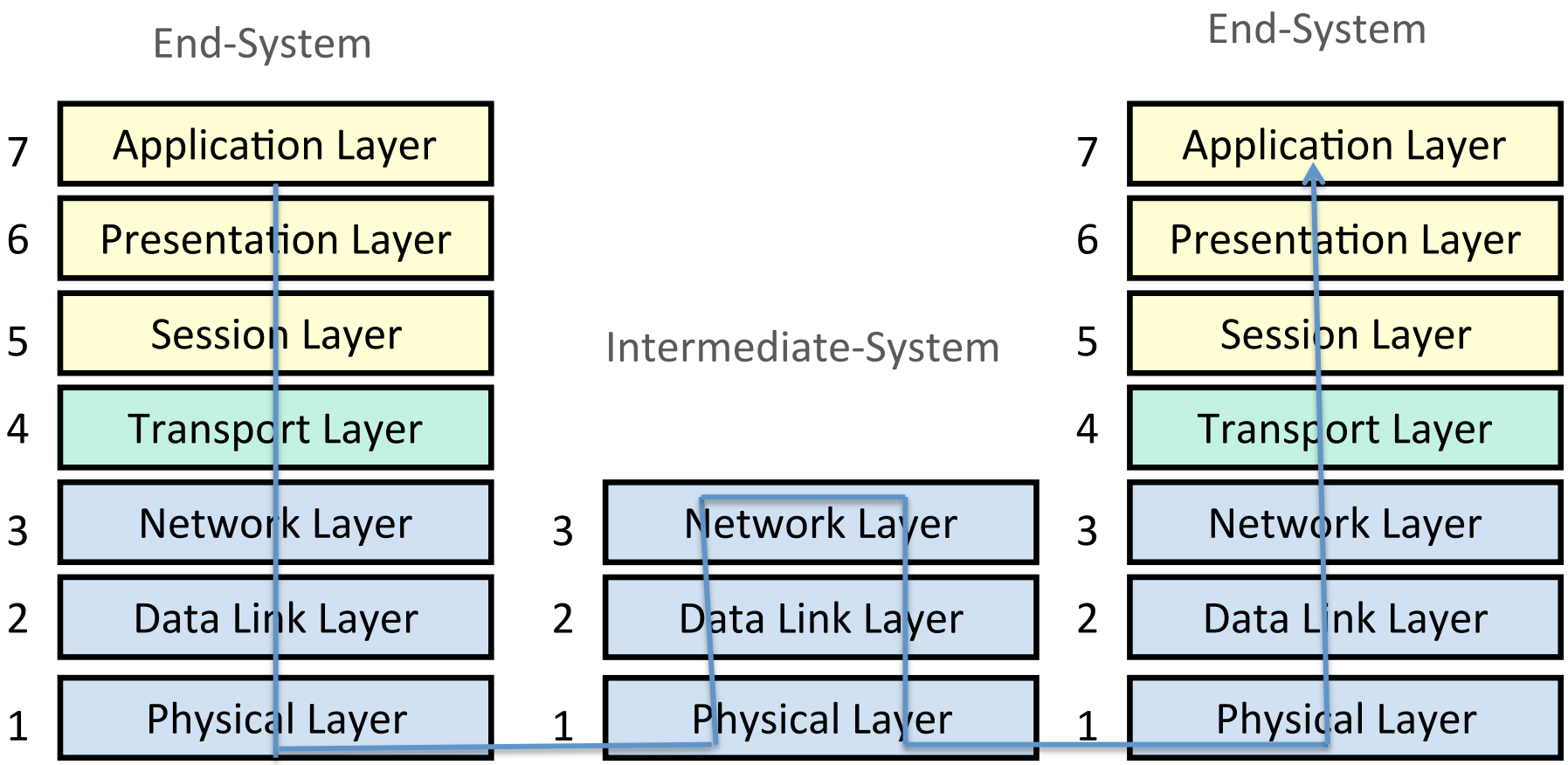
- International Standards Organisation's Open Systems Interconnection Reference Model (ISO-OSI/RM)
  - Widely known as the 7-layer Reference Model
- IEEE Local Area Network Reference Model (IEEE LAN/RM)
  - Widely known from the IEEE 802 standards
- Purpose
  - Define benchmarks (a reference model) by which different protocols and systems can be compared
    - Functionality split into, and defined by (protocol) layers, each with well defined interfaces protecting it from changes in other layers
    - Layers make it easy to add new protocols using same interface

Protocol Stacks



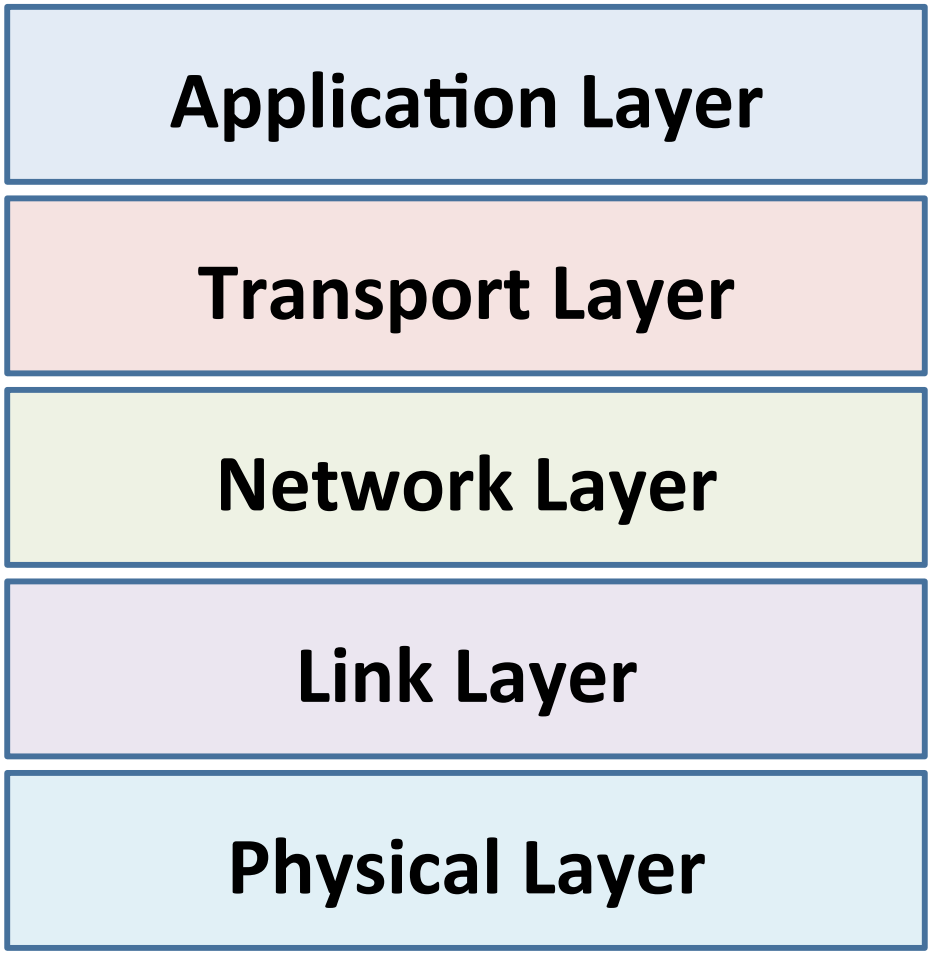
Data Flow in the OSI Model

Lancaster University

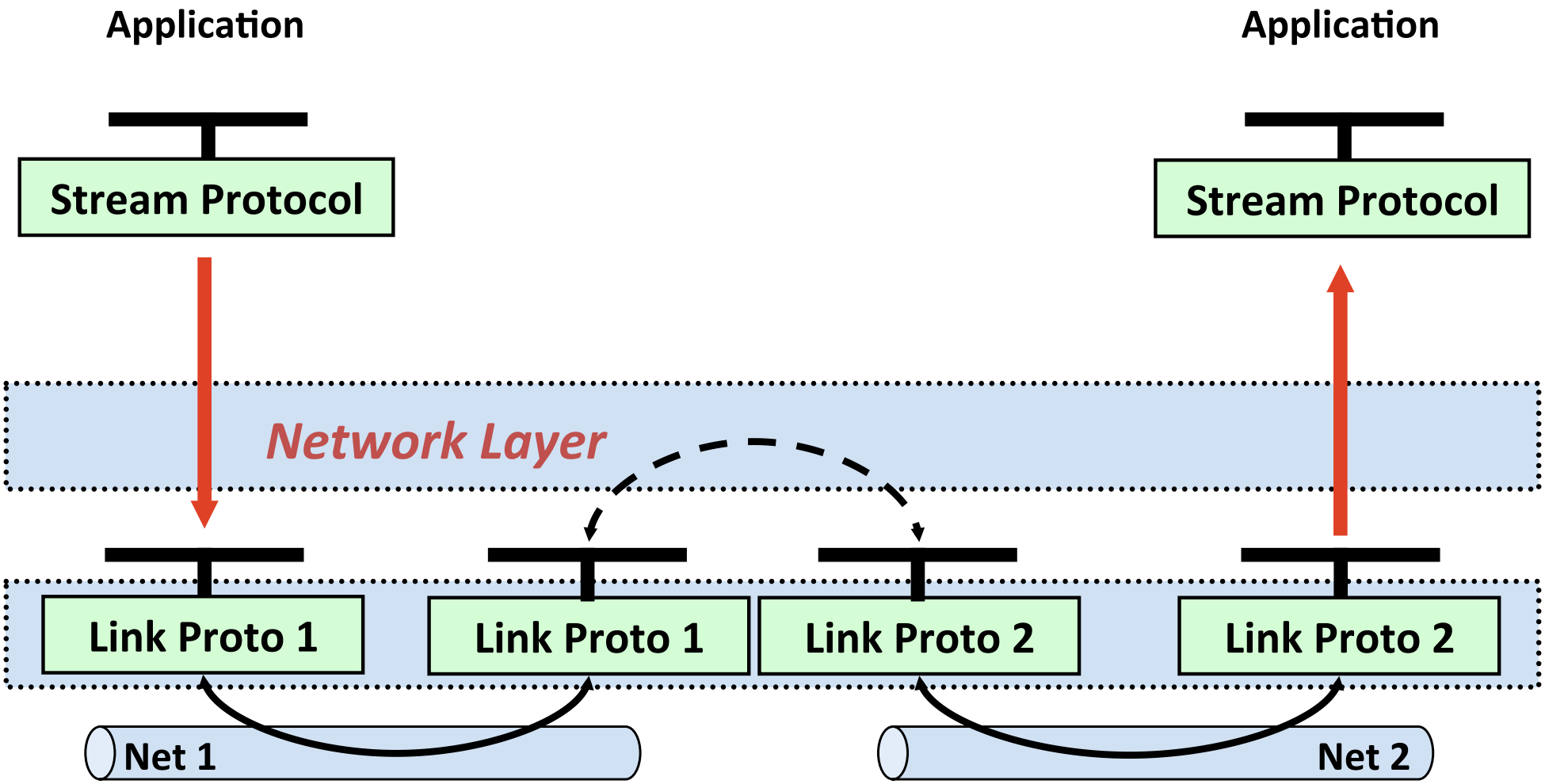


5-layer Internet Model

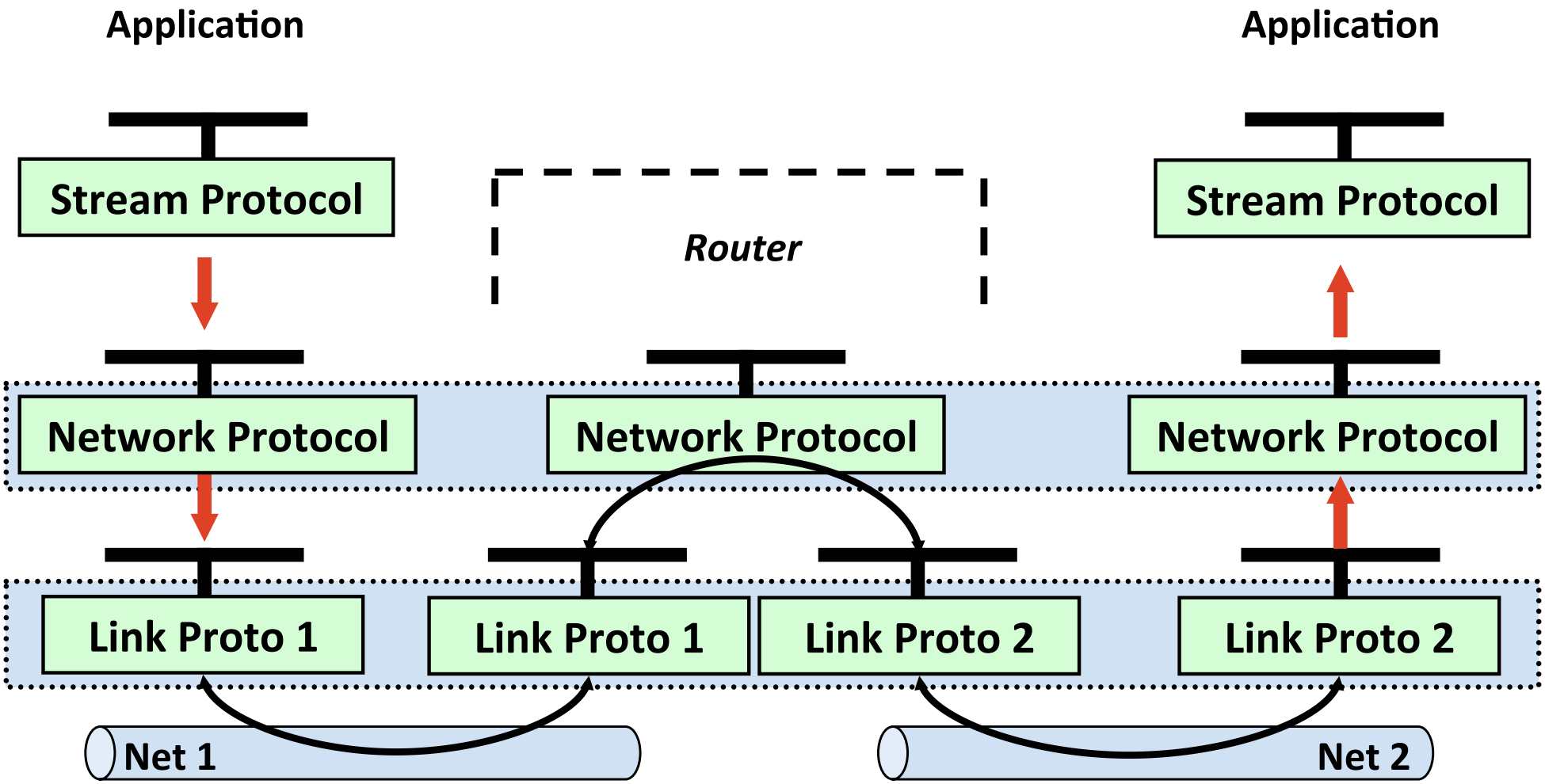
Lancaster University



Multiple Networks

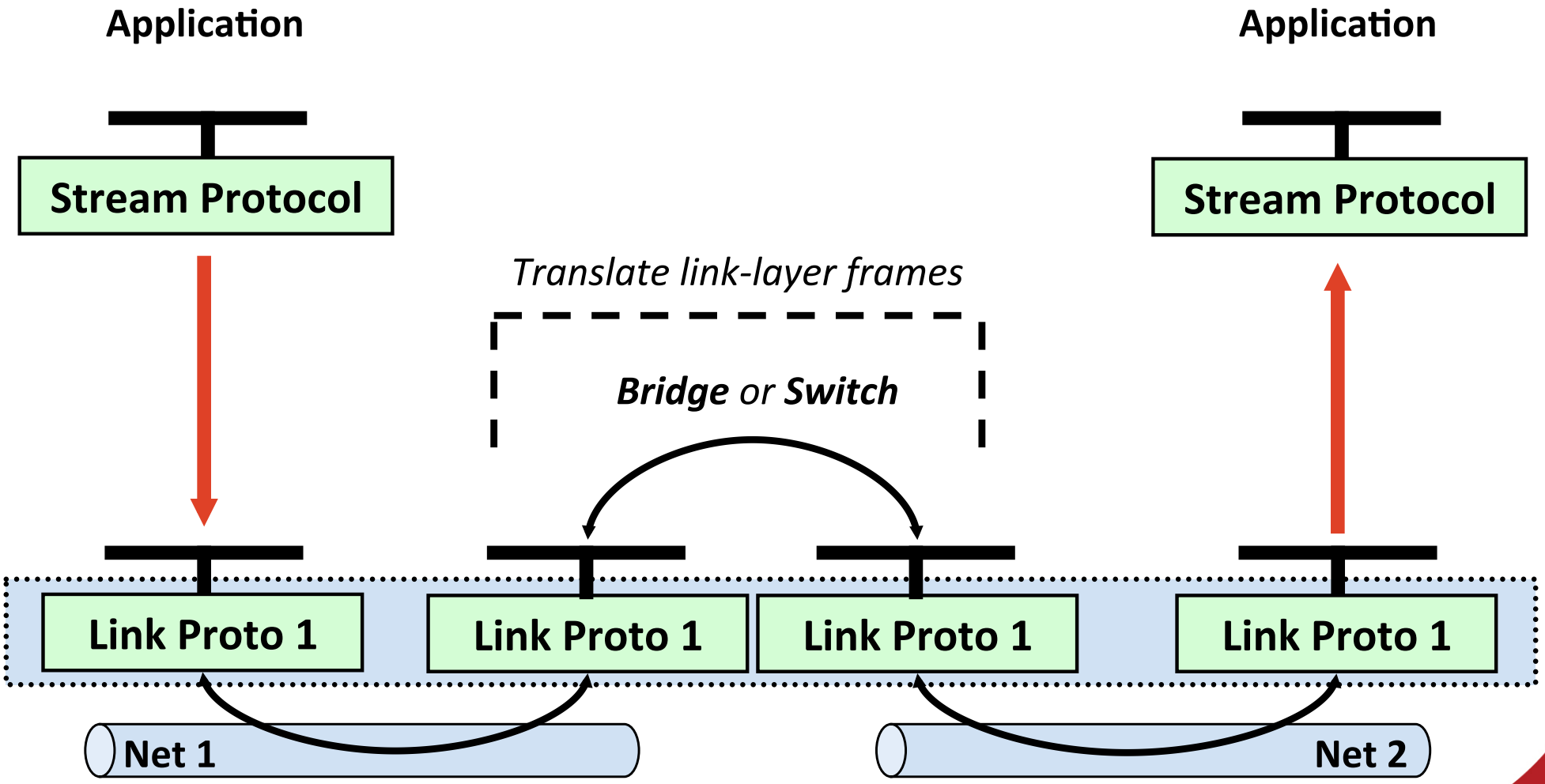


Multiple Networks



If link layers are same or very similar

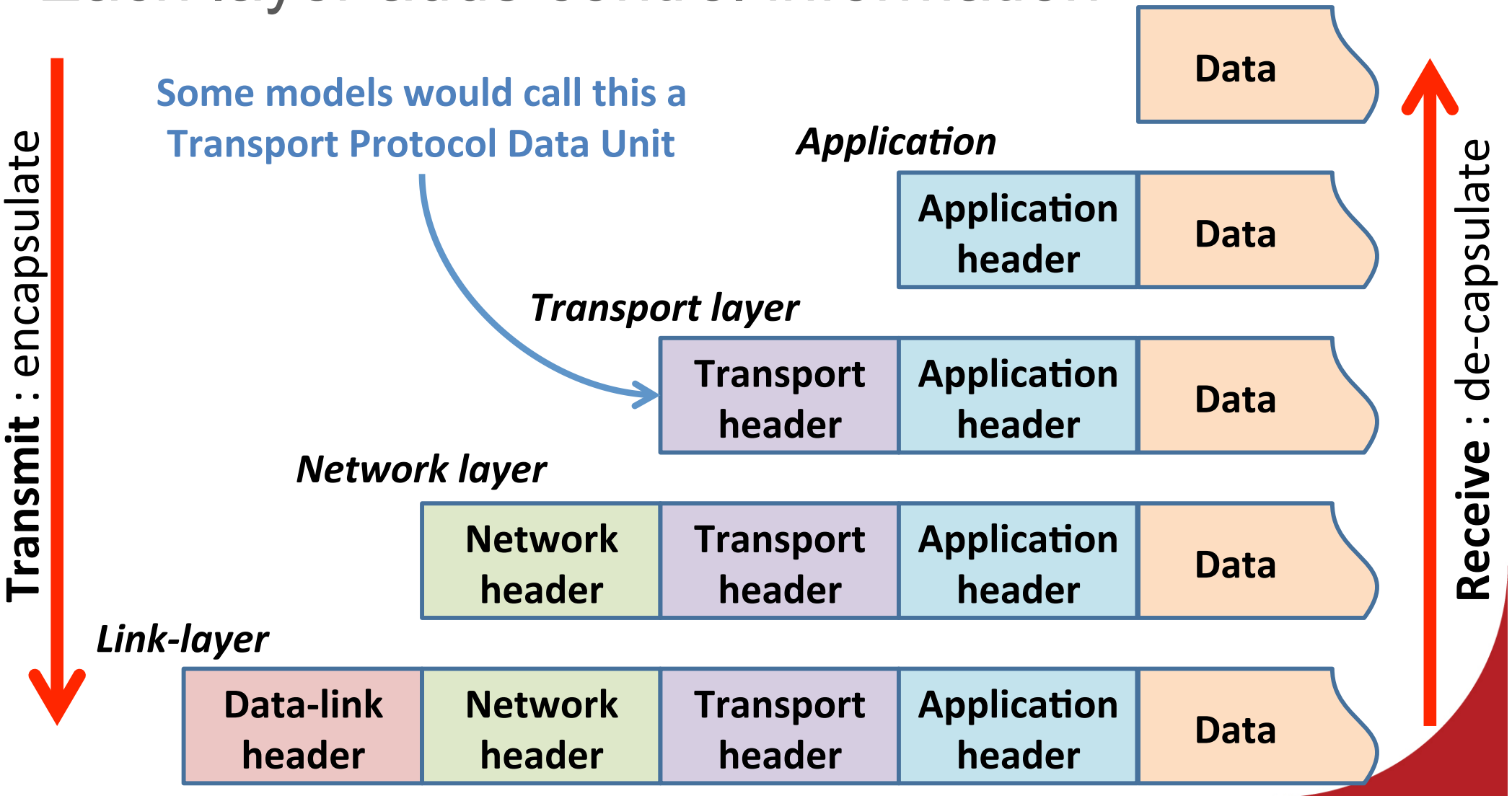
…and need no intelligence/ special functions



Encapsulation

Lancaster University

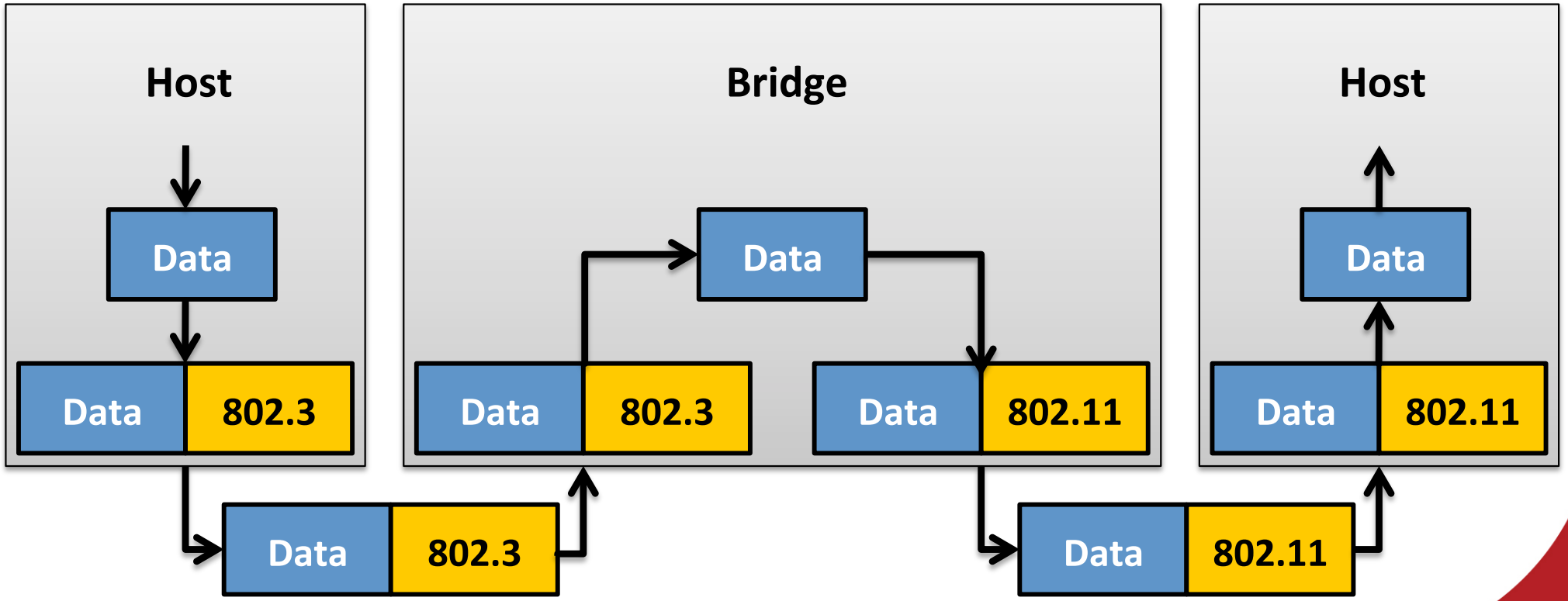
Each layer adds control information



Changing Networks

Lancaster University

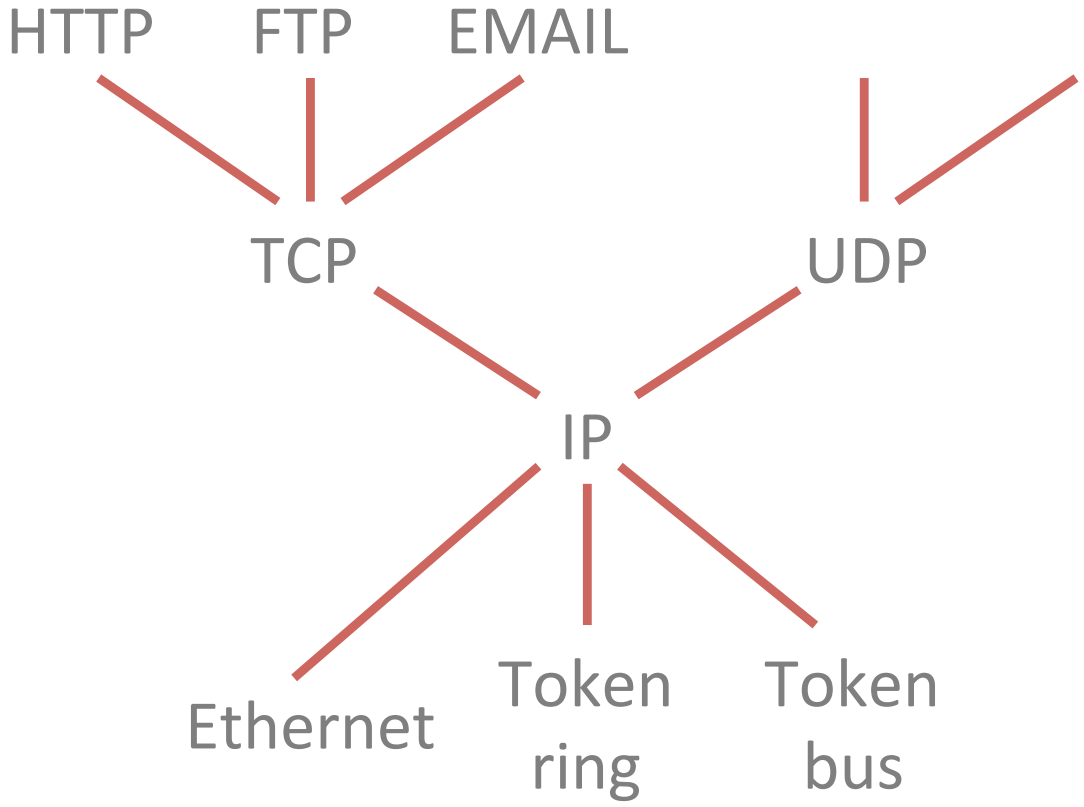
- Different link types will each have own
  - Protocol (access mechanisms, data formats, …)
  - Headers

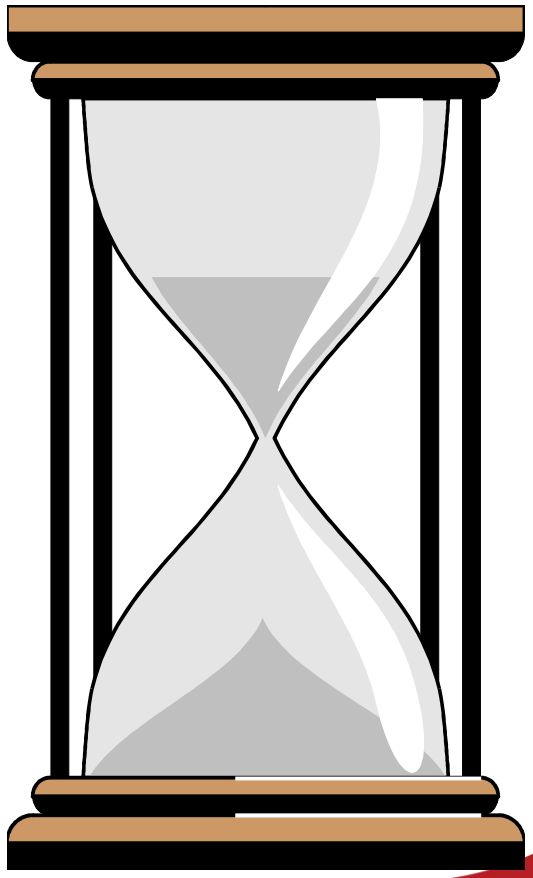


Hourglass Model (of the Internet)

Lancaster University

- Few **network** protocols
  - They are the 'glue' that allow inter-operation

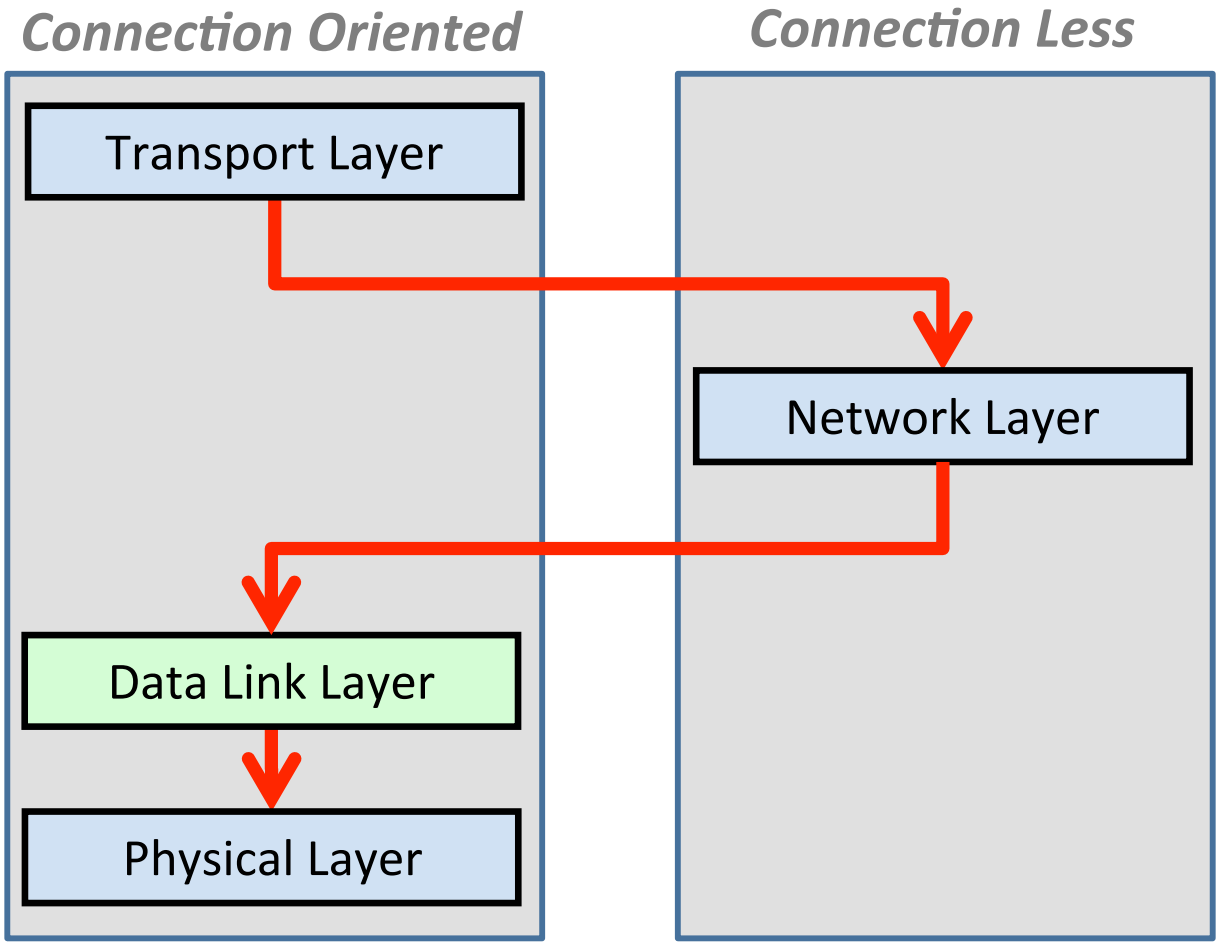




Connection Oriented vs. Connection Less

Lancaster University

- Can mix and match as we step through layers
  - Some combinations can be inefficient



OSI/RM

Lancaster University

- Application
- Presentation
- Session
- Transport
- Network
- Data Link
- Physical

} *Upper Layer Architecture*

} *Network Layers*

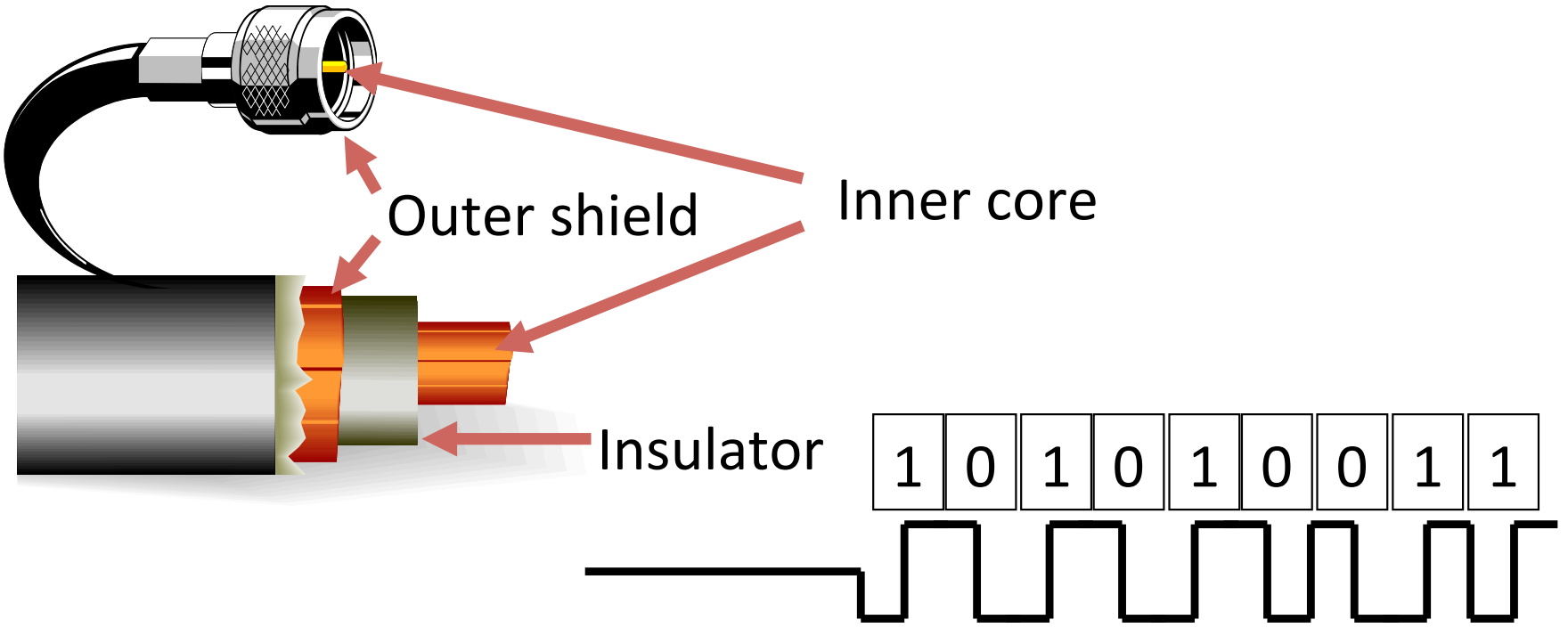
Lancaster University

Communication Layers

Overview of layers and what they represent

1. Physical Layer

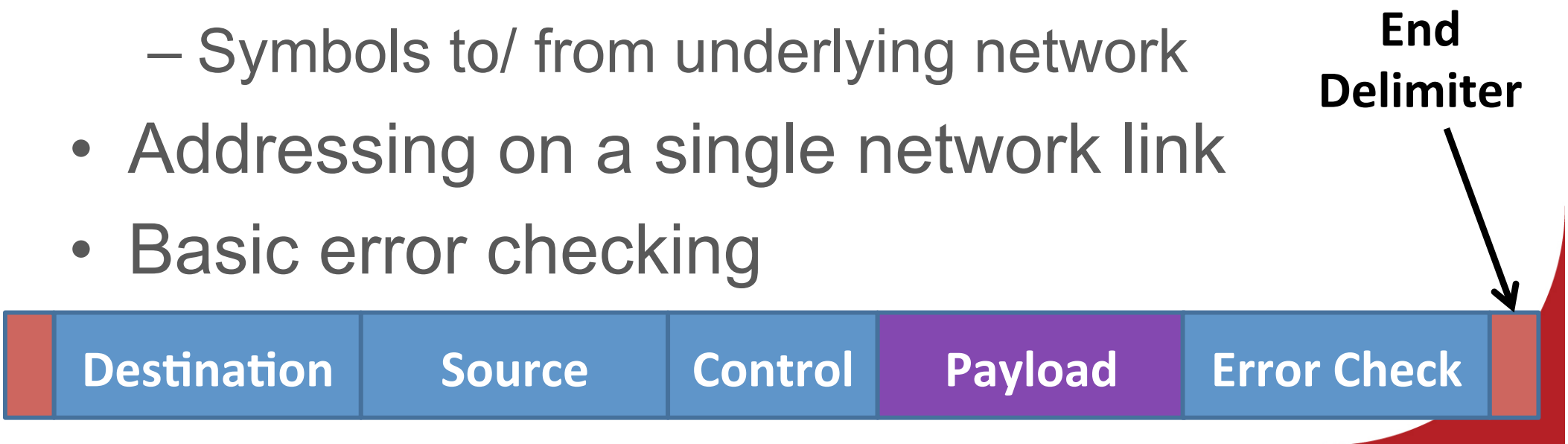
- Mechanical and electrical specification
- Plugs, sockets, wiring, bit transmission, etc.



2. Data Link Layer

Lancaster University

- Frame based transmission
  - Typically use frame delimiters marking start and end
- Sends and receives
  - Bytes to/ from host
  - Symbols to/ from underlying network
- Addressing on a single network link
- Basic error checking



IEEE 802/ ISO 8802 LAN Standards

Lancaster University

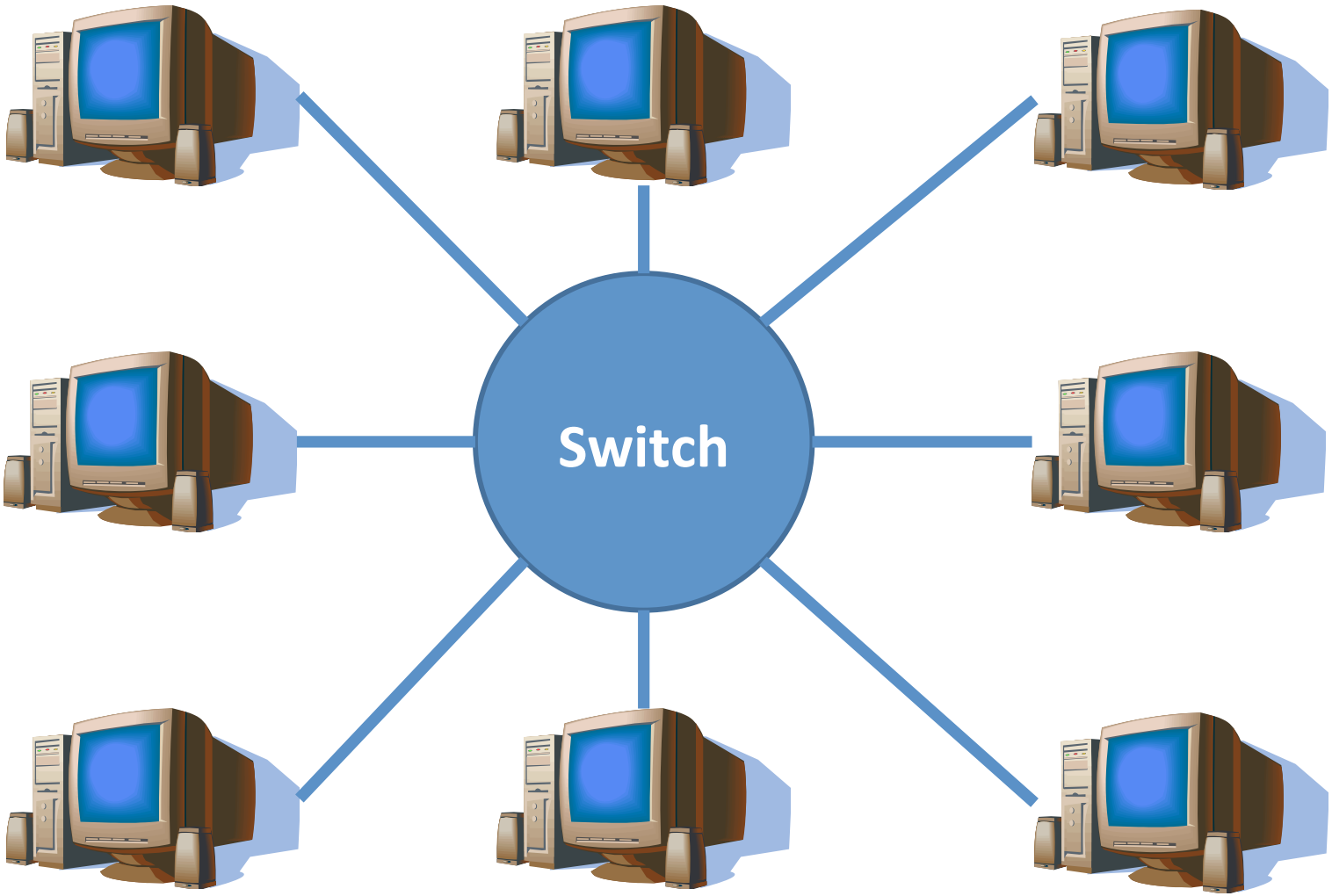
- Organised around a set of Working Groups

802.1	…	Higher Layer LAN Protocols
802.2	…	Logical Link Control (LLC) Layer
802.3	**…**	**Ethernet**
802.4	…	Token Bus
802.5	…	Token Ring
802.11	**…**	**Wireless LANs**
802.15	…	Wireless Personal Area Networks
802.16	…	Broadband Wireless Networks
802.22	…	Wireless Regional Area Networks
802.24	…	Smart Grid
etc.		

Simple Shared Network

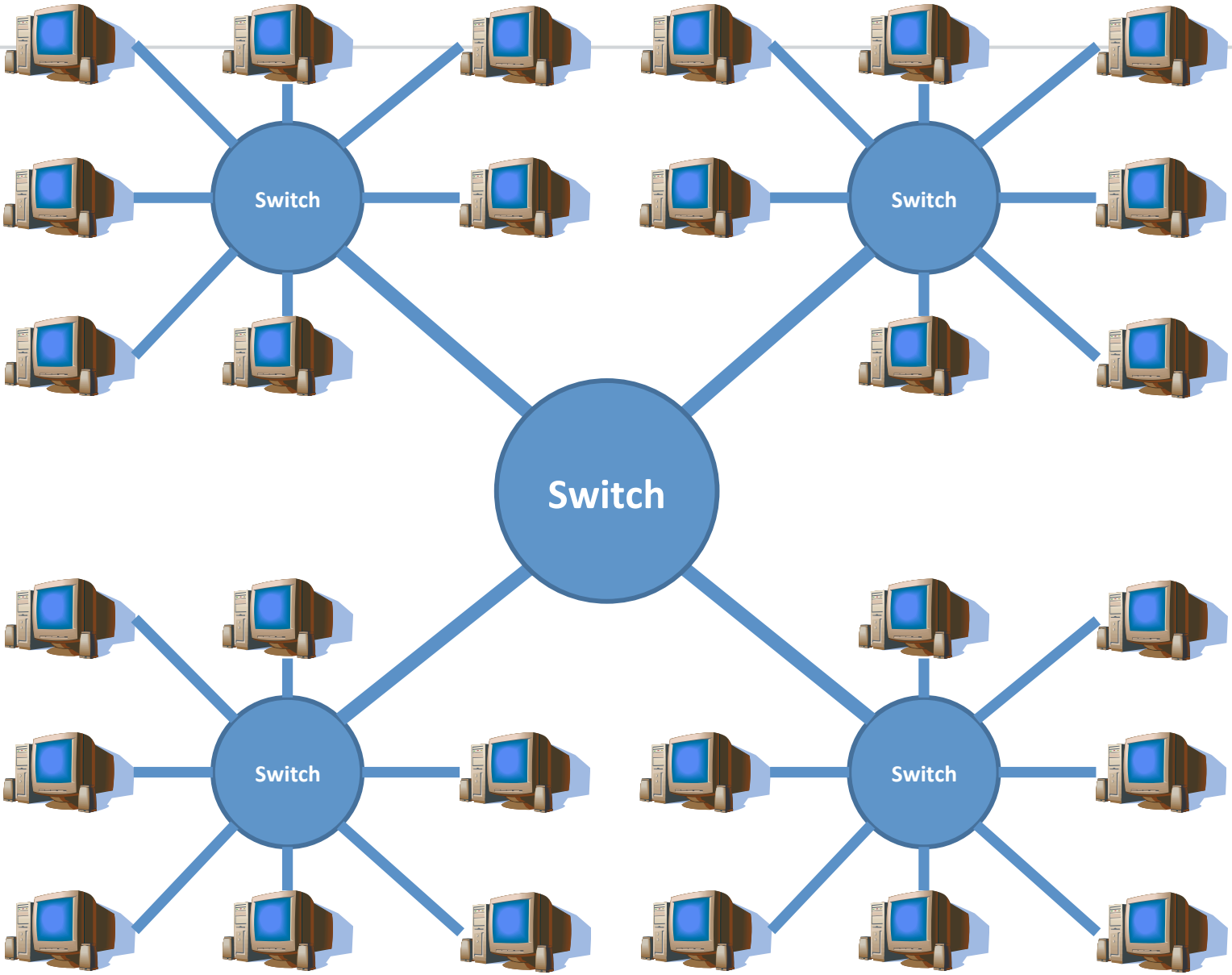
Lancaster University

- Switched network, all links running same link protocol



Unstructured, Shared Network

Lancaster University

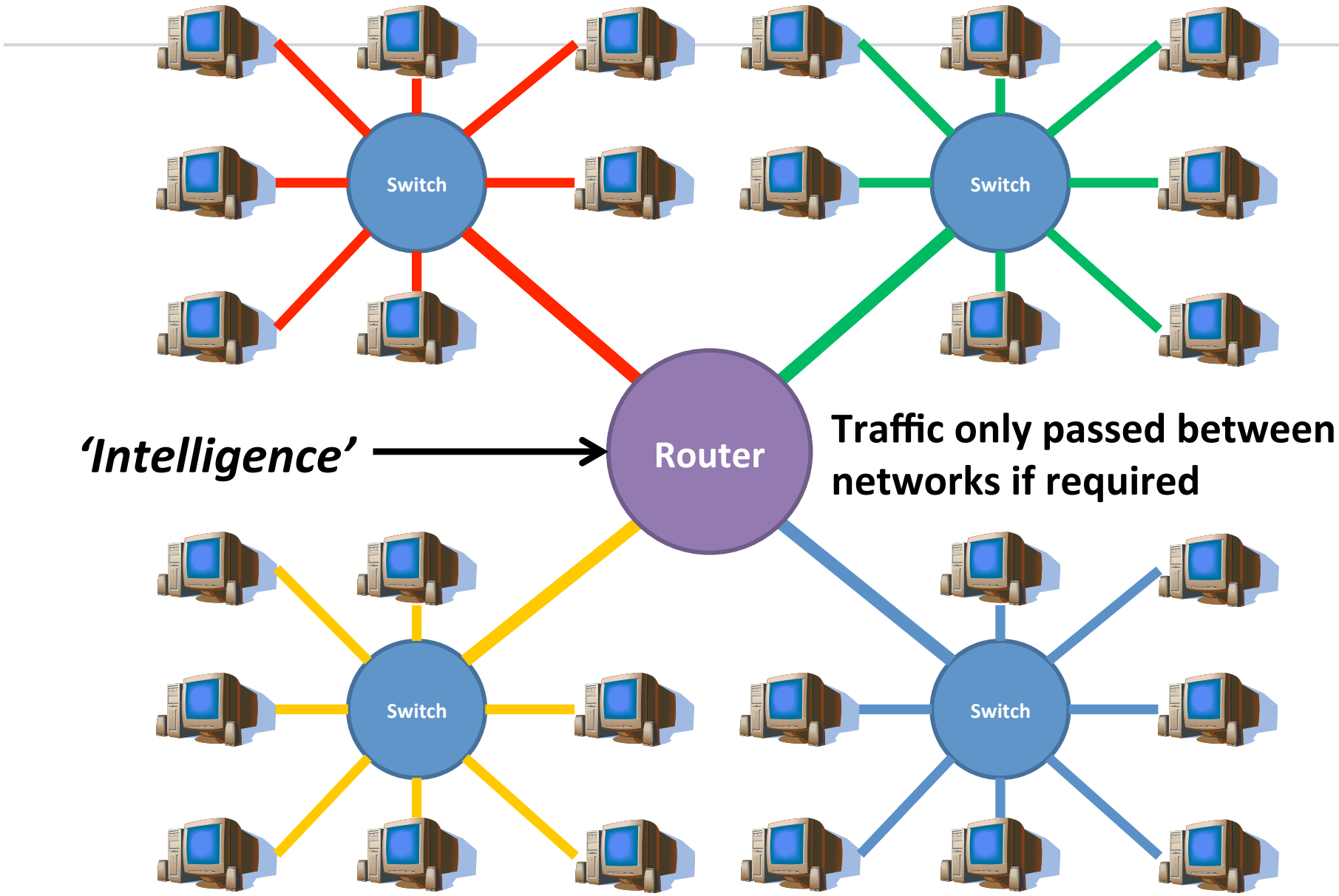


3. Network Layer

Lancaster University

- Network independent host-to-host transmission
  - Introduces network address
    - Allows global network to be partitioned into sub-networks
    - Separate/ distribute network management and traffic
  - Globally routable host addresses
    - Allows us to contact any public node, anywhere
- Incorporates mechanisms to
  - Determine path between sender and receiver
  - Establish and maintain network connections
  - Forward traffic

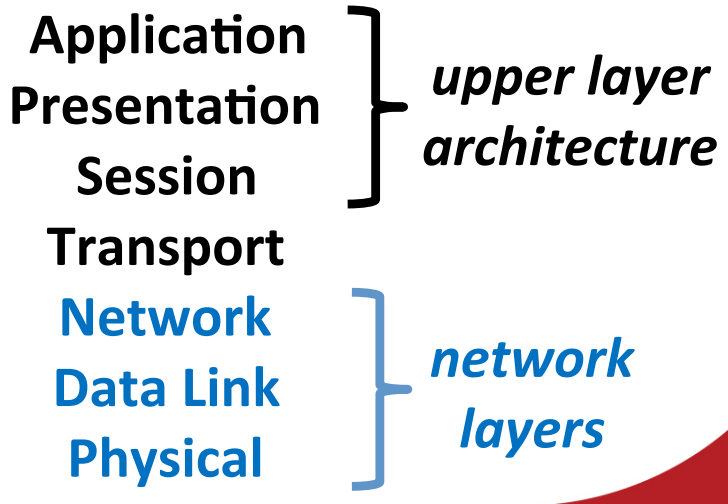
Routers Give Distinct Networks



Network Layers

Lancaster University

- Formed from Physical, Data Link, and Network layers
- Provide **network independent** message service
- Allow common service **across many network types**
- Hide network boundaries



4. Transport Layer

Lancaster University

- Offer ‘more useful’ services
- User oriented global addressing scheme
- Identify applications within machines
  - Process and application addressing
- Provides wider range of application oriented services
  - Connection-Oriented (CO)
  - Connection-Less (CL)

Upper Layer Architecture

Lancaster University

- Session, Presentation, Application layers
- Provide user oriented services
  - Far more tightly coupled than network layers
  - Often shared addresses, identifiers and variables
  - **Purely OSI!! LAN & Internet models not standardized at this level**

5. Session Layer

Lancaster University

- ‘Invented’ by ISO
- Very thin layer - little visible functionality
- Provides Value Added Services
  - Synchronization by defining major and minor sync. points
  - Activity and dialog management – notion of transactions
  - Graceful termination
- Intended as buffer against transport failure
  - Ability to restart a connection at agreed point in transfer

6. Presentation Layer

Lancaster University

- Data representation
  - Described in Abstract Syntax Notation (ASN.1)
- Compression
- Encryption
  - Can also be done at layers 1 - 4

7. Application Layer

Lancaster University

- Within context of Internet this is where we have:
  - Directory Service (DS) — DNS
  - File Transfer and Management (FTAM) — FTP
  - Message Handling Service (MHS) — Email
  - Virtual Terminal (VT) — Telnet

Internet Data Conventions

Lancaster University

- Transport layer: ***segments***
  - Formed from one or more packets
- Network layer: ***packets***
  - Formed from one or more frames
- Data link layer: ***frames***
  - Formed from 'one or more' bytes
- Physical: ***symbols***
  - Don't necessarily send one bit per time period, hence symbols

Recap: Main Protocol Layers

Lancaster University

- Physical Layer
  - How to physically and electrically connect devices
  - Data Link Layer (incl. Medium Access Control and Logical Link Control)
  - How and when to transmit each bit of data
  - How to transmit identifiable groups of bytes across a single network
- Network Layer
  - Common, network independent way of transmitting data host to host
- Transport Layer
  - Supports multiple data flows to each device
  - Connection-Oriented or Connection-Less styles of communication

Lancaster University

The Internet

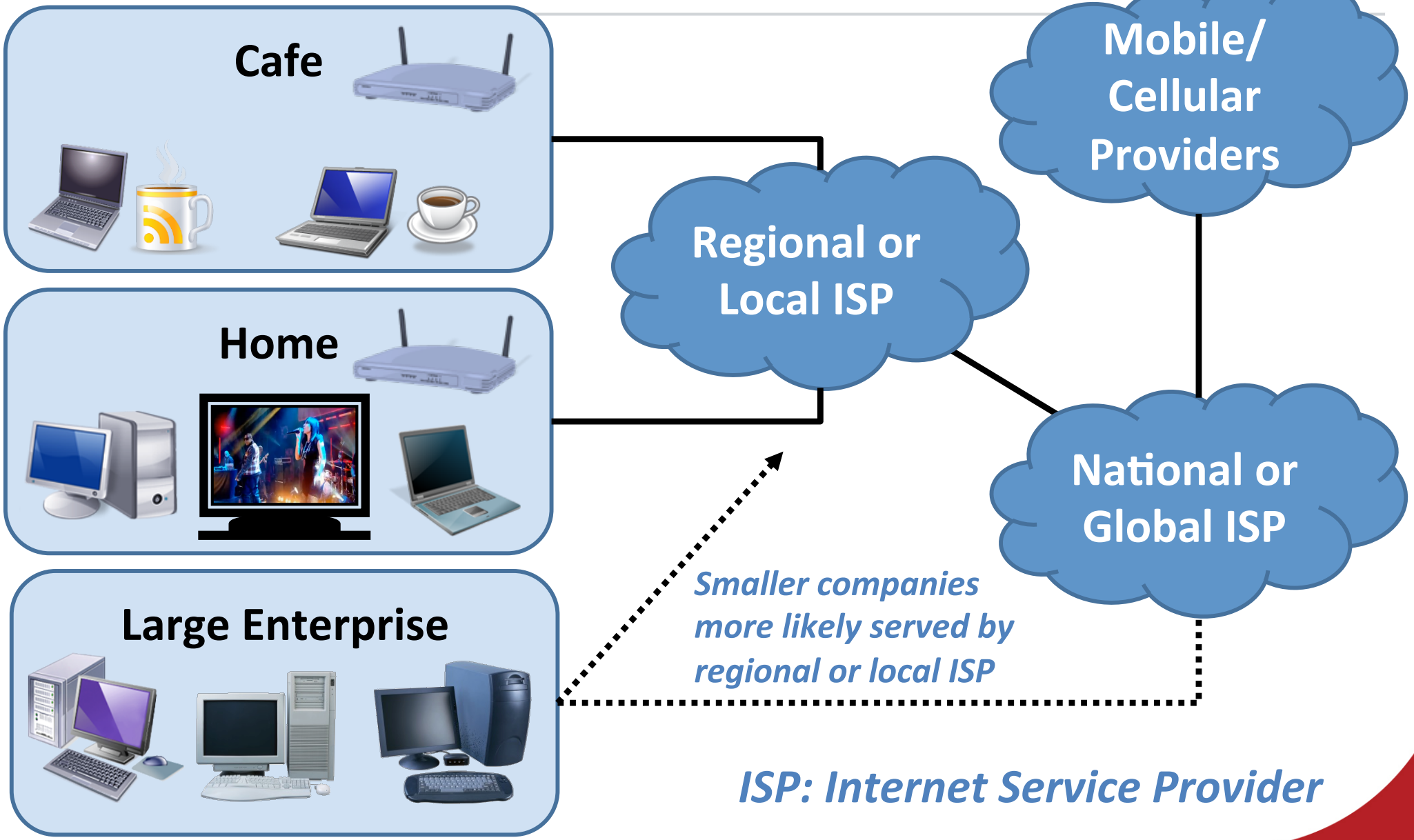
Probably the most prominent Protocol Stack in the World

Internet

Lancaster University

- Standardised by Internet Engineering Task Force
  - IETF Request for Comments (RFCs)
- Loosely hierarchical collection of networks
  - Network of networks
  - Best Effort service

High-Level View of Internet



Internet End-to-End Argument

Lancaster University

- Network will always be unreliable
  - Problems can occur as data enters host (or later)
- End systems must therefore perform
  - Error checks, data (re-)ordering, flow control
  - Little point in devices within network repeating these checks
  - Better to make them simple and efficient
  - Downside: problematic data may be sent a long way
- Network devices must always check addressing information
  - Misdirecting traffic unacceptable

Network becomes '*best effort*' service

Summary

Lancaster University

- Layering is used to separate communication functions and concerns
  - Each layer is responsible for different tasks
  - A lower layer offers a service to the layer above
- Reference models
  - Help us understand, categorise, compare, and describe protocols
  - Implementations
    - MUST provide expected interfaces and conform to them
    - Need not implement layers as distinct layers
- Issues with communication
  - Hour-glass model of the Internet
  - Connection-less vs. Connection Oriented Communication
    - Can be layered over each other
- Communication Layers
  - Overview over layers
    - From Physical to Application Layer
    - Different standards
    - Network structures
  - Upper layer architecture
    - OSI concept
      - Session, Presentation & Application Layers
- The Internet
  - Overview of structure
  - End-to-end argument

Summary

Lancaster University

- Relevant networking principle and concepts
  - Network structures, addressing, packet formats, communication modes

  ➔ These are abstractions and concepts on which computer communication is built!!
- Communication controlled by protocols
  - Tightly and completely defined
  - Control exactly what's transmitted, how, and when
    - Detail all states sender and receiver can be in
    - When and how to transition between states

Questions?

Lancaster University

Questions!!

Lancaster University

- Multiple Choice
  - Which is not a layer in the ISO-OSI model?
    - A) Security Layer; B) Physical Layer; C) Data Link Layer; D) Network Layer E) Transport Layer
  - Which of the following statements is true?
    - I) In a well designed system, the higher layer does not have to worry about the implementation details of lower layers; II) A layer offers a service to the next higher layer; III) Two entities of the same layer handle a protocol
  - Why does the Data Link Layer append a trailer to frames?
    - A) This is due to the router configuration.; B) The Data Link Layer runs always in reverse mode; Defined by IP.; The Data Link Layer only appends a header, but never a trailer; E) This way the checksum for error correction can be done “on the fly”.
  - Which layer does IP belong to?
    - A) Physical Layer; B) Data Link Layer; C) Network Layer; D) Transport Layer; E) Application Layer
- Explain the terms service and protocol in the context of the layer model
- What are the differences of the ISO-OSI Model to the TCP/IP Model?

Lancaster University

SCC203 Computer Networks: Local Area Networks

Dr Andreas Mauthe (a.mauthe@lancaster.ac.uk)
Dr Andrew Scott (a.scott@lancaster.ac.uk)

Contents

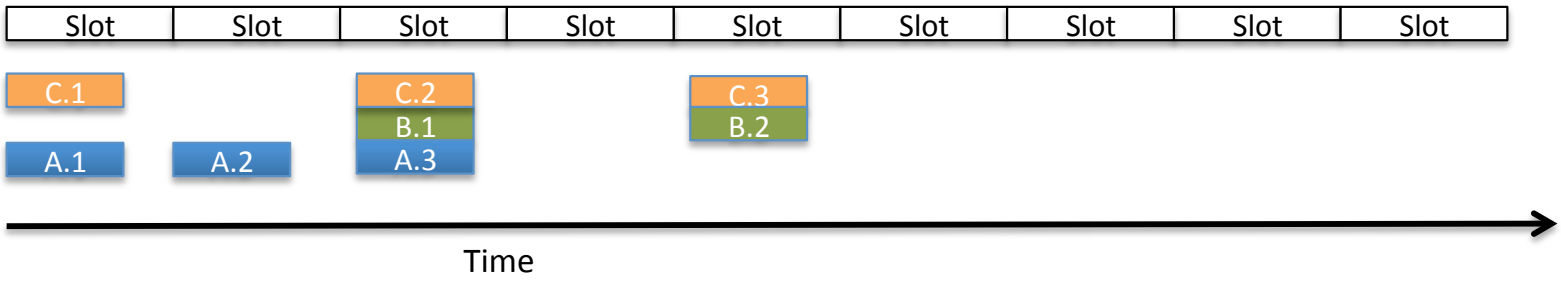
Lancaster University

- Outline
  - Multiple Access Links (cont.)
    - Transmission control
    - MAC protocols
      - Random Access Protocol, ALOHA, Slotted ALOAH, CSMA/CD, etc.
  - Ethernet
    - What are the basics elements
      - Topology, frame structure, service types
    - Switched Ethernet and how it works
      - Procedures
      - Self-learning forwarding tables
  - Other LINK Layer Protocols
  - VLAN – a means to structure communication over the same infrastructure
    - Port-based VLAN, Multiple Switches, VLAN Frame Format
- Objectives
  - To understand and apply the concepts realising the main functions of Link Layer Access Control
    - You should understand how Multiple Access Protocol works, be able to describe pros and cons, how they achieve their objectives, and apply you knowledge to a given problem.
  - To learn how Ethernet works, specifically the switched mode.
    - You should know what characteristics of Ethernet are and how frames are sent and received over Ethernet.
  - To get familiar with the principles of Virtual Local Area Network
    - You should know about the motivation for VLAN and be familiar of some of the key functional aspects of LAN virtualisation

Questions!!

Lancaster University

- Multiple Choice
  - In which layer is CRC used for error detection?
    - A) Physical Layer B) Correction Layer C) Data Link Layer D) Network Layer E) Transport Layer
  - Which of these encoding mechanisms are not “self-clocking”?
    - I) Manchester encoding, II) Differential Manchester encoding, III) Binary encoding, IV) Nonreturn to zero
- Which are the functions of the data link layer?
- Show how the following packets are transmitted using synchronous and asynchronous TDM. Note, if packets arrive at the same time, process them in alphabetical order



Lancaster University

Multiple Access Control Protocols (cont.)

or ALOHA is not just a Hawaiian Greeting

Multiple access protocols

Lancaster University

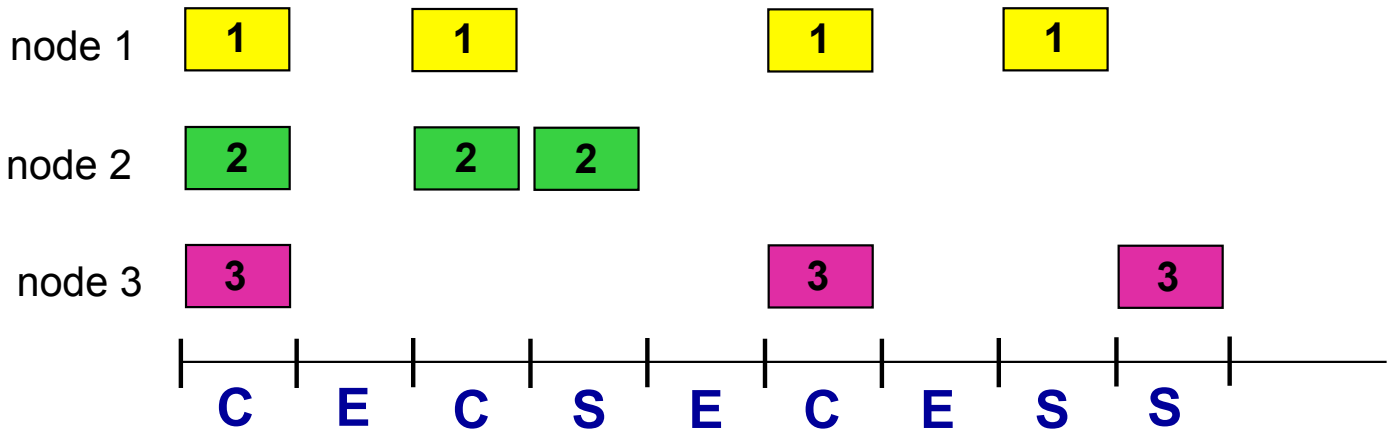
- Single shared broadcast channel
- Two or more simultaneous transmissions by nodes:
  - Collision if node receives two or more signals at the same time
- Multiple access protocol
  - Distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
  - Communication about channel sharing must use channel itself!
    - No out-of-band channel for coordination

Slotted ALOHA

Lancaster University

- Assumptions:
  - All frames same size
  - Time divided into equal size slots (time to transmit 1 frame)
  - Nodes start to transmit only at the beginning of slots
  - Nodes are synchronized (i.e. know when slot begins)
  - If 2 or more nodes transmit in slot, all nodes detect collision before slot ends
- Operation:
  - When node obtains fresh frame, waits for beginning of next slot, and transmits
    - If no collision: success!, and node can send new frame in next slot
    - If collision: node retransmits frame in each subsequent slot with prob. p until success

Slotted ALOHA



- Pros:
  - Single active node can continuously transmit at full rate of channel
  - Highly decentralized: only slots in nodes need to be in sync
  - Simple
- Cons:
  - Collisions, wasting slots
  - Idle slots
  - Nodes may be able to detect collision in less than time to transmit packet
  - Clock synchronization

Slotted ALOHA: efficiency

Lancaster University

efficiency: *long-run fraction of successful slots (many nodes, all with many frames to send)*

- Suppose: N nodes with many frames to send, each transmits in slot with probability p
- Prob that given node has success in a slot = p(1-p)N-1
- Prob that any node has a success = Np(1-p)N-1

- Max efficiency: find p* that maximizes Np(1-p)N-1
- For many nodes, take limit of Np*(1-p*)N-1 as N goes to infinity, gives:
- max efficiency = 1/e = .37

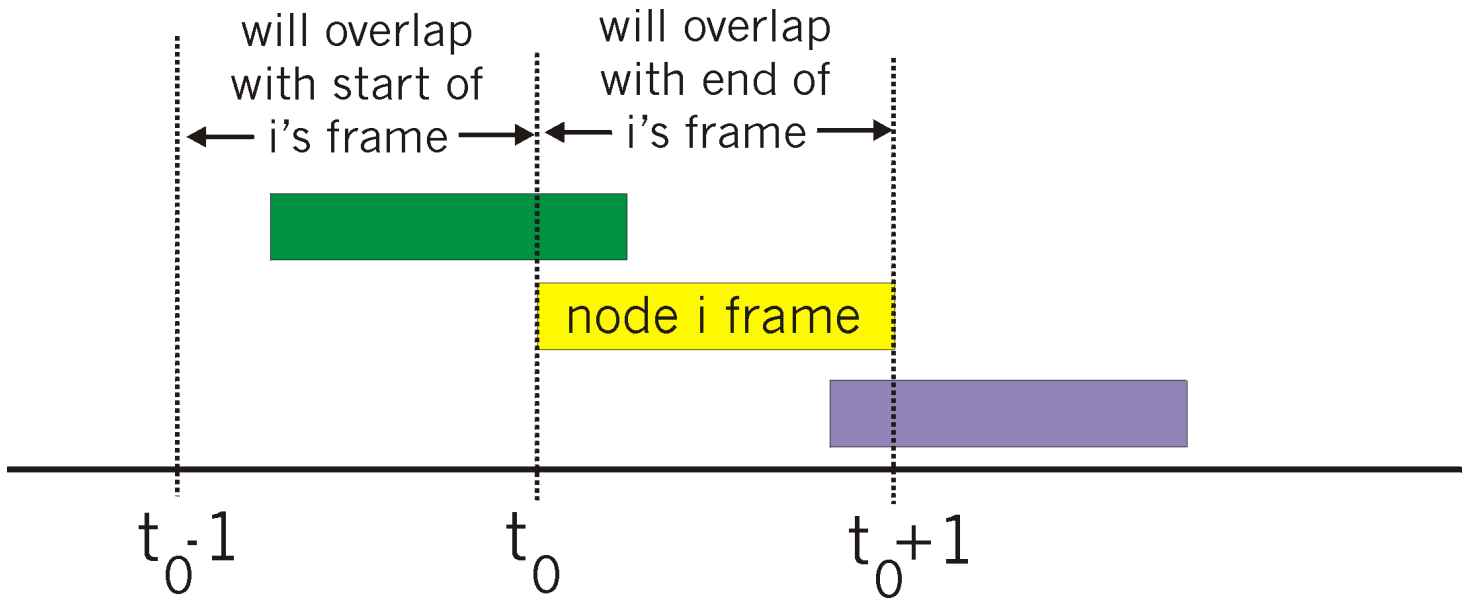
at best: channel used for useful transmissions 37% of time!

!

Pure (unslotted) ALOHA

Lancaster University

- Earlier version of the protocol - 1970
- No synchronization required
- When frame first arrives: transmit immediately
- Collision probability increases:
  - frame sent at t0 collides with other frames sent in [t0-1,t0+1]
- Efficiency about half that of slotted ALOHA



CSMA (carrier sense multiple access)

Lancaster University

- Carrier
  - Signal used to carry data
  - Presence indicates that some device is transmitting, or is about to transmit
- CSMA: listen before transmit:
  - if channel sensed idle: transmit entire frame
  - if channel sensed busy, defer transmission
- CSMA/CD
  - Carrier Sense, Multiple Access with Collision Detection
- Human analogy:
  - Listen before speaking
  - If someone else begins talking at the same time – stop talking!

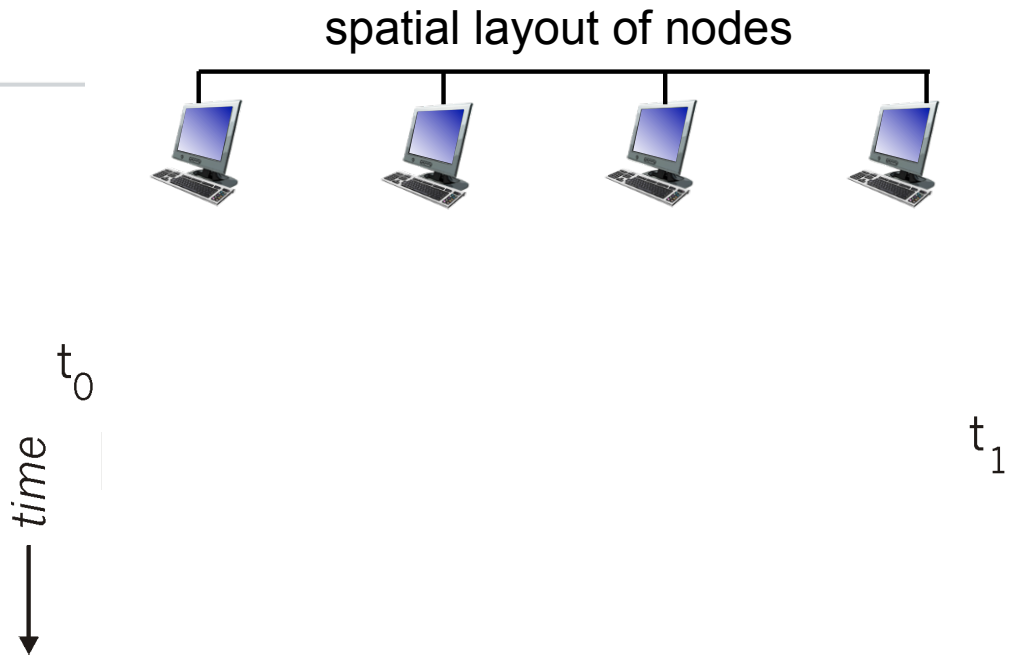
Question?

Lancaster University

If all nodes perform carrier sensing, why would collisions still occur?

CSMA collisions

- Collisions can still occur since the propagation delay means two nodes may not hear each other's transmission
- Collision mean that the entire packet transmission time wasted
  - distance & propagation delay play key role in determining collision probability



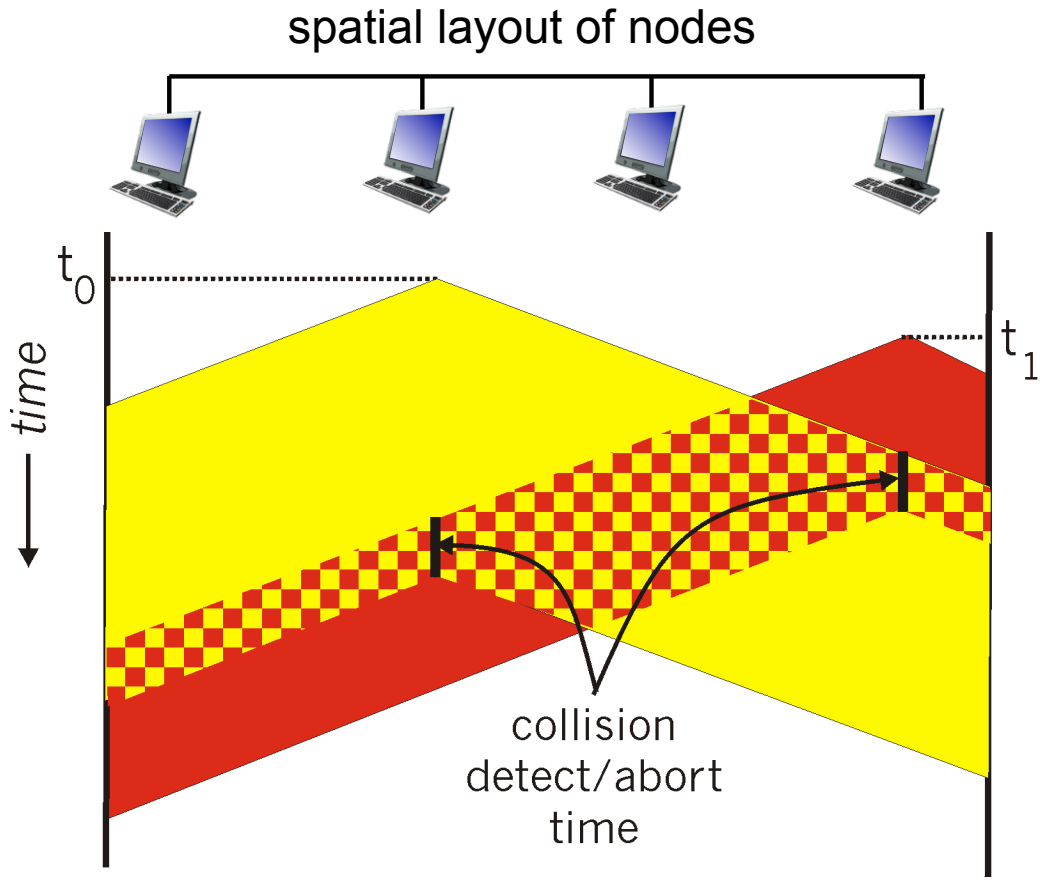
CSMA/CD (collision detection)

Lancaster University

- CSMA/CD: carrier sensing, deferral as in CSMA
  - Collisions detected within short time
  - Colliding transmissions aborted, reducing channel wastage
- Collision detection:
  - Easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - Difficult in wireless LANs: received signal strength overwhelmed by local transmission strength

CSMA/CD (collision detection)

Lancaster University



Ethernet CSMA/CD algorithm

Lancaster University

- 1. Network Interface Controller (NIC) receives datagram from network layer, creates frame
- 2. If NIC senses channel idle, starts frame transmission. If NIC senses channel busy, waits until channel idle, then transmits.
- 3. If NIC transmits entire frame without detecting another transmission, NIC is done with frame !
- 4. If NIC detects another transmission while transmitting, aborts and sends jam signal
- 5. After aborting, NIC enters *binary (exponential) backoff*:
  - NIC waits random amount of time, returns to Step 2
  - Longer backoff interval with more collisions

Binary Exponential Back-off

Lancaster University

- After each respective collision wait:
  - 0 or 51.2 μs (chosen at random)
  - 0, 51.2, 102.4 or 153.6 μs (equal probability)
  - 0, 51.2, 102.4, 153.6, … 51.2 x (23 – 1) μs
  - 0, 51.2, 102.4, 153.6, … 51.2 x (2n – 1) μs
    - Maximum value of n is capped at 10
- Size of the sets from which a wait is chosen grows exponentially with number of collisions

Note: The figure of 51.2 microseconds represents time needed to send 512 bits into a 10 Mbps Ethernet – would be 5.12 microseconds for 100 Mbps

Lancaster University

Ethernet

Most commonly used Link Protocol

Ethernet

Lancaster University

- Based on experimental ALOHA network
  - Not slotted Aloha
- 10Mbps CSMA/CD network

- IEEE 802.3
  - Covers any 1 - 10Mbps CSMA/CD network
  - Different variations of Ethernet
  - Variants include
    - IEEE 802.3u: 100Mbps Fast Ethernet
    - IEEE 802.3z: 1Gbps Ethernet
    - IEEE 802.3ae: 10Gbps Ethernet

Ethernet ‘Base’ Notation

Lancaster University

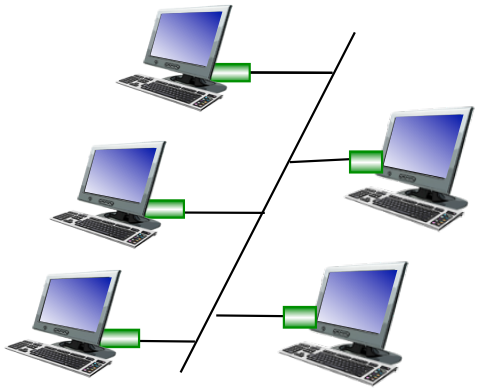
Speed Base-Cable Type / Distance (in 100 m)

Name	Cable	Run	Mbps	
10Base-5	Coax	500m	10	Thick Ethernet
10Base-2	Coax	200m	10	Thin Ethernet “Cheapernet”
10Base-T	U/STP	100m	10	
100Base-T4	UTP	100m	100	Cat 3 using 4 pairs
100Base-TX	UTP	100m	100	Cat 5 Full duplex
100Base-FX	Fibre	2km	100	Full duplex, multimode fibre
1000Base-SX	Fibre	550m	1000	Multimode fibre
1000Base-LX	Fibre	5Km	1000	Single or Multimode fibre
1000Base-CX	STP	25m	1000	2 pairs STP
1000Base-T	UTP	100m	1000	4 pairs Cat 5e UTP

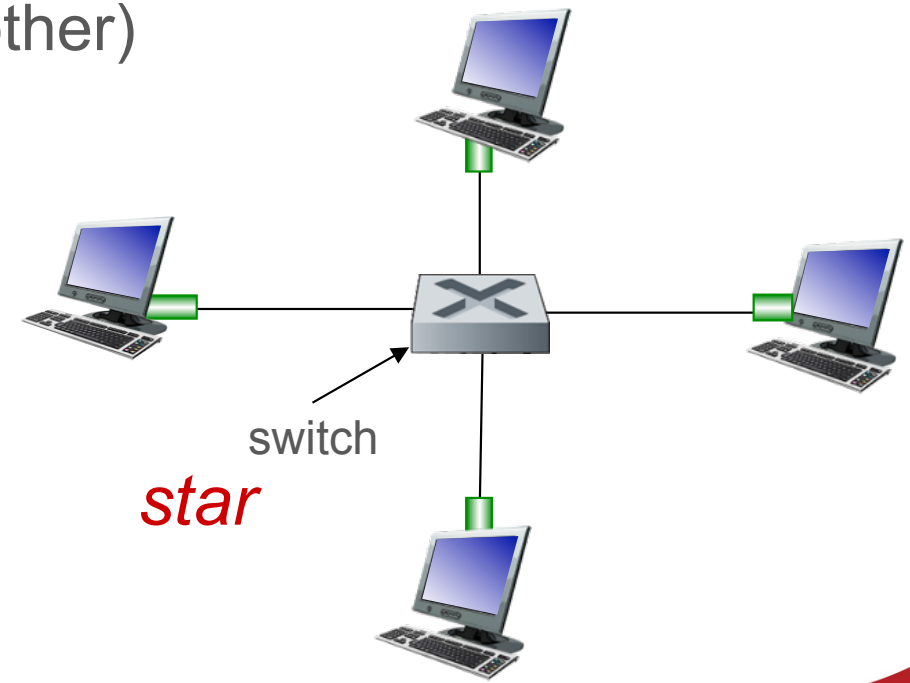
Ethernet: Physical Topology

Lancaster University

- Bus: popular through mid 90s
  - All nodes in same collision domain (can collide with each other)
- Star: prevails today
  - Active switch in center
  - Each "spoke" runs a (separate) Ethernet protocol (nodes do not collide with each other)



bus: coaxial cable



star

Ethernet Frame Structure

Lancaster University

- Sending adapter encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame

- Preamble:
  - 7 bytes with pattern 10101010 followed by one byte with pattern 10101011
  - Used to synchronize receiver, sender clock rates

Ethernet Frame Structure (cont.)

Lancaster University

- Addresses: 6 byte source, destination MAC addresses
  - If adapter receives frame with matching destination address, or with broadcast address (e.g. ARP packet), it passes data in frame to network layer protocol
  - Otherwise, adapter discards frame
- Type: indicates higher layer protocol (mostly IP but others possible, e.g., Novell IPX, AppleTalk)
- CRC: cyclic redundancy check at receiver
  - Error detected: frame is dropped

IEEE 802.3 Frame Format

Lancaster University

	Bytes
Preamble	7
Start of Frame	1
Destination	2 or 6
Source	2 or 6
Data Length	2
Data	0-1500
Padding	0-46
Checksum	4

10101…1010 to synchronize clocks

10101011

Ethernet always 6 uses bytes

Addresses:

top bit	= 0	→ unicast
	= 1	→ multicast
111….111		→ broadcast

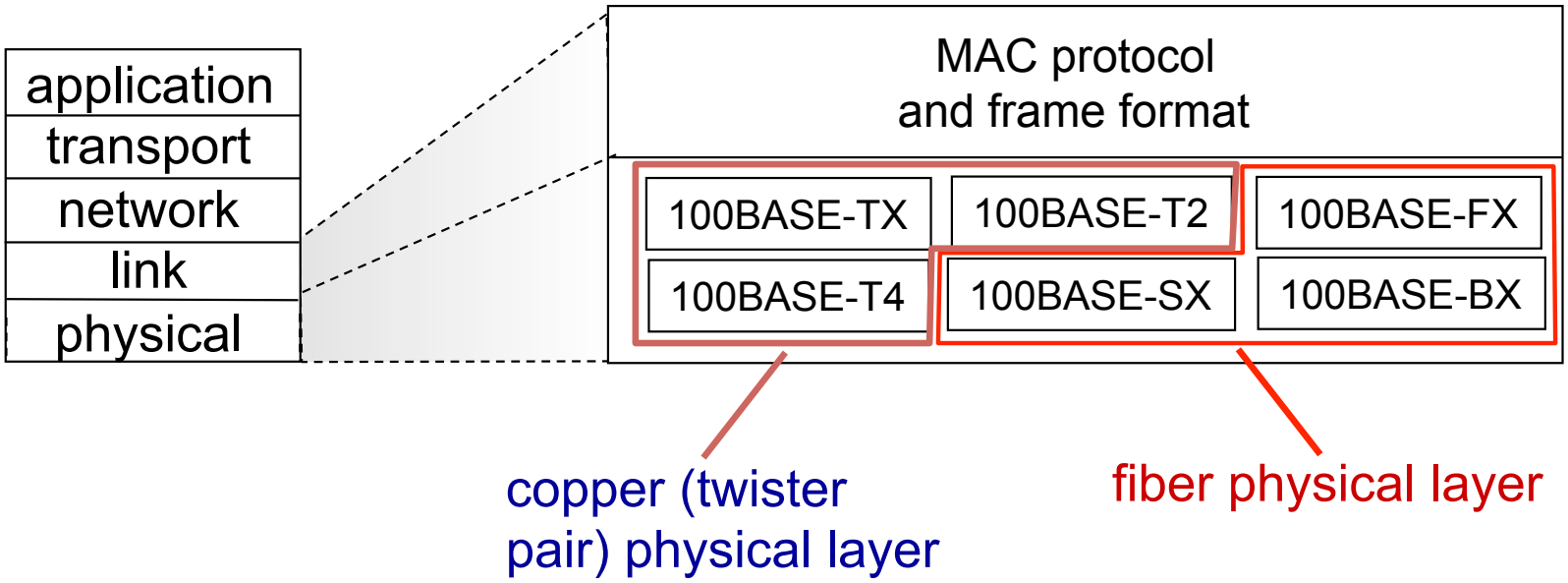
Ethernet: unreliable, connectionless

Lancaster University

- Connectionless: no handshaking between sending and receiving NICs
- Unreliable: receiving NIC doesn't send Acks or Nacks to sending NIC
  - Data in dropped frames recovered only if initial sender uses higher layer protocol (e.g., TCP), otherwise dropped data lost
- Ethernet's MAC protocol: unslotted CSMA/CD with binary backoff

802.3 Ethernet Standards: Link & Physical Layers

- Many different Ethernet standards
  - Common MAC protocol and frame format
  - Different speeds: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10G bps
  - Different physical layer media: fiber, cable



Lancaster University

Switches

How to relay frames at the Link Layer

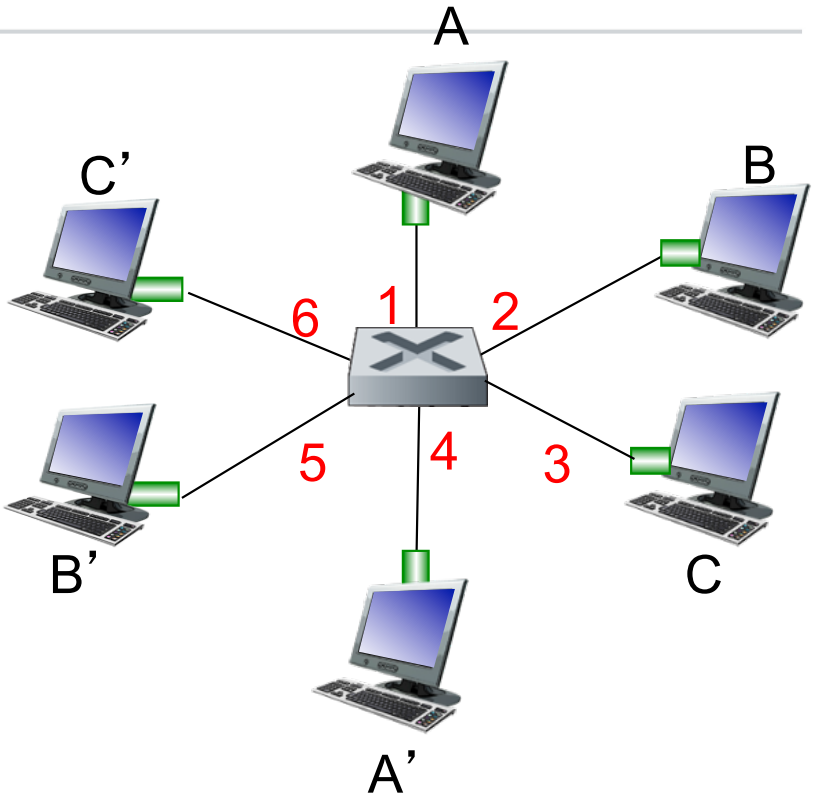
Ethernet Switch

Lancaster University

- Link Layer device: takes an active role
  - Store, forward Ethernet frames
  - Examine incoming frame's MAC address, selectively forward frame to one-or-more outgoing links when frame is to be forwarded on segment, uses CSMA/CD to access segment
- Transparent
  - Hosts are unaware of presence of switches
- Plug-and-play, self-learning
  - Switches do not need to be configured
  - Do not have MAC addresses

Switch: *Multiple* Simultaneous Transmissions

- Hosts have dedicated, direct connection to switch
- Switches buffer packets
- Ethernet protocol used on each incoming link, but no collisions; full duplex
  - Each link is its own collision domain
- Switching: A-to-A' and B-to-B' can transmit simultaneously, without collisions

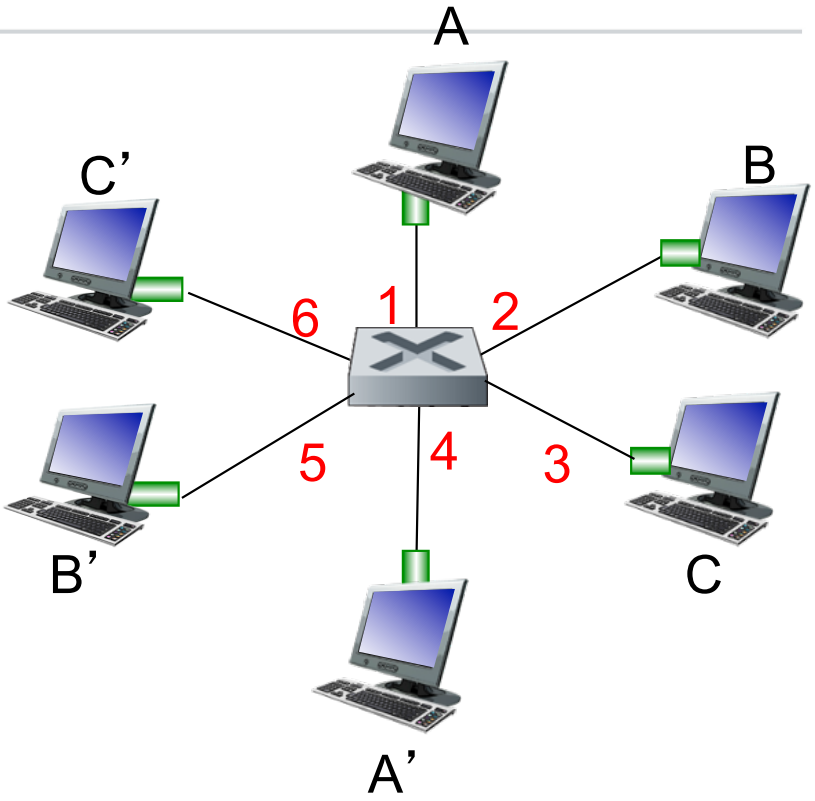


switch with six interfaces (1,2,3,4,5,6)

Switch Forwarding Table

Lancaster University

- Q: how does switch know A' reachable via interface 4, B' reachable via interface 5?
  - A: each switch has a switch table, each entry:
    - (MAC address of host, interface to reach host, time stamp)
    - looks like a routing table!
  - Q: how are entries created, maintained in switch table?
    - something like a routing protocol?



switch with six interfaces (1,2,3,4,5,6)

Switch: Self-Learning

- switch learns which hosts can be reached through which interfaces
  - When frame received, switch "learns" location of sender: incoming LAN segment
  - Records sender/ location pair in switch table

MAC addr	interface	TTL
A	*1*	*60*

Switch table (initially empty)

Switch: Frame Filtering/ Forwarding

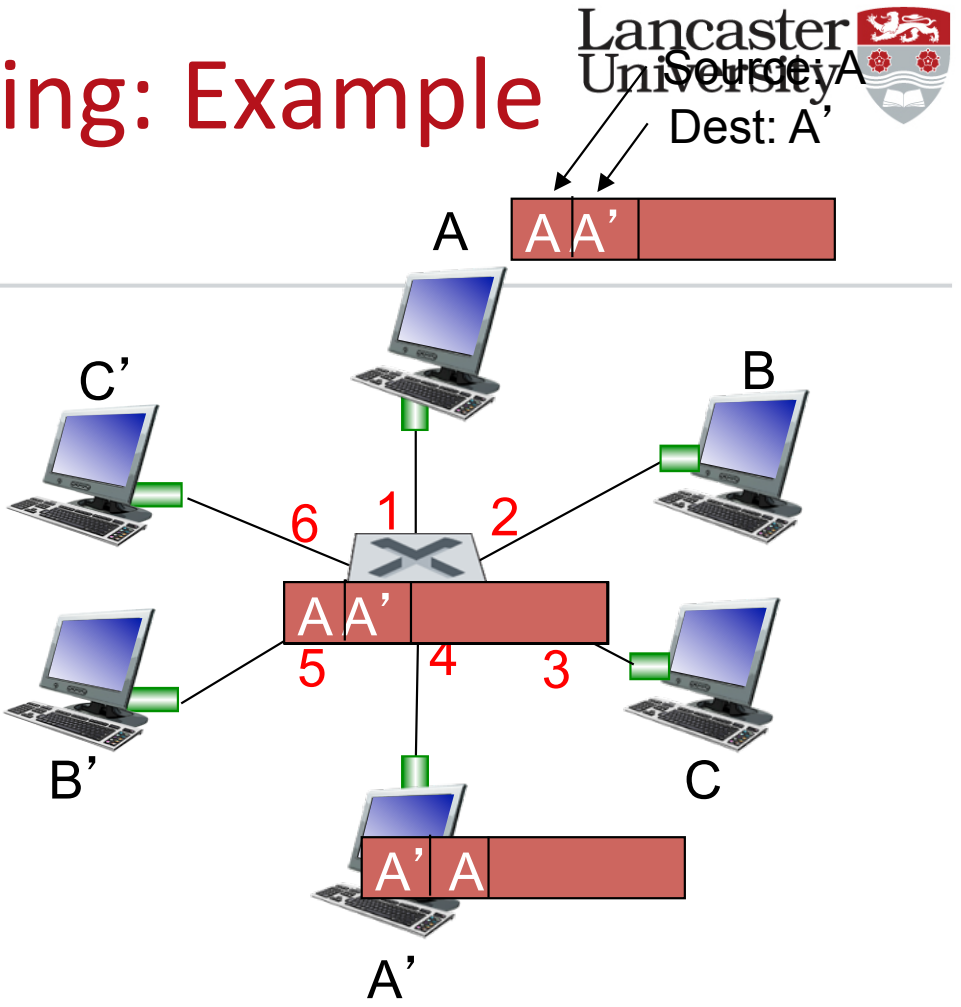
Lancaster University

```
when  frame received at switch:

    1. record incoming link, MAC address of sending host
    2. index switch table using MAC destination address
    3. if entry found for destination
       then {
              if destination on segment from which frame arrived
              then drop frame
              else forward frame on interface indicated by entry
             }
       else flood  /* forward on all interfaces except
                              arriving interface */
```

-

Self-learning, Forwarding: Example



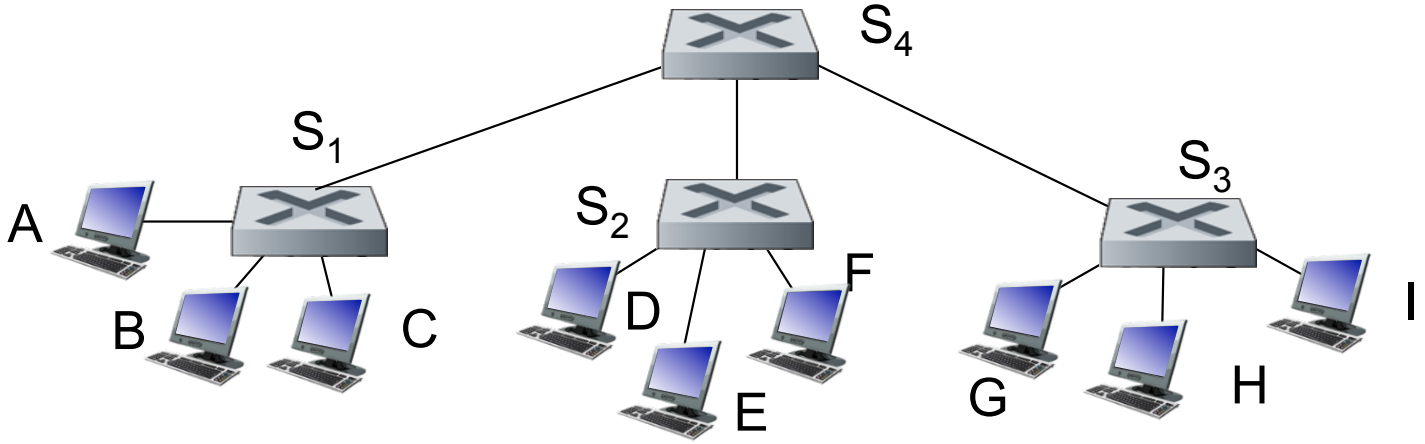
- Frame destination, A', location unknown: *flood*
- Destination A location known: *selectively send on just one link*

MAC addr	interface	TTL
A	*1*	*60*
A'	*4*	*60*

switch table (initially empty)

Interconnecting switches

- switches can be connected together



Q: sending from A to G - how does S1 know to forward frame destined to G via S4 and S3?

A: self learning! (works exactly the same as in single-switch case!)

Lancaster University

Other Link Layer Protocols

Not so much used... but still clever

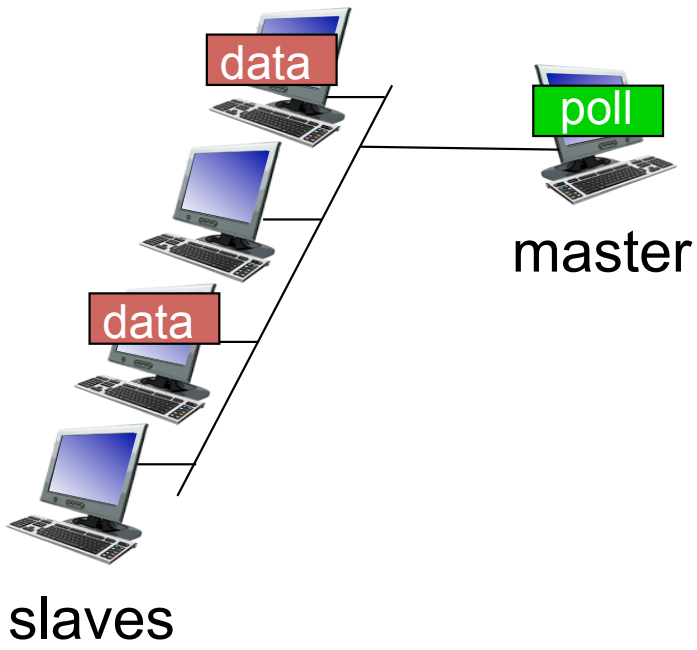
“Taking turns” MAC protocols

Lancaster University

- Channel partitioning MAC protocols:
  - Share channel efficiently and fairly at high load
  - Inefficient at low load: delay in channel access, 1/N bandwidth allocated even if only 1 active node!
- Random access MAC protocols
  - Efficient at low load: single node can fully utilize channel
  - High load: collision overhead
- “Taking turns” protocols
  - Look for best of both worlds!

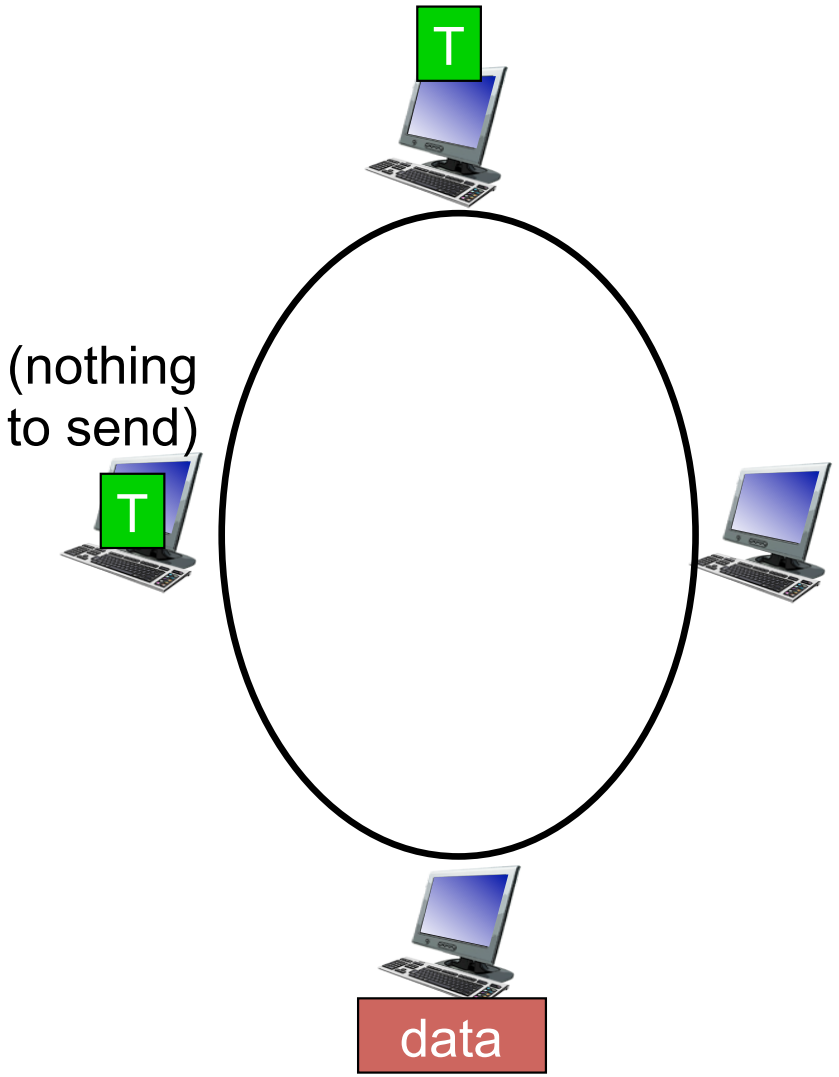
“Taking turns” MAC protocols (cont.)

- Polling:
  - Master node “invites” slave nodes to transmit in turn (round robin)
  - Efficient approach
  - Typically used with “dumb” slave devices
  - Concerns:
    - Polling overhead
    - Latency
    - Single point of failure (master)
- 802.15 and Bluetooth



“Taking turns” MAC protocols

- Token passing:
  - Control token passed from one node to next sequentially.
  - Token message
  - Concerns:
    - Token overhead
    - Latency
    - Single point of failure (token)
  - 802.5 token ring

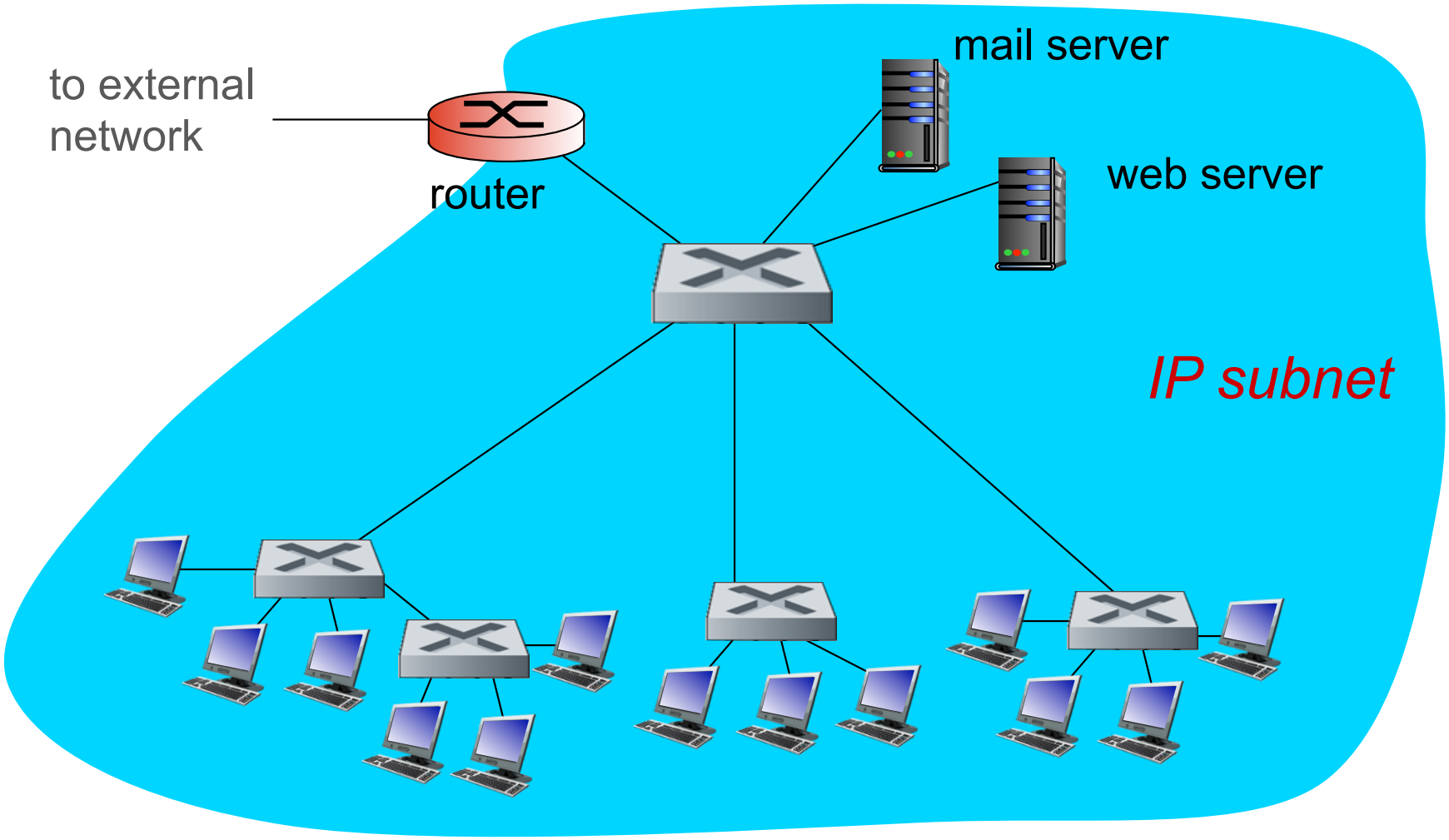


Lancaster University

Virtual Local Area Networks

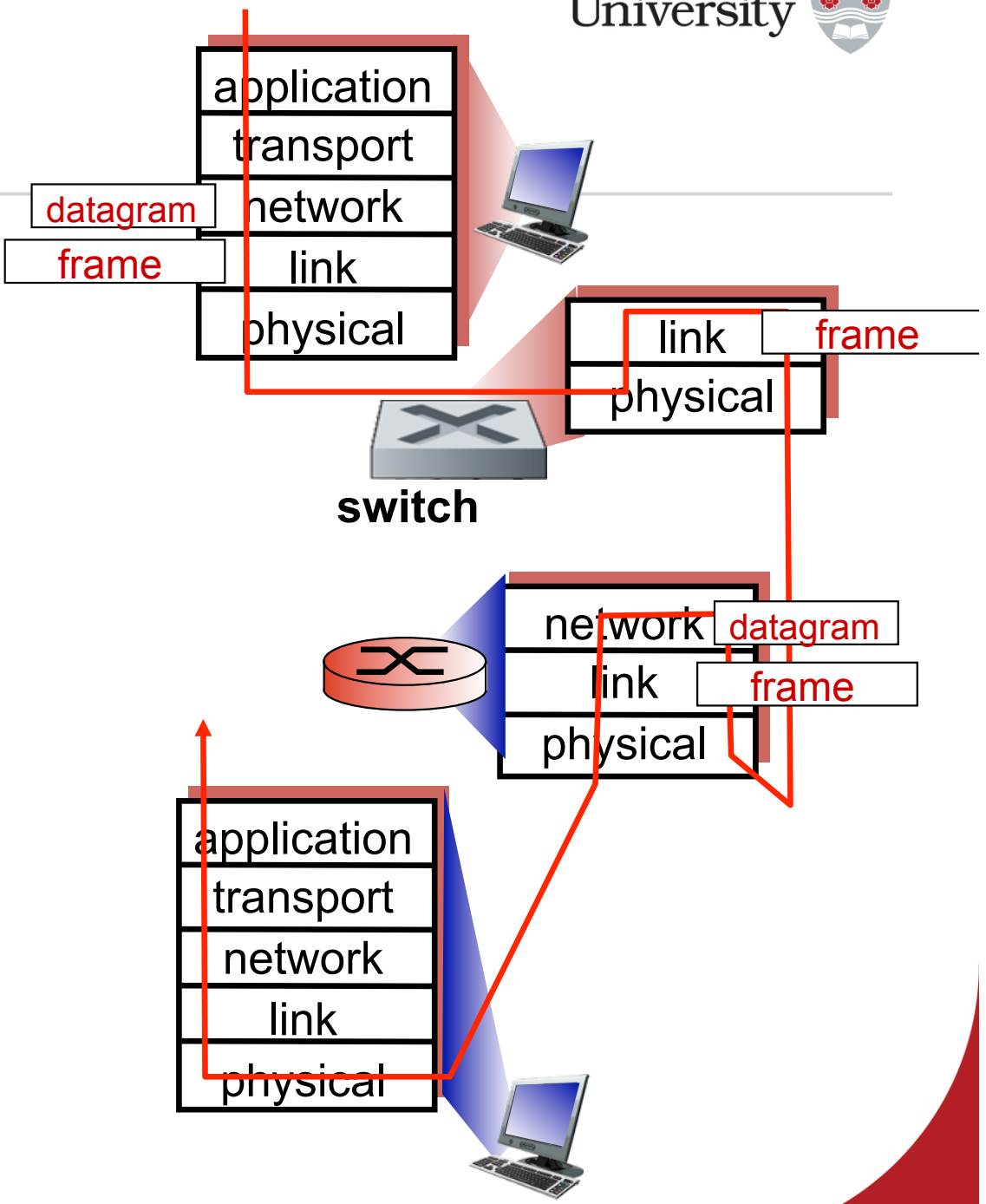
... or how an I have my own network without having a network

Institutional network

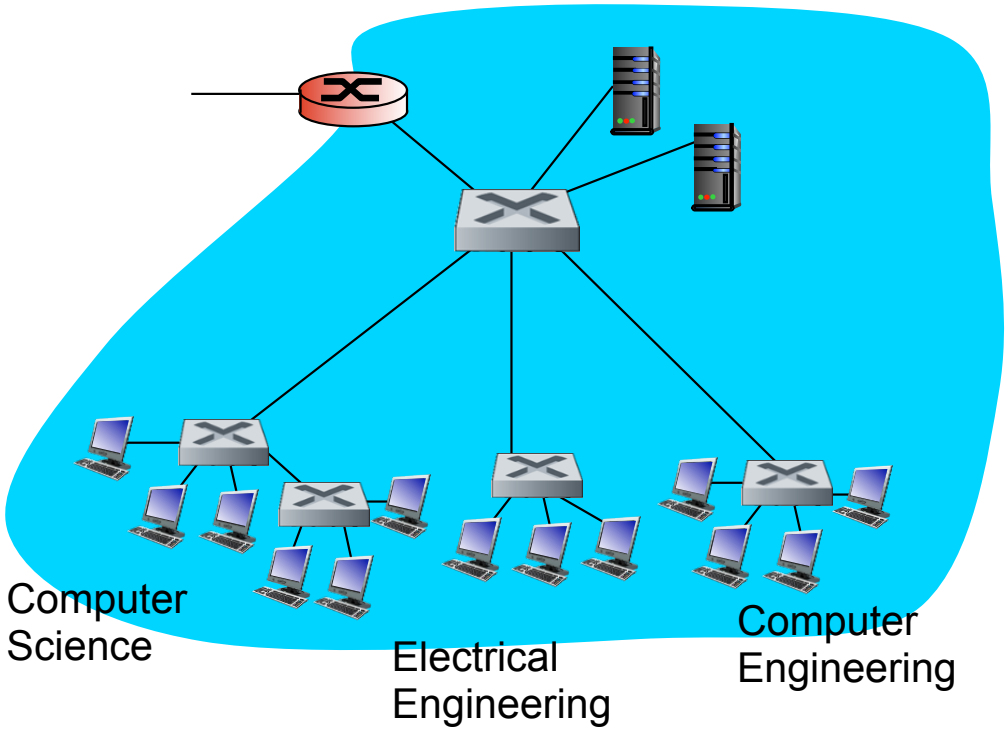


Switches vs. routers

- Both are store-and-forward:
  - Routers: network-layer devices (examine network-layer headers)
  - Switches: link-layer devices (examine link-layer headers)
- Both have forwarding tables:
  - Routers: compute tables using routing algorithms, IP addresses
  - Switches: learn forwarding table using flooding, learning, MAC addresses



VLANs: motivation

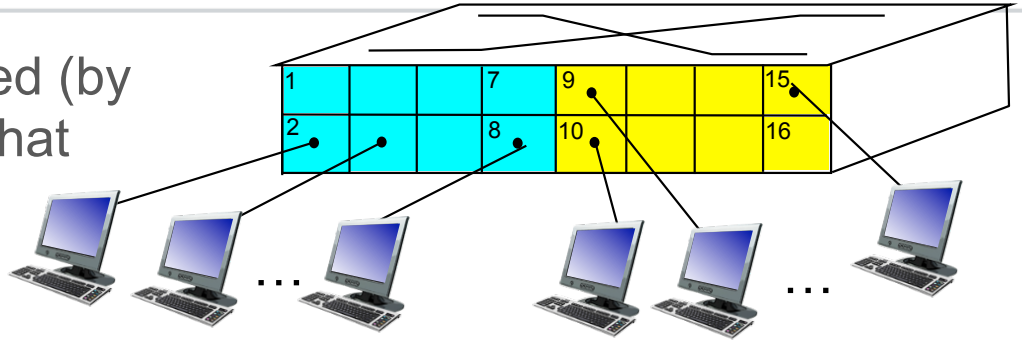


- Scenario:
  - CS user moves office to EE, but wants to connect to CS switch?
  - single broadcast domain:
    - All layer-2 broadcast traffic (ARP, DHCP, unknown location of destination MAC address) must cross entire LAN
    - Security/privacy, efficiency issues

VLANs

Lancaster University

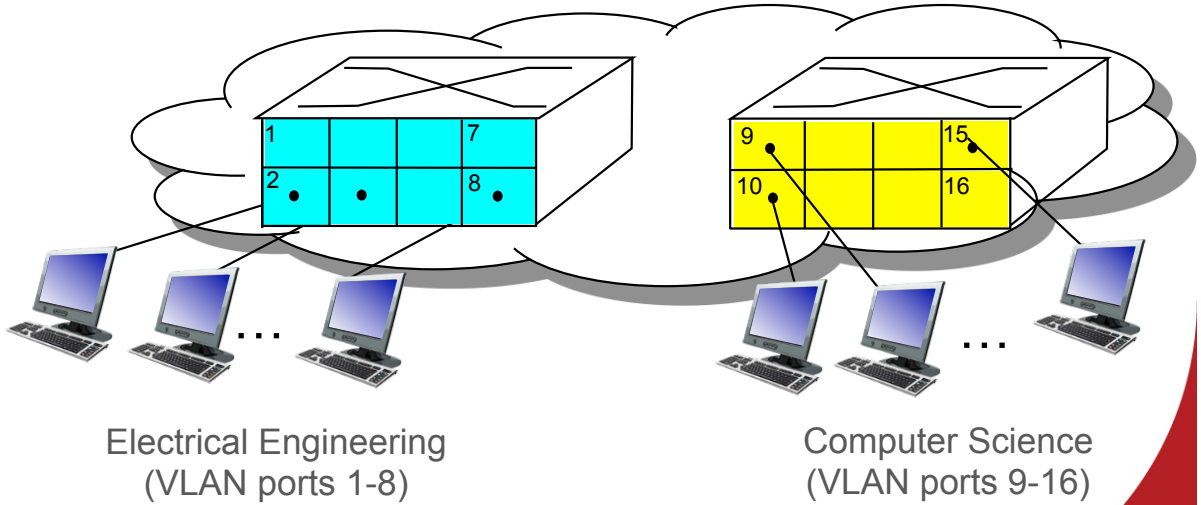
Port-based VLAN: switch ports grouped (by switch management software) so that *single* physical switch ……



Virtual Local Area Network

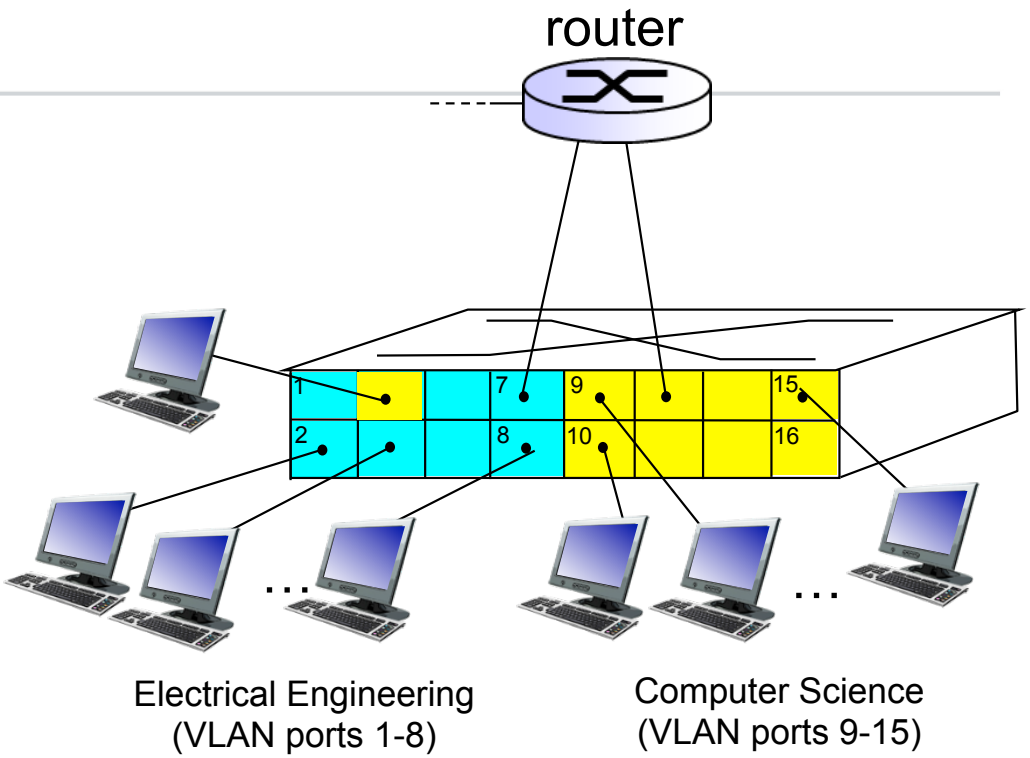
Switch(es) supporting VLAN capabilities can be configured to define multiple ***virtual*** LANs over single physical LAN infrastructure.

… operates as multiple virtual switches



Port-based VLAN

- *Traffic isolation:* frames to/ from ports 1-8 can *only* reach ports 1-8

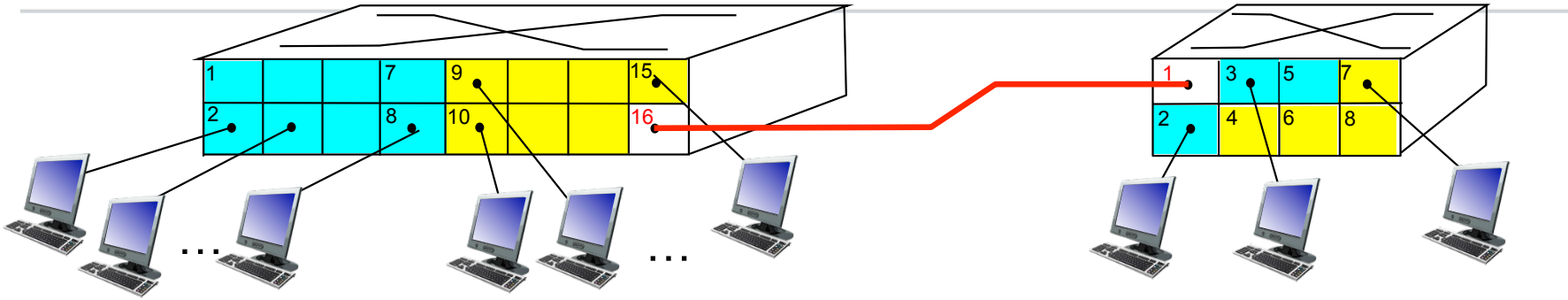


Dynamic membership: ports can be dynamically assigned among VLANs

Forwarding between VLANs: done via routing (just as with separate switches)

in practice vendors sell combined switches plus routers

VLANs Spanning Multiple Switches



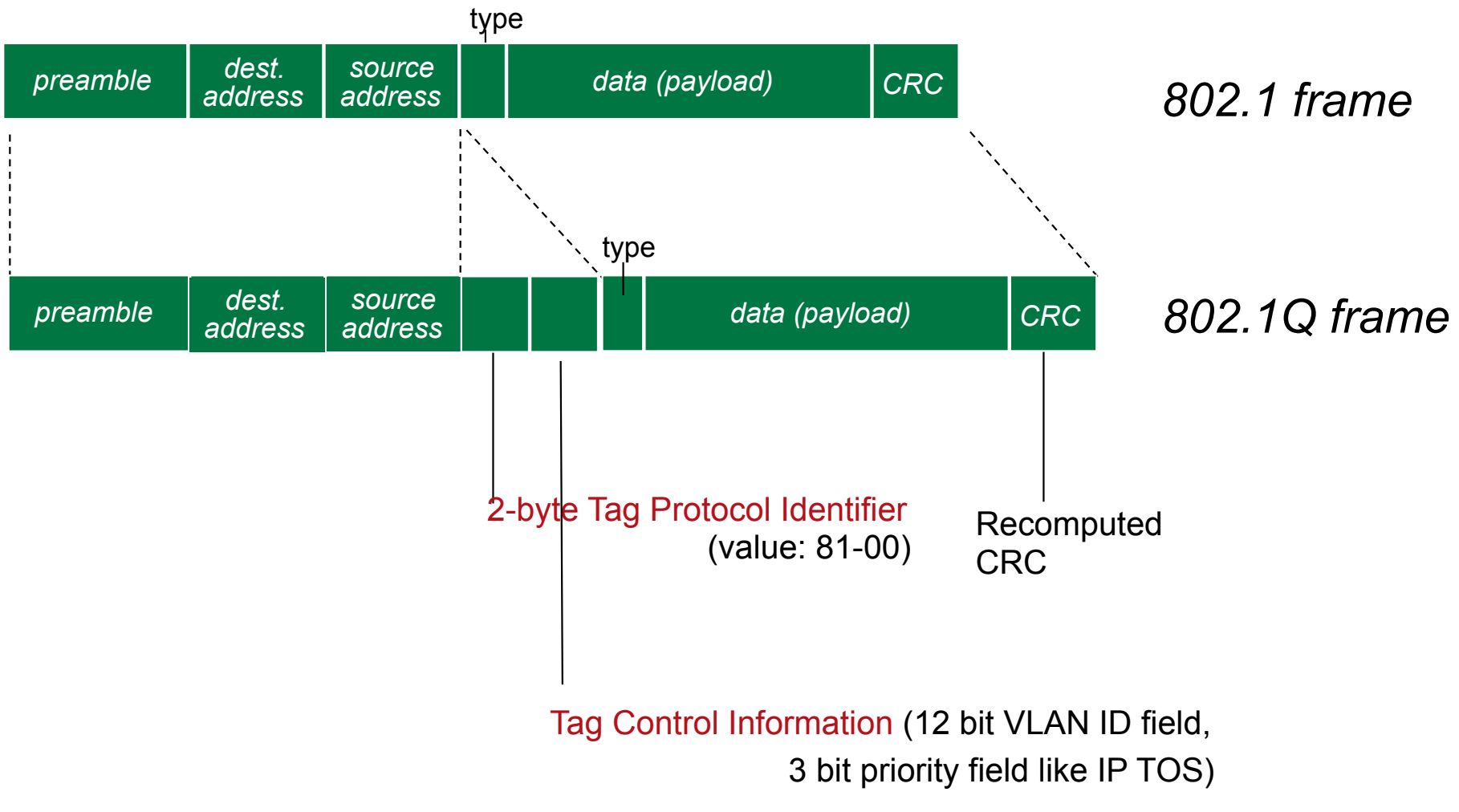
Electrical Engineering
(VLAN ports 1-8)

Computer Science
(VLAN ports 9-15)

Ports 2,3,5 belong to EE VLAN
Ports 4,6,7,8 belong to CS VLAN

- Trunk port: carries frames between VLANs defined over multiple physical switches
  - Frames forwarded within VLAN between switches can't be vanilla 802.1 frames (must carry VLAN ID info)
  - 802.1q protocol adds/removed additional header fields for frames forwarded between trunk ports

802.1Q VLAN frame format



Summary

Lancaster University

- Link Layer (cont.)
  - MAC Protocols
    - Channel partitioning, by time, frequency or code
      - Time Division, Frequency Division
    - Random access (dynamic),
      - ALOHA, S-ALOHA, CSMA, CSMA/CD
      - carrier sensing: easy in some technologies (wire), hard in others (wireless)
      - CSMA/CD used in Ethernet
    - Taking turns
      - polling from central site, token passing
      - bluetooth, FDDI, token ring
- Ethernet
  - Characteristics:
    - Topology, frame structure, service types
  - Switches
    - How does switched Ethernet work
    - Forwarding tables
- VLAN
  - Port-based VLAN
  - Multiple Switches
  - VLAN Frame Format

Questions?

Lancaster University

Questions!!

Lancaster University

- Multiple Choice
  - The forwarding tables in switched Ethernet learns addresses by:
    - A) polling all nodes B) computing routing tables C) recording addresses of incoming frames D) configuration
  - Which of the following statements are/is true?
    - I) There is a maximum length for Ethernet frames.
      II) In an Ethernet, channel utilization is higher when frames are shorter. III) There is a minimum length for Ethernet frames.
- What is the main difference between circuit switching and virtual circuit switching?
- For medium access control a Time Division Multiple Access (TDMA) or a Carrier Sense Multiple Access (CSMA) can be taken. Compare the two approaches considering pro & cons.

Answers

Lancaster University

- Multiple Choice
  - The forwarding tables in switched Ethernet learns addresses by:
    - A) polling all nodes B) computing routing tables C) recording addresses of incoming frames D) configuration
  - Which of the following statements are/is true?
    - I) There is a maximum length for Ethernet frames.
      II) In an Ethernet, channel utilization is higher when frames are shorter. III) There is a minimum length for Ethernet frames.
- What is the main difference between circuit switching and virtual circuit switching?
  - Virtual circuit switching is still a packet-switching technology. No physical connection is actually reserved in virtual circuit switching but only a logical path where the traffic has to pass through certain nodes. In circuit switching, a physical communication line is reserved for the whole connection duration like in classical telephony service.
- For medium access control a Time Division Multiple Access (TDMA) or a Carrier Sense Multiple Access (CSMA) can be taken. Compare the two approaches considering pro & cons.
  - TDMA: + high channel utilisation at high network traffic, +all stations have equal chance to send a packet at a given time frame; (no node starvation), – poor channel usage at low network traffic – synchronisation necessary, – central coordinator required
  - CSMA: +good channel utilisation under normal load, +not wastage of resources if node has not got anything to transmit , – high chance of collision at high network traffic, – node starvation might happen (chance that a node has not the chance to transmit a packet)

Internet Protocol (IP)

Dr Andrew Scott

a.scott@lancaster.ac.uk

Internet Protocol

- Connectionless **network protocol**
  - No attempt to build path prior to transmission
- Best-effort – *packets may be*:
  - Lost
  - Delivered out of order
  - Duplicated
  - Delayed

Key Internet Design Decisions

- Test before release
  - Require multiple working implementations
    c.f. other standards that release and fix
- Simplicity: Avoid unnecessary features
  - Only accept essentials
  - Avoid multiple ways of achieving same thing

Key Internet Design Decisions

- Modularity
  - e.g. protocol stack of independent layers
- Support Heterogeneity
  - Allow different network hardware, structures, and applications
  - Avoid fixed options and parameters
    - Have sender and receiver negotiate values

Key Internet Design Decisions

- Choose design over supporting everything
  - Don't allow one desired feature to break design
- Be strict when sending, tolerant when receiving
  - Rigorously stick to standard when sending
  - Expect errors in received messages
- Scalability
  - Assume no centralised system will handle global load
  - Spread load as evenly as possible
  - Design in performance and design out cost

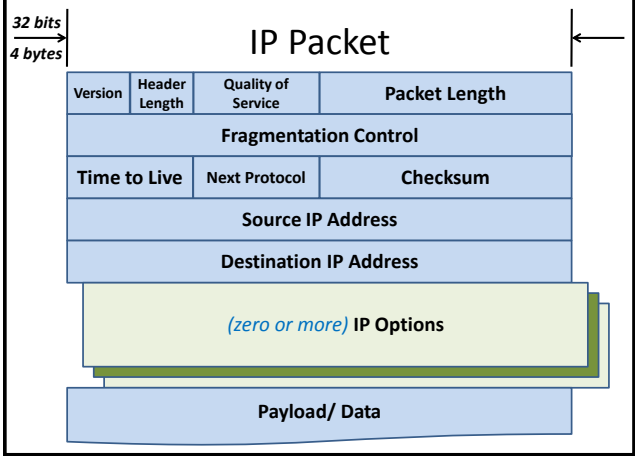
RFC 791 (updated by RFC2474, …)

INTERNET PROTOCOL VERSION 4

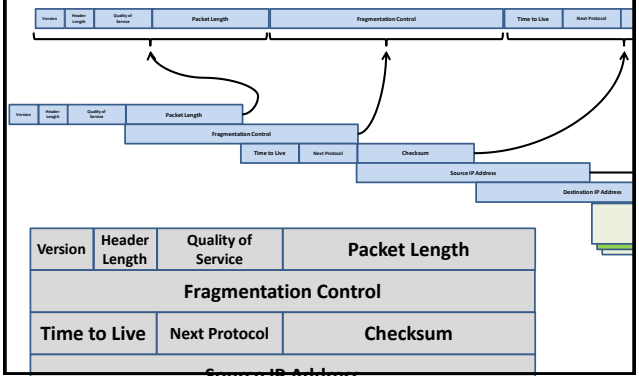
Internet Protocol version 4 (IPv4)

- Current generation of protocol
- Being phased out in favour of IPv6
  - *More on this later…*

IP Packet



Packet Transmission



Implementation

```
struct iphdr {
   /*
    *   x86 GCC packing from 0 up so packed fields seem reversed
    */
   uint8_t   hdrlen:4;       //  In DWORDS
   uint8_t   version:4;
   uint8_t   ecn:2;          //  Explicit Congestion Notification
   uint8_t   dscp:6;         //  DiffServ Code Point
   uint16_t  length;
   uint16_t  ident;
   uint16_t  fragoff:13;
   uint16_t  flags:3;
   uint8_t   ttl;
   uint8_t   protocol;
   uint16_t  checksum;
   uint32_t  srcip;
   uint32_t  dstip;
   uint32_t  options[ ];     //  Only present if hdrlen > 5
} __attribute__((__packed__));
```

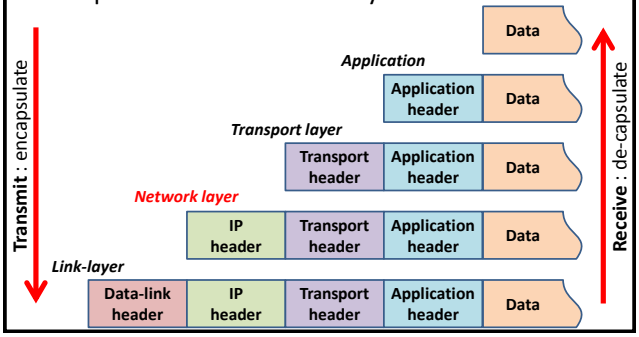
Alternative to avoid use of C bit fields:

```
uint8_t  x = version << 4 | hdrlen;

version = x >> 4;
hdrlen  = x & 0x0f;
```

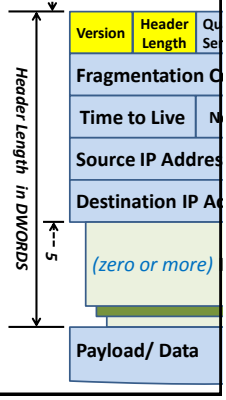
Remember Encapsulation

- IP packets sit within link-layer frames

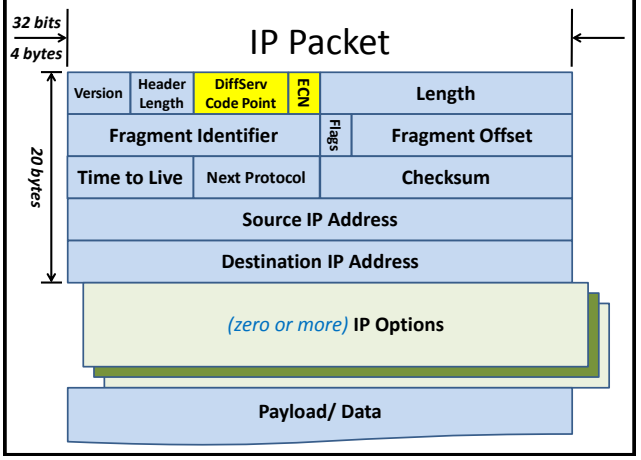


IP Packet

- Version
  - Either 4 or 6
  - Unknown values rejected
- Header Length
  - *n* x 4 bytes
  - More than 5 implies options
    - i.e. **basic header 20 bytes**



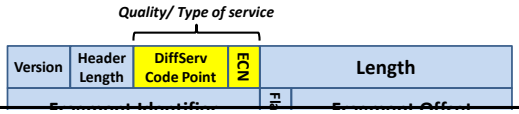
IP Packet



Quality of Service : *DS field*

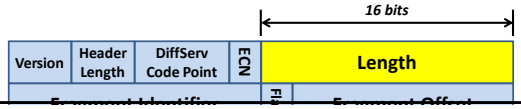
- Differentiated Services (DiffServ) Code Point
  - Identifies priority of traffic -- *Voice, file transfer, …*
    - *Note: Use of this field has changed over the years*
- Explicit Congestion Notification (ECN)
  - Flags whether packet has experienced congestion

***Note**: Quality of Service covered in Advanced Networking module*



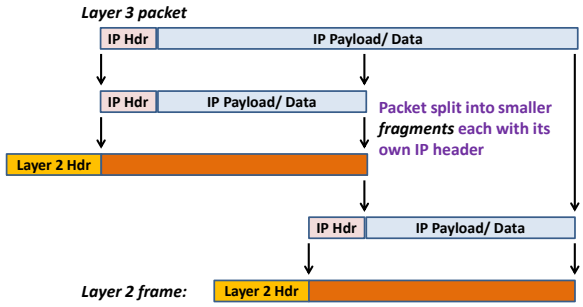
Packet Length

- Length of whole IP packet in bytes
  - Includes IP header and payload
    - Excludes surrounding layer 2 headers/ encapsulation
- Max length of IP packet therefore 65535 bytes
  - Normally limited to 1500 bytes by Ethernet



Fragmentation

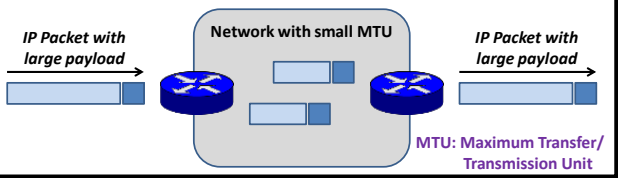
- Packet may be too large for underlying network



Transparent Fragmentation

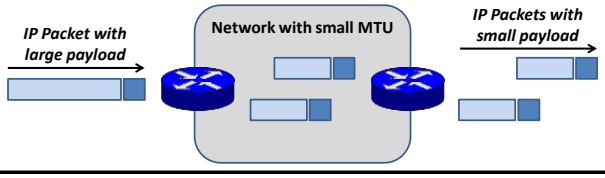
Good/ Bad

- ✔ Following devices on path unaware of fragmentation
- ✘ Fragments must exit network through same router
- ✔ Overhead of extra packet headers removed at exit
- ✘ Ties up resources until last fragment of packet arrives
  - *Better left to destination as must handle this anyway?*



Non-Transparent Fragmentation

- **Standard for IP traffic**
- Simplifies task of routers, penalty being …
  - All subsequent devices have to carry and process ***additional IP packet headers*** (more overhead)



Lost Fragments

- If a fragment is lost

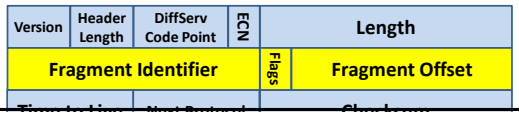
  whole packet is retransmitted

- Host/ application should therefore avoid sending large packets likely to be fragmented
  - Send smaller packets -- *choose small value, or…*
  - Find ***Maximum Transfer Unit*** (MTU) for path between source and destination

We'll look at how to do this later

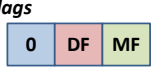
Fragmentation

- Fragment Identifier (ID)
  - Sending host uses incrementing ID in each packet
  - If fragmented, this value copied to each fragment
- At receiver
  - Packets with same ID belong to same source packet

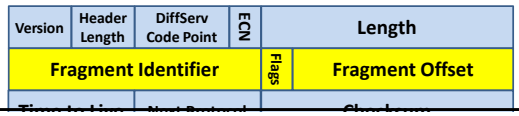


Fragmentation

- Flags

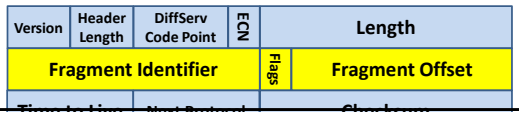


  - Don't Fragment (DF)
    - Tells devices not to fragment packet
    - Device must reply with error if packet too big
  - More Fragments (MF)
    - Set to one for all but last fragment; 0 if only packet

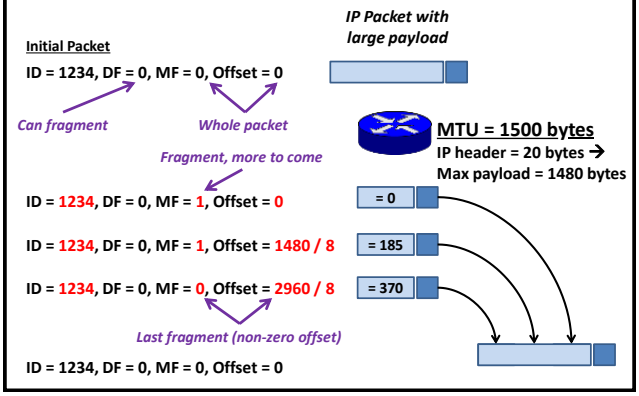


Fragmentation

- Fragment Offset
  - Tells receiver where to place fragment in reconstituted packet
  - Offset in 8 byte chunks
    - All but last fragment must be a multiple of 8 bytes

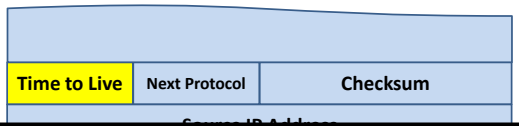


Fragmentation

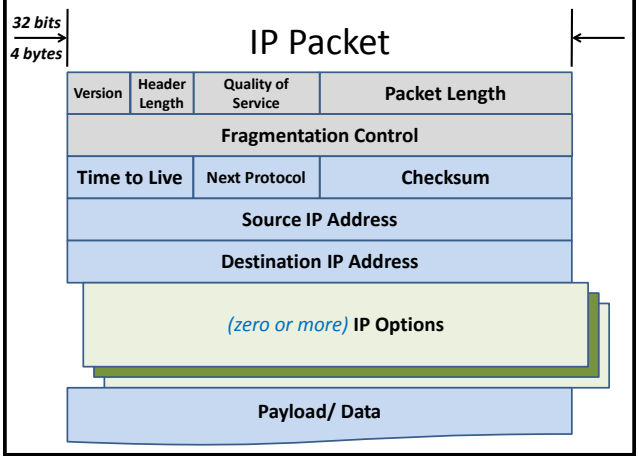


Time To Live (TTL)

- Routers decrement TTL by
  - Number of seconds spent processing packet
  - At least 1
- Any packet with a TTL of 0 is discarded
  - Device typically replies with *TTL_EXCEEDED* error

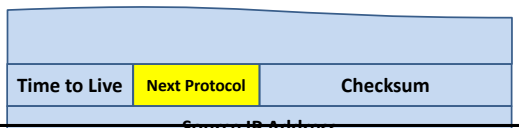


IP Packet



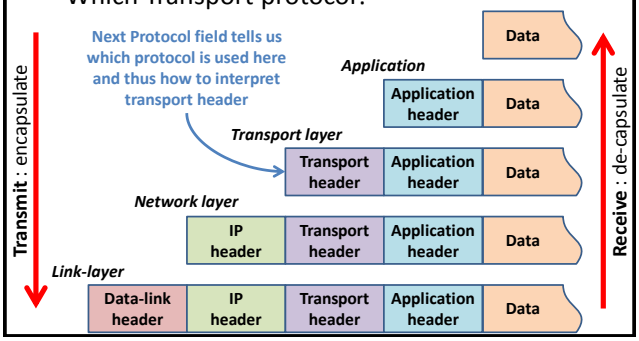
Protocol

- Indicates
  - Protocol used to encapsulate following user data
  - Determines how payload will be processed
  - Example: 17 for UDP

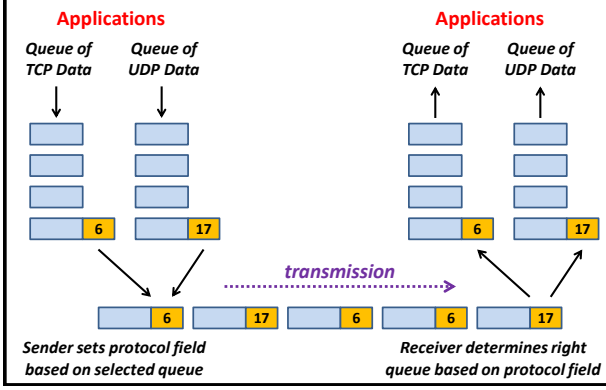


Encapsulation

- Which Transport protocol?



De-multiplexing using Protocol



Key Protocol Numbers

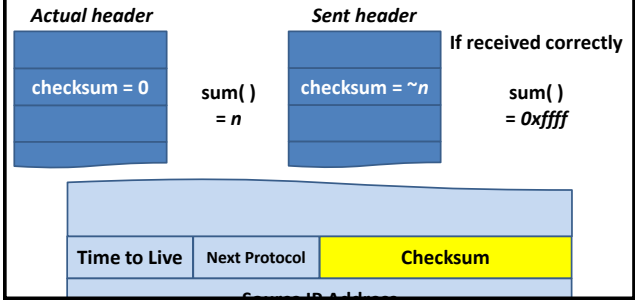
Number	Protocol	Description
1	ICMP	Internet Control Message Protocol
2	IGMP	Internet Group Management Protocol
6	TCP	Transmission Control Protocol
17	UDP	User Datagram Protocol

TCP is a Virtual Circuit/ stream based protocol

UDP is a Connection-Less, Datagram protocol

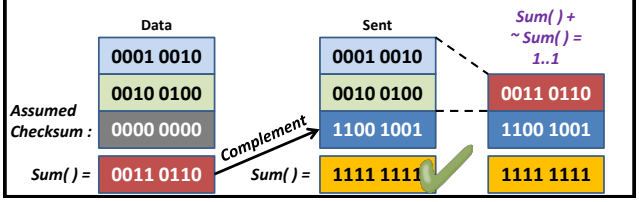
Checksum

- One's complement (~) of 16 bit sum of words



How is Checksum Working?

- We calculate sum and send its complement
- When we calculate sum at receiver
  - We're getting orig. sum + complement of sum
  - If we don't get 0xffff we must have an error

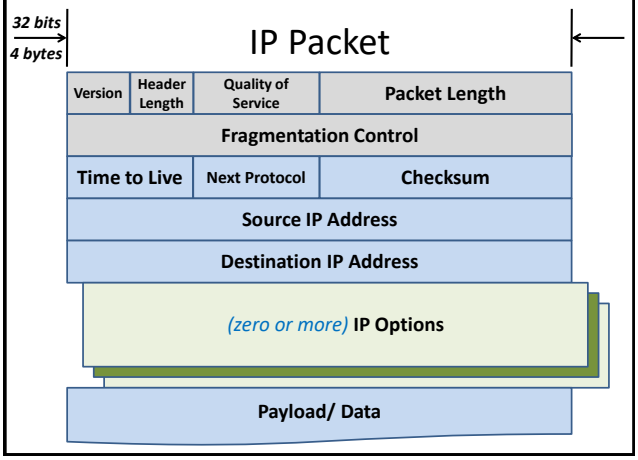


Internet Protocol (IP) contd.

Dr Andrew Scott

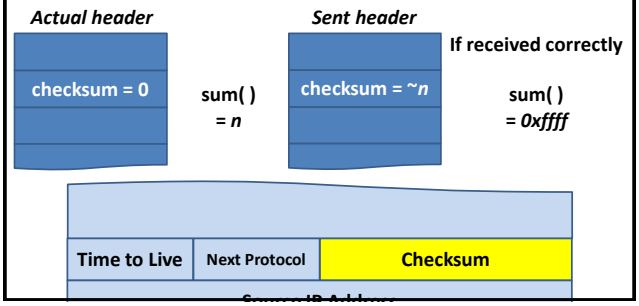
a.scott@lancaster.ac.uk

IP Packet



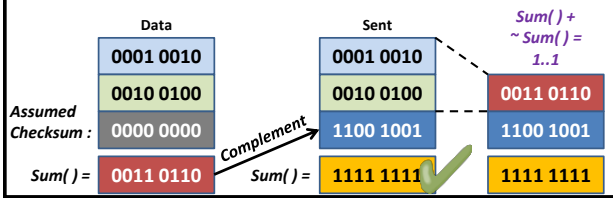
Checksum

- One's complement (~) of 16 bit sum of words



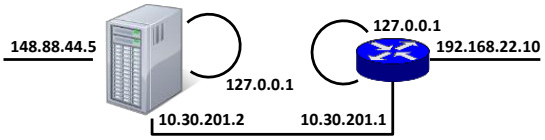
How is Checksum Working?

- We calculate sum and send its complement
- When we calculate sum at receiver
  - We're getting orig. sum + complement of sum
  - If we don't get 0xffff we must have an error



IPv4 Addresses

- 32 bit/ 4 byte identifier
  - Identify network interfaces ***not*** *hosts/ devices* *
  - A device will have multiple addresses
    - At least one per virtual or physical network interface
      - Notice all devices use same loopback address: 127.0.0.1



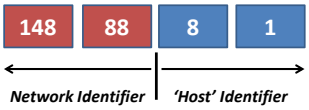
** Physical interfaces can have multiple IP addresses but not generally useful*

Checksums on Loopback Interface

- Loopback interface (IP address 127.0.0.1) used for traffic destined for same host as sender
- Some systems (including Linux) assume errors highly unlikely as packet will stay within machine
  - Don't set or check checksum on loopback interface
    - Generally considered waste of CPU cycles
  - May see 'bad checksum' errors on 127.0.0.1
    - Can confirm with *tcpdump –i lo -vv*

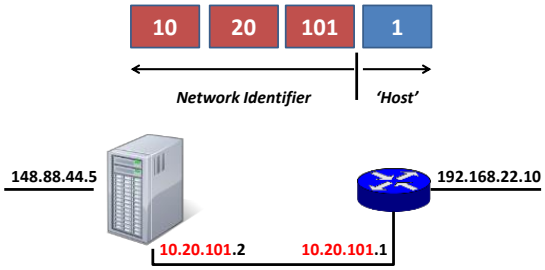
IPv4 Addresses

- Typically written as 4 bytes, e.g., 148.88.8.1
  - Often treated as a 32 bit *long*, i.e., 0x94580801
- Split into network and host parts
  - Split can be made at different positions



IPv4 Addresses

- Interfaces on same link share network part



Private Addresses

- Available for (organisation) internal/ test networks
- **Must not be made externally visible**

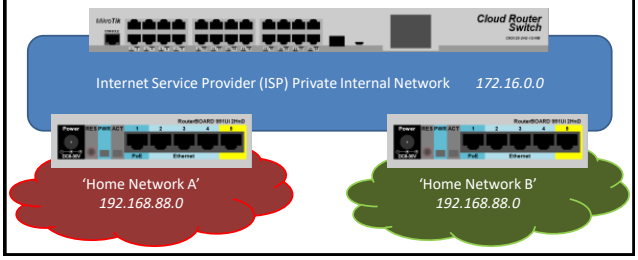
10.0.0.0 -- 10.255.255.255
172.16.0.0 -- 172.31.255.255
192.168.0.0 -- 192.168.255.255

Free for use, but check your organisation isn't already using part of range, for example, University makes extensive use of 10.xxx addresses

RFC1918

Private Addresses

- Allow autonomy from any numbering authority
  - Note that private addresses are not globally unique
    - Therefore 'meaningless' outside of organisation



Special Addresses

- Multicast 224.0.0.0 – 239.255.255.255
  - Packets delivered to a group of 'interested' devices
  - Hosts subscribe to group in order to receive packets
    - Can be useful for media or data distribution services

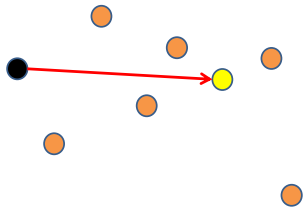
RFC5771

- Broadcast 255.255.255.255
  - For delivery to all devices on local network

RFC919

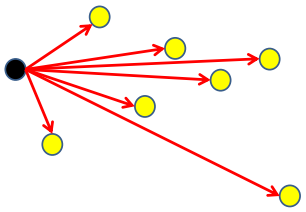
Addressing: *Unicast (the normal case)*

- Deliver to specific addressed host



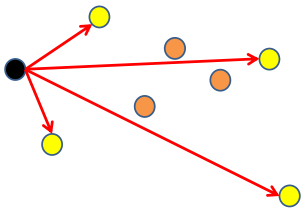
Addressing: *Broadcast*

- Deliver to all hosts on network



Addressing: *Multicast*

- Deliver to defined group of hosts
  - On one or more networks

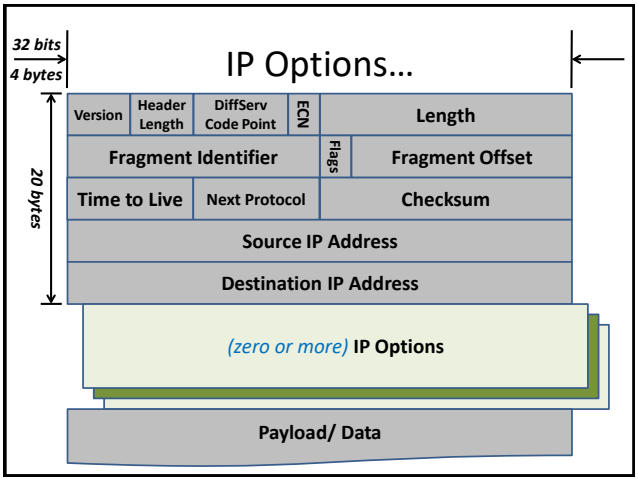


Special Addresses

0 … 0		Current host
0 … 0	*host*	Some host on current network
network	1 … 1	All hosts on specified network
1 … 1		All hosts on current link

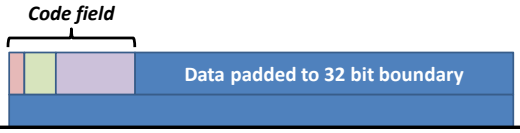
Note: a network may be formed from more than one link

IP Options…



IPv4 Options

- Variable length field
  - Often not present at all
  - Padded to 32 bit boundary by PAD field
- All options must start with a code field
  - May be the only field in option



IPv4 Option Code Field

- Copy (1 bit)
  - 0 → copy only to first fragment
  - 1 → copy to all fragments
- Option Class (2 bits)
  - 0 → Datagram or network control
  - 1 → Reserved
  - 2 → Debugging and measurement
  - 3 → Reserved
- Option Number (5 bits)

Questions…

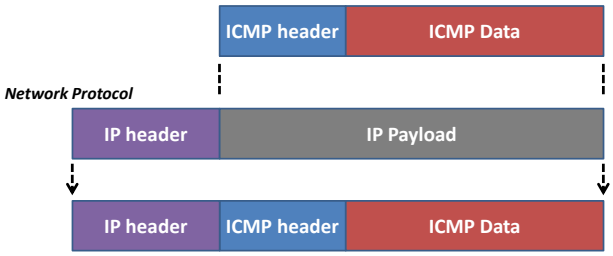
- How do we find the Maximum Transfer Unit?
- What happens when things go wrong?
- How do we control or test IP based systems?

RFC 792

INTERNET CONTROL MESSAGE PROTOCOL (ICMP)

Internet Control Message Protocol

- Encapsulated in IP packet
  - Note ICMP not considered a transport protocol



Internet Control Message Protocol

- Mechanism by which
  - Routers send error messages to source devices
  - One IP instance communicates important state to another
- When used for errors
  - **Only used to report errors** back to source
  - Source must **use other means to correct error**
  - Typically, errors only sent if error caused packet discard

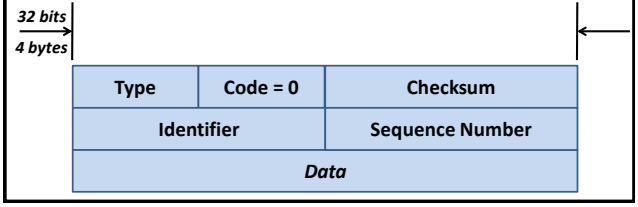
Main ICMP Message Types

ICMP Type	Description
0	Echo reply -- response to 'ping'
3	Destination unreachable
4	Source quench *(sending too fast, slow down)*
5	Redirect *(change route)*
8	Echo request -- 'ping'
9	Router advertisement
10	Router discovery
11	Time exceeded *(Code = 0 due to TTL, =1 due to frag. Reassembly time)*
12	Bad IP header/ parameter problem
13	Timestamp Request
14	Timestamp Reply
17	Address Mask Request
18	Address Mask Reply

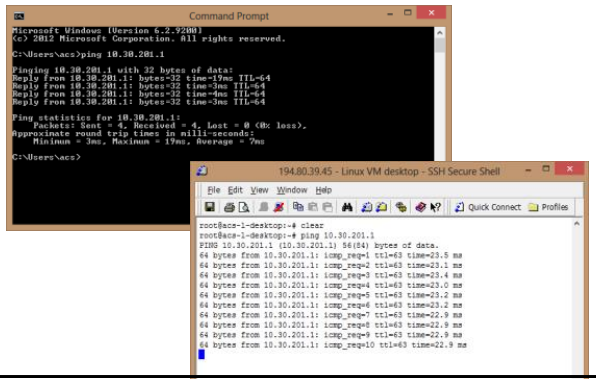
Green entries most likely to be seen

ICMP Echo and Echo-Reply

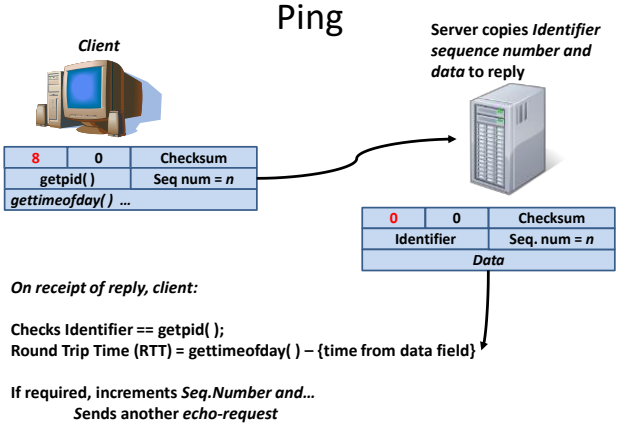
- Type = 8 Echo-request
- Type = 0 Echo-reply
  - Receiver copies Identifier, Seq. number, and data into reply for matching by source



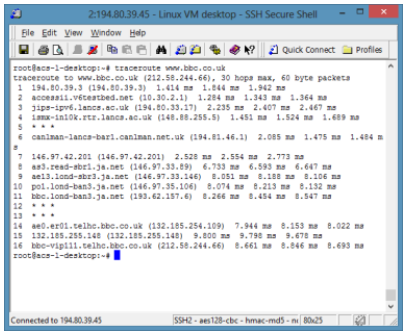
Finding Round Trip Time: *ping*



Ping



Finding a Path: *traceroute*

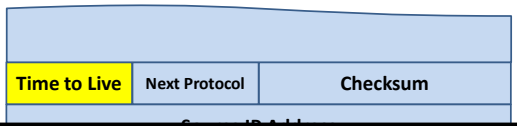


Finding a Path: *traceroute*

- Note: Important that *Don't Fragment* flag set
  - Need to find single path to destination
    - Possible for fragments to take different paths

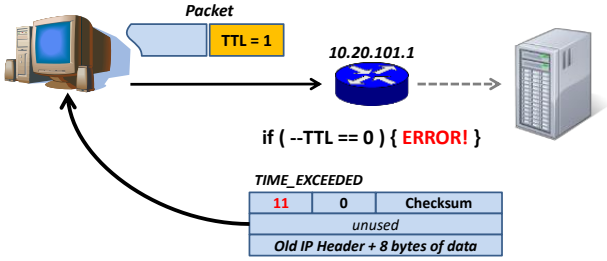
Remember: *Time To Live (TTL)*

- Routers decrement TTL by
  - Number of seconds spent processing packet
  - At least 1
- Any packet with a TTL of 0 is discarded
  - Device typically replies with *TIME_EXCEEDED* error

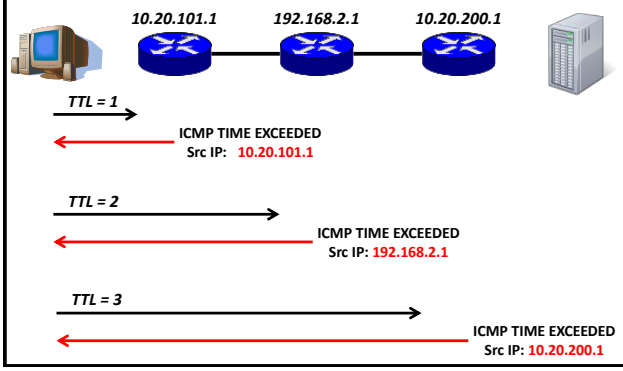


Finding a Path: *traceroute*

- ICMP Error message sent when TTL = 0
  - Sourced from IP address of router that found error



Finding a Path: *traceroute*

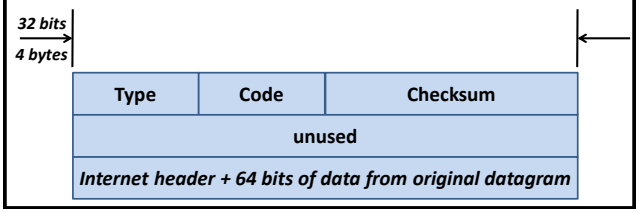


Finding Maximum Transfer Unit

- Like to know MTU of path between A and B
- Repeat with reducing packet sizes/ binary chop:
  - Send large packet from A to B with DF flag set
    - Expect ICMP Destination Unreachable [Type 3, Code 4]
      - Packet exceeds MTU and *Don't Fragment* (DF) set
  - Until success or B sends port unreachable error
- Better:
  - Select packet sizes based on likely network types compensating for size of IP header etc.

ICMP Type 3: Destination Unreachable

ICMP Type	Code	Description
3	**0**	Destination **network** unreachable *(no route to network)*
3	**1**	... **host** unreachable *(typ. unknown link-layer addr.)*
3	**2**	... **protocol** unreachable *(sent by destination host)*
3	**3**	... **port** unreachable *(sent by destination host)*
3	**4**	Packet exceeds MTU and *Don't Fragment* (DF) set
3	**5**	Source route failed



LINKING NETWORK AND DATA-LINK LAYERS…

Addressing in the Network

- Network Layer
  - Addresses network end-systems
    - Source Address : Original sender of packet
    - Destination Addr.: Final destination of packet
  - Beware: Network layer knows routers * exist
    - It just treats them as other end systems
- Data-Link layer
  - Addresses end-points on same physical link *…hop-by-hop*
    - Source Address: Sender on current physical link
    - Destination Addr.: Next hop's interface on current link

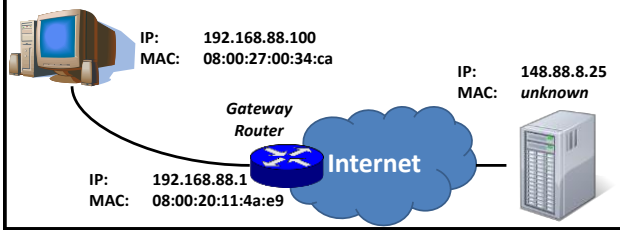
*** Routers forward packets between network links and thus around network**

Need Address Mapping Functions

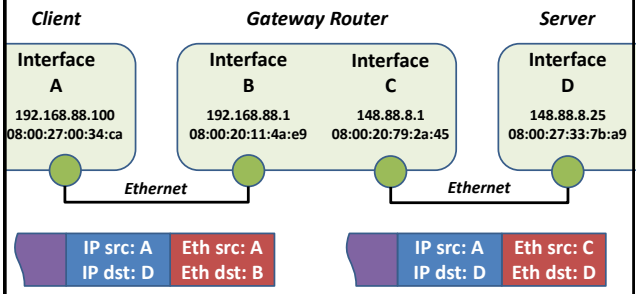
- Network layer deals with network addresses
  - E.g. 192.168.88.2 *hierarchical*
- Data-link layer deals with hardware addresses
  - Medium Access Control (MAC) addresses
  - E.g. 00:80:37:00:20:5a *flat*

Sending a Network (IP) Packet

- We must tell our link-layer to send packet to hardware (MAC) address of gateway router
  - Packet is addressed to IP address of server
    …and sent by link-layer to MAC address of gateway router



Looking Hop-by-Hop

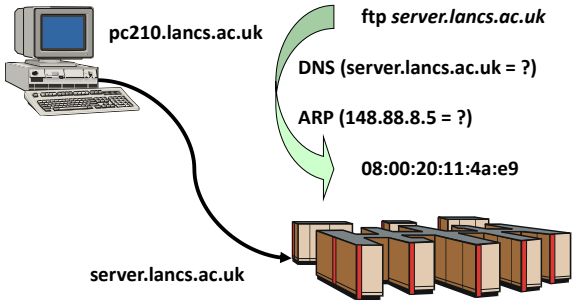


Mapping Addresses

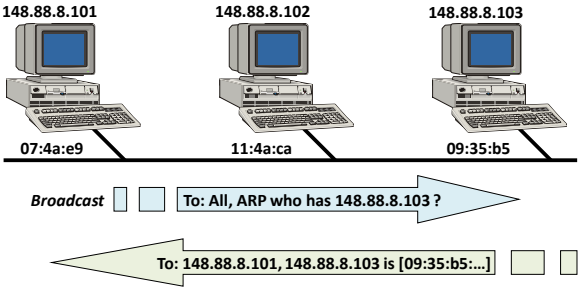
- Names ⇔ IP Addresses
  - Domain Name Service (DNS)
- IP Addresses ⇒ Hardware/ Link-Layer Addr
  - Address Resolution Protocol (ARP)
- Link-Layer Address ⇒ IP Address
  - Reverse Address Resolution Protocol (RARP)

Hardware addresses more commonly known as Medium Access Control (MAC) addresses
See the IEEE LAN/RM in first lecture to see why

Names and Addresses

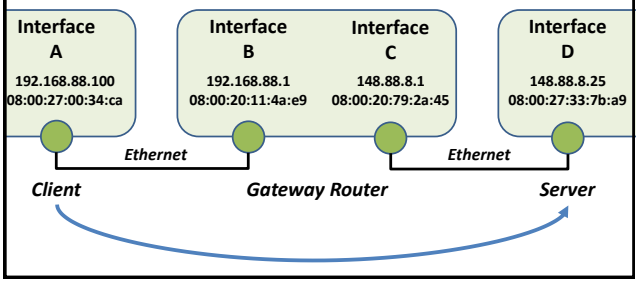


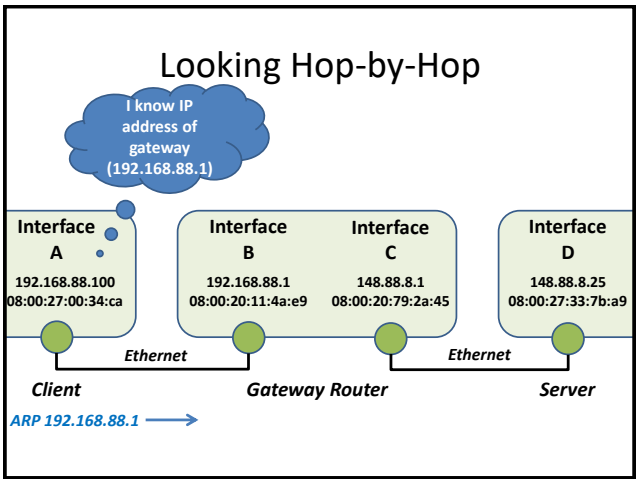
Address Resolution Protocol (ARP)

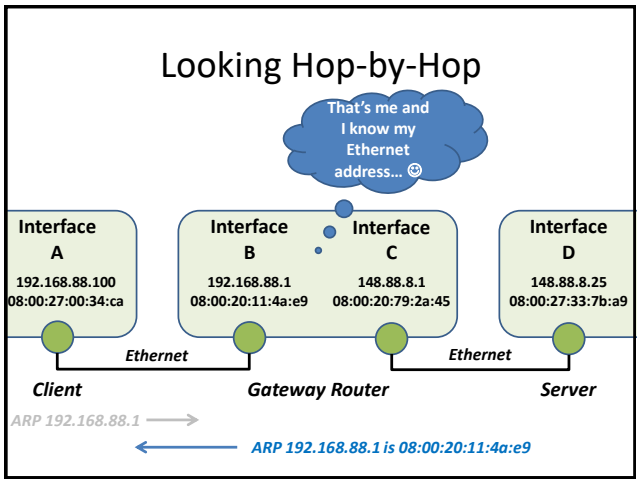


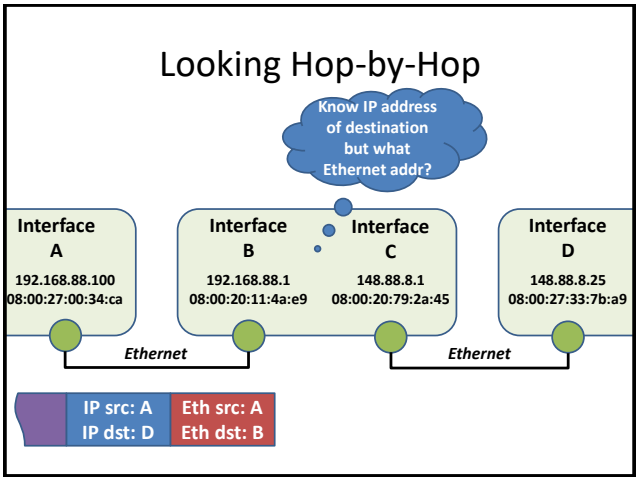
Looking Hop-by-Hop

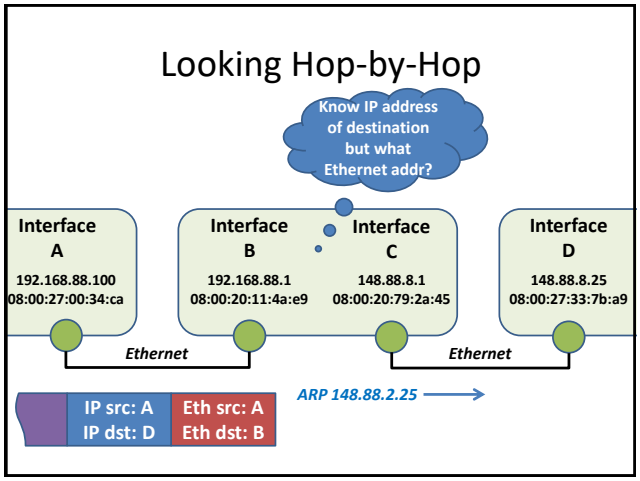
- Consider the following network
  - Client sending packet to server…

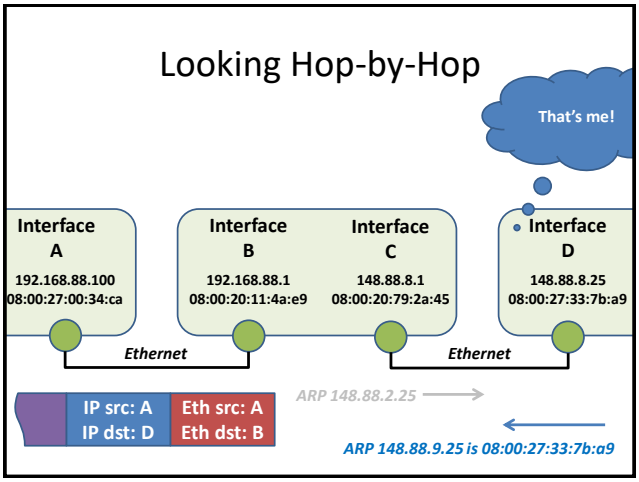


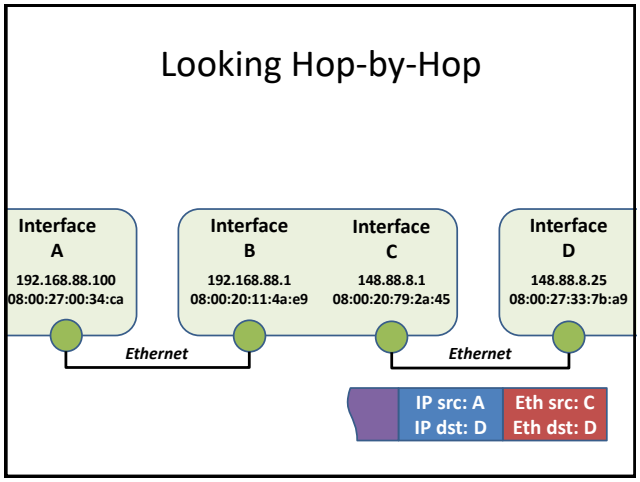

Looking Hop-by-Hop
I know IP address of gateway (192.168.88.1)
Interface A
192.168.88.100
08:00:27:00:34:ca
Interface B
192.168.88.1
08:00:20:11:4a:e9
Interface C
148.88.8.1
08:00:20:79:2a:45
Interface D
148.88.8.25
08:00:27:33:7b:a9
Ethernet
Ethernet
Client
Gateway Router
Server
ARP 192.168.88.1


Looking Hop-by-Hop
That's me and I know my Ethernet address...
Interface A
192.168.88.100
08:00:27:00:34:ca
Interface B
192.168.88.1
08:00:20:11:4a:e9
Interface C
148.88.8.1
08:00:20:79:2a:45
Interface D
148.88.8.25
08:00:27:33:7b:a9
Ethernet
Ethernet
Client
Gateway Router
Server
ARP 192.168.88.1
ARP 192.168.88.1 is 08:00:20:11:4a:e9


Looking Hop-by-Hop
Know IP address of destination but what Ethernet addr?
Interface A
192.168.88.100
08:00:27:00:34:ca
Interface B
192.168.88.1
08:00:20:11:4a:e9
Interface C
148.88.8.1
08:00:20:79:2a:45
Interface D
148.88.8.25
08:00:27:33:7b:a9
Ethernet
Ethernet
IP src: A
IP dst: D
Eth src: A
Eth dst: B

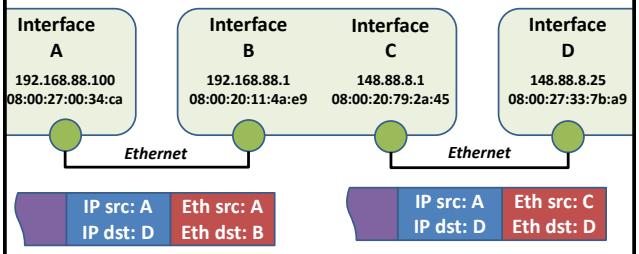

Looking Hop-by-Hop
Know IP address of destination but what Ethernet addr?
Interface A
192.168.88.100
08:00:27:00:34:ca
Interface B
192.168.88.1
08:00:20:11:4a:e9
Interface C
148.88.8.1
08:00:20:79:2a:45
Interface D
148.88.8.25
08:00:27:33:7b:a9
Ethernet
Ethernet
IP src: A
IP dst: D
Eth src: A
Eth dst: B
ARP 148.88.2.25


Looking Hop-by-Hop
That's me!
Interface A
192.168.88.100
08:00:27:00:34:ca
Interface B
192.168.88.1
08:00:20:11:4a:e9
Interface C
148.88.8.1
08:00:20:79:2a:45
Interface D
148.88.8.25
08:00:27:33:7b:a9
Ethernet
Ethernet
IP src: A
IP dst: D
Eth src: A
Eth dst: B
ARP 148.88.2.25
ARP 148.88.9.25 is 08:00:27:33:7b:a9


Looking Hop-by-Hop
Interface A
192.168.88.100
08:00:27:00:34:ca
Interface B
192.168.88.1
08:00:20:11:4a:e9
Interface C
148.88.8.1
08:00:20:79:2a:45
Interface D
148.88.8.25
08:00:27:33:7b:a9
Ethernet
Ethernet
IP src: A
IP dst: D
Eth src: C
Eth dst: D

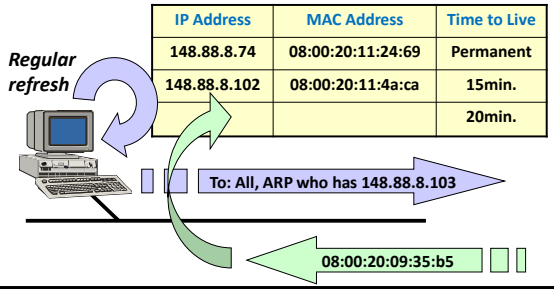
Notice Addresses

- IP addresses stay same end-to-end
- Ethernet/ Link-layer addresses change hop-by-hop



ARP Table/ Cache

- Allows hosts to remember recently used mappings
  - Time to live allows for machines changing address



Example : Windows

```
C:\> arp -a

Interface: 10.20.101.72 --- 0xc
  Internet Address      Physical Address      Type
  10.20.101.1           00-0c-42-f8-b2-e2     dynamic
  10.20.101.127         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Linux

```
acs:~$ arp -a
? (194.80.37.1) at 00:07:b4:00:25:02 [ether] on eth0
```

Example: Cisco IOS

```
wallace# show ip arp
Protocol  Address     Age(min)  Hardware Addr   Type   Interface
Internet  10.20.0.1          -  0007.b404.7601  ARPA   GigE0/1.70
Internet  10.20.0.4        123  000c.299f.9e80  ARPA   GigE0/1.70
Internet  10.20.0.5         70  0004.2353.2b46  ARPA   GigE0/1.70
Internet  10.20.0.6        128  0015.17d1.541f  ARPA   GigE0/1.70
Internet  10.20.0.10         2  000c.29a8.396e  ARPA   GigE0/1.70
Internet  10.20.0.12         0  0015.1776.63cf  ARPA   GigE0/1.70
Internet  10.20.0.15         6  000c.29c2.75b9  ARPA   GigE0/1.70
```

Example: MikroTik

```
[admin@MikroTik] > /ip arp  print
Flags: X - disabled, I - invalid, H - DHCP, D - dynamic, P - published
 #    ADDRESS         MAC-ADDRESS         INTERFACE
 0 D 10.10.21.31     00:07:B4:04:76:01   bridge-local
 1 D 10.10.21.18     00:0C:29:9F:9E:80   wlan1
 2 D 10.10.21.22     00:04:23:53:2B:46   wlan1
 3 D 10.10.21.25     00:15:17:D1:54:1F   wlan1
[admin@MikroTik] >
```

Gratuitous ARP

- Typically sent on address configuration
  - Source and Destination IP both set to sender's address
  - Sent to Ethernet/ MAC broadcast address ff:ff:ff:ff:ff:ff
  - **i.e. an ARP request to itself**
- Forces other nodes to update their ARP caches
  - Hosts will avoid sending data to old address
  - Network switches will get 'new' location of device
- Leads to Duplicate Address Detection
  - Any machine with same address will spot conflict

Problems with ARP

- Fundamental to communications system
  - Possible/ too easy to spoof or fake ARP replies
    - Thus *poison* ARP caches to redirect traffic

Lancaster University

SCC203 Computer Networks: MPLS & Addressing

Dr Andreas Mauthe (a.mauthe@lancaster.ac.uk)
Dr Andrew Scott (a.scott@lancaster.ac.uk)

Contents

Lancaster University

- Outline
  - Multi Protocol Label Switching
    - MPLS routing
    - MPLS vs. IP paths
  - Dynamic Host Configuration Protocol
    - How DHCP works
    - DHCP message formats
    - DHCP message Types
  - Network Address Translation
    - How NAT works
    - NAT Example
- Objectives
  - To know and understand about MPLS works
    - You should know how label switching works and what technique MPLS employs to forward packets
  - To understand how DHCP and NAT works
    - You should be familiar with the ideas principles underlying DCHP and NAT, be able to explain them and apply your knowledge in the lab.

Lancaster University

Reminder

Questions from the last lecture

Questions!!

Lancaster University

- Multiple Choice
  - The forwarding tables in switched Ethernet learns addresses by:
    - A) polling all nodes B) computing routing tables C) recording addresses of incoming frames D) configuration
  - Which of the following statements are/is true?
    - I) There is a maximum length for Ethernet frames.
      II) In an Ethernet, channel utilization is higher when frames are shorter. III) There is a minimum length for Ethernet frames.
- What is the main difference between circuit switching and virtual circuit switching?
- For medium access control a Time Division Multiple Access (TDMA) or a Carrier Sense Multiple Access (CSMA) can be taken. Compare the two approaches considering pro & cons.

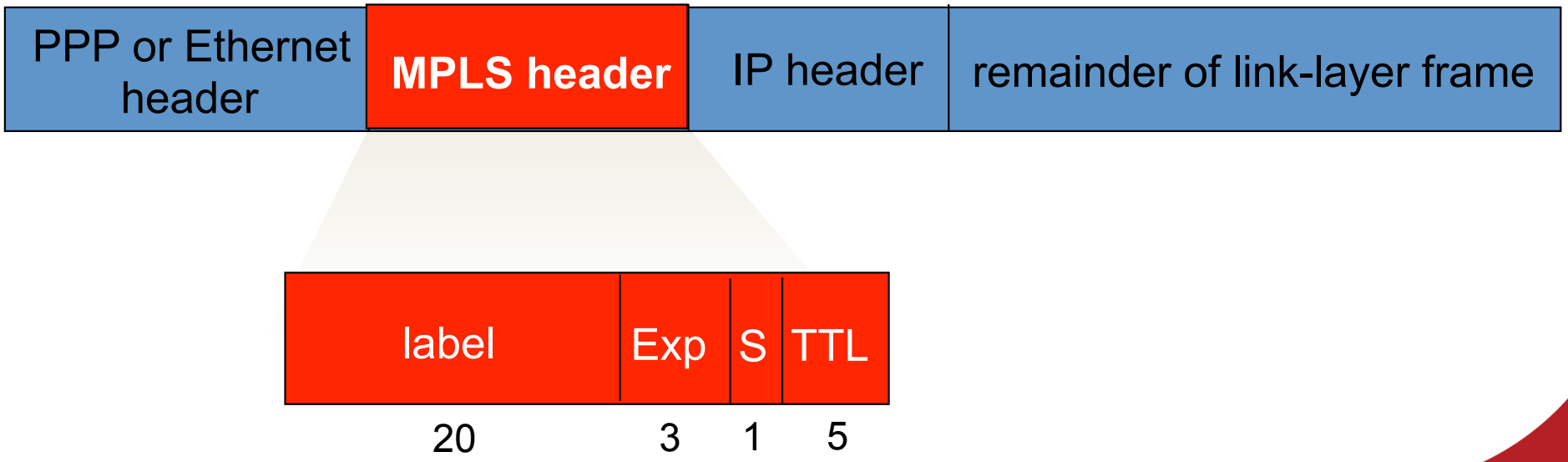
Lancaster University

LINK Virtualisation: MPLS

From Virtual Link to Virtual Networks

Multiprotocol label switching (MPLS)

- Initial goal:
  - High-speed IP forwarding using fixed length label (instead of IP address)
    - Fast lookup using fixed length identifier (rather than shortest prefix matching)
    - Borrowing ideas from Virtual Circuit (VC) approach
    - But IP datagram still keeps IP address!



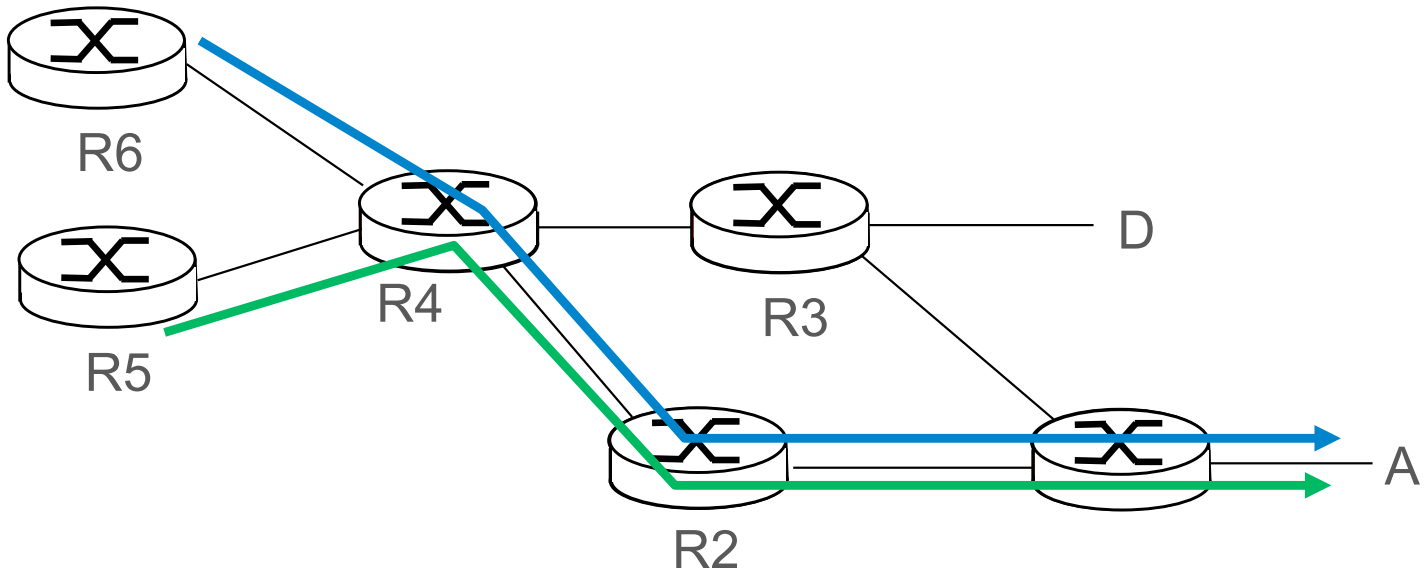
MPLS Capable Routers

Lancaster University

or *label-switched router*

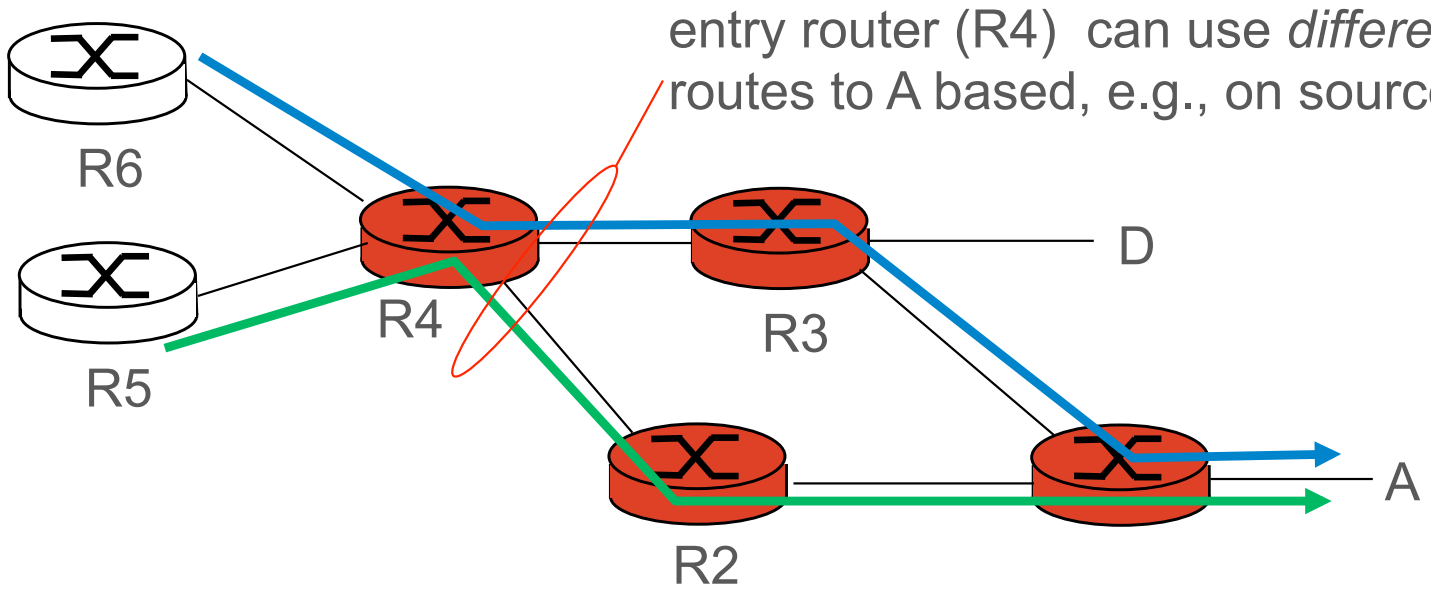
- Basic Idea
  - Forward packets to outgoing interface based only on label value (don't inspect IP address)
    - MPLS forwarding table distinct from IP forwarding tables
- Advantage
  - Flexibility: MPLS forwarding decisions can differ from those of IP
    - Use destination and source addresses to route flows to same destination differently (traffic engineering)
    - Re-route flows quickly if link fails: pre-computed backup paths (useful for VoIP)

MPLS versus IP paths



IP routing: path to destination determined by destination address alone

MPLS versus IP paths

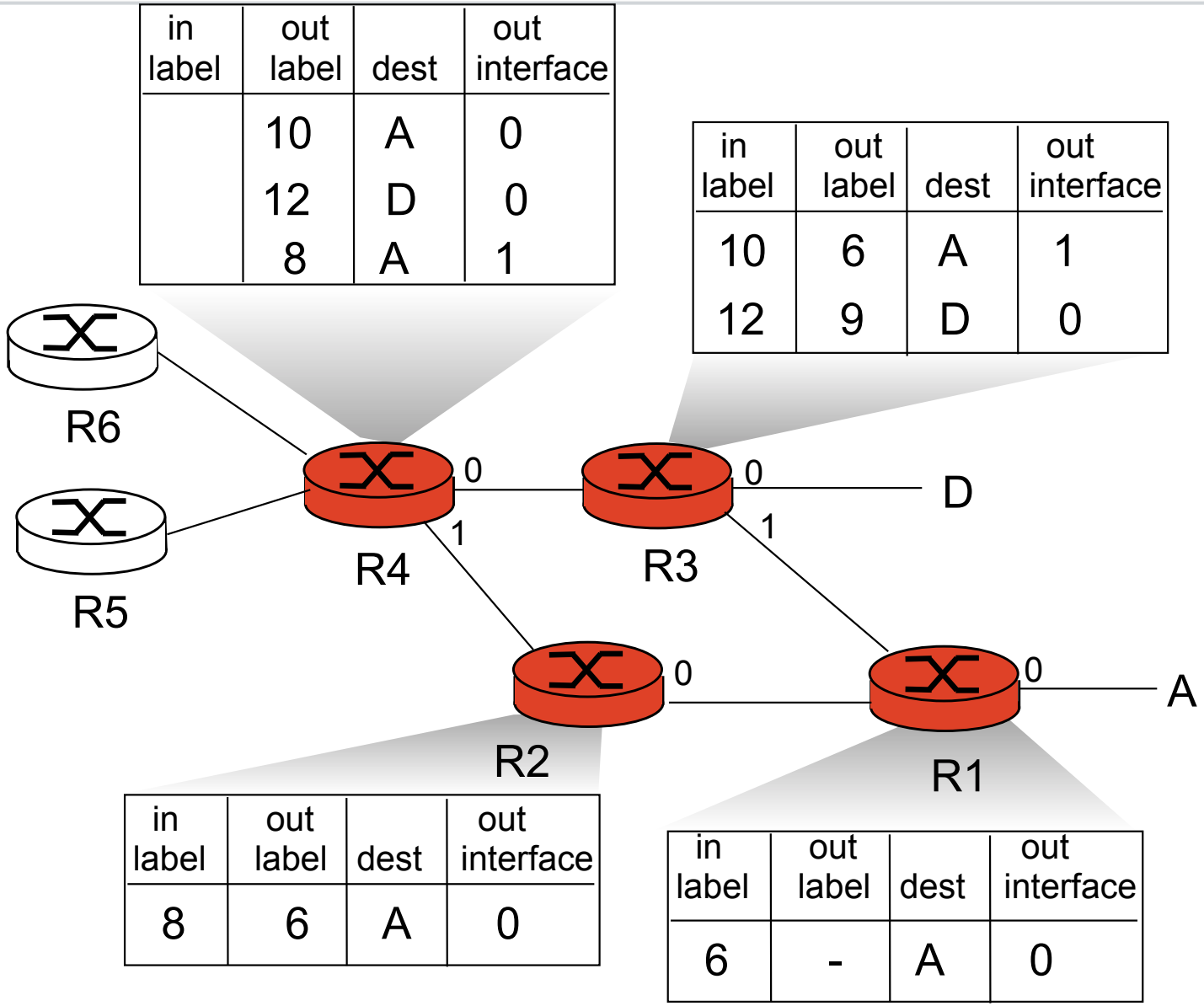


- IP routing: path to destination determined by destination address alone
- MPLS routing: path to destination can be based on source and dest. address
  - Fast reroute: pre-compute backup routes in case of link failure

IP-only router

MPLS and IP router

MPLS forwarding tables



Lancaster University

Dynamic Host Configuration Protocol

Getting your IP address…

DHCP Background

Lancaster University

- Allows a computer to join an IP network without having a pre-configured IP address
  - Extension of earlier BOOTP
    - Runs over UDP/ IP
  - Temporarily binds IP address and other parameters to DHCP client
  - Provides framework for passing configuration information to hosts
- DHCP assigns a unique IP address
  - Simplifies installation and configuration of end systems
  - Allows for manual and automatic IP address assignment
  - May provide additional configuration information
    - DNS server, sub-netmask, default router, etc.
- Used by
  - ISPs to minimise set-up costs
  - LANs and organisational networks
  - Using DHCP means system administrators do not need to configure parameters individually for each client

DHCP Components

Lancaster University

- DHCP Server
  - Assigned to specific network
  - Configuration parameters
    - Pool of available IP addresses
    - Correct subnet masks
    - Network gateway
    - Name server addresses
  - DHCP Databases
    - 1st database for manual IP acquisition
      - Permanently bound to hardware address
    - 2nd database for pool of addresses
      - Dynamically assigned on request (FCFS)
- DHCP Clients
  - Automatically retrieve DCHP settings
  - Have to “speak” DHCP protocol

DHCP Procedure

Lancaster University

- Client broadcasts DHCP DISCOVER packet
  - Server answers
- DHCP servers lease addresses to clients
  - Client sends request
  - Server allocates address from an address pool
  - Server adds client to (lease) database *with timeout*
  - Server replies to client with address, servers, …
- Client sends subsequent request to renew address lease
  - After ½ the lease time client can renew the lease
  - Provided not timed-out, server sends same address

DHCP Leases

Lancaster University

- Address Usage
  - After address has expired client must stop using address and acquire a new address
  - If there are more than one DHCP server client can select the best “offer”
- Address Leases
  - Manuel Lease
    - Network manager explicitly assigns all IP addresses
  - Automatic Lease
    - DHCP server permanently assigns some addresses and dynamically others
  - Dynamic Lease
    - DHCP server dynamically assigns IP addresses for a specific period of time when permanent address is not required

DHCP Message Format

Lancaster University

OpCode	Hardware Type	Hardware Address Length	Hop Count
Number of Seconds		Unused (in BOOTP) Flags (in DHCP)	
Transaction ID			
Client IP address			
Your IP address			
Server IP address			
Gateway IP address			
Client hardware address (16 bytes)			
Server host name (64 bytes)			
Boot file name (128 bytes)			
Options			

(There are >100 different options)

Message Fields

Lancaster University

- OpCode: Indicates a request or a reply
  - 1 Request
  - 2 Reply
- HWtype: The type of Link Layer hardware, e.g.
  - 1 Ethernet, 6 IEEE 802 networks
- HA Length: Hardware address length in bytes, e.g.
  - Ethernet and token-ring both use 6 bytes.
- Hop Count: The client sets this to 0. It is incremented by a router that relays the request to another server and is used to identify loops.
  - RFC 951 suggests that a value of 3 indicates a loop.
- Number of Seconds: Set by the client; giving the elapsed time (seconds) since client started boot process

Message Fields (cont.)

Lancaster University

- Flags field:
  - Most significant bit is used as a broadcast flag
  - Other bits must be set to zero, are reserved for future use
  - *Note:* normally, DHCP servers attempt to deliver DHCP messages directly to a client using unicast delivery. The destination address in the IP header is set to the DHCP Your IP address and the MAC address is set to the DHCP client hardware address. If a host is unable to receive a unicast IP datagram until it knows its IP address, then this broadcast bit must be set (=1) to indicate to the server that the DHCP reply must be sent as an IP and MAC broadcast. Otherwise this bit must be set to zero
- Transaction ID: A random number used to match this boot request with the response it generates
- Client IP address: Set by the client
  - Either client's known IP address, or 0.0.0.0
- Your IP address: Set by the server if the client IP address field was 0.0.0.0
- Server IP address: Set by the server

Message Fields (cont.)

Lancaster University

- Router IP address: This is the address of a BOOTP relay agent, not a general IP router to be used by the client. It is set by the forwarding agent when BOOTP forwarding is being used
- Client hardware address: Set by the client
  - MAC address *or* client identifier option that is not used
- Server host name: Optional server host name terminated by X'00'
- Boot file name: The client either leaves this null or specifies a generic name, such as router, indicating the type of boot file to be used.
  - In a DHCPDISCOVER request this is set to null. The server returns a fully qualified directory path name in a DHCPOFFER request. The value is terminated by X'00'.
- Options: Subnet Mask, Name Server, Hostname, Domain Name, Forward On/Off, Default IP TTL, Broadcast Address, Static Route, Ethernet Encapsulation, X Window Manager, X Window Font, DHCP Msg Type, DHCP Renewal Time, DHCP Rebinding, Time SMTP-Server, SMTP-Server, Client FQDN, Printer Name, …

DHCP Message Type

Lancaster University

Value	Message Type
1	DHCPDISCOVER
2	DHCPOFFER
3	DHCPREQUEST
4	DHCPDECLINE
5	DHCPACK
6	DHCPNACK
7	DHCPRELEASE
8	DHCPINFORM

Message type is sent as an option

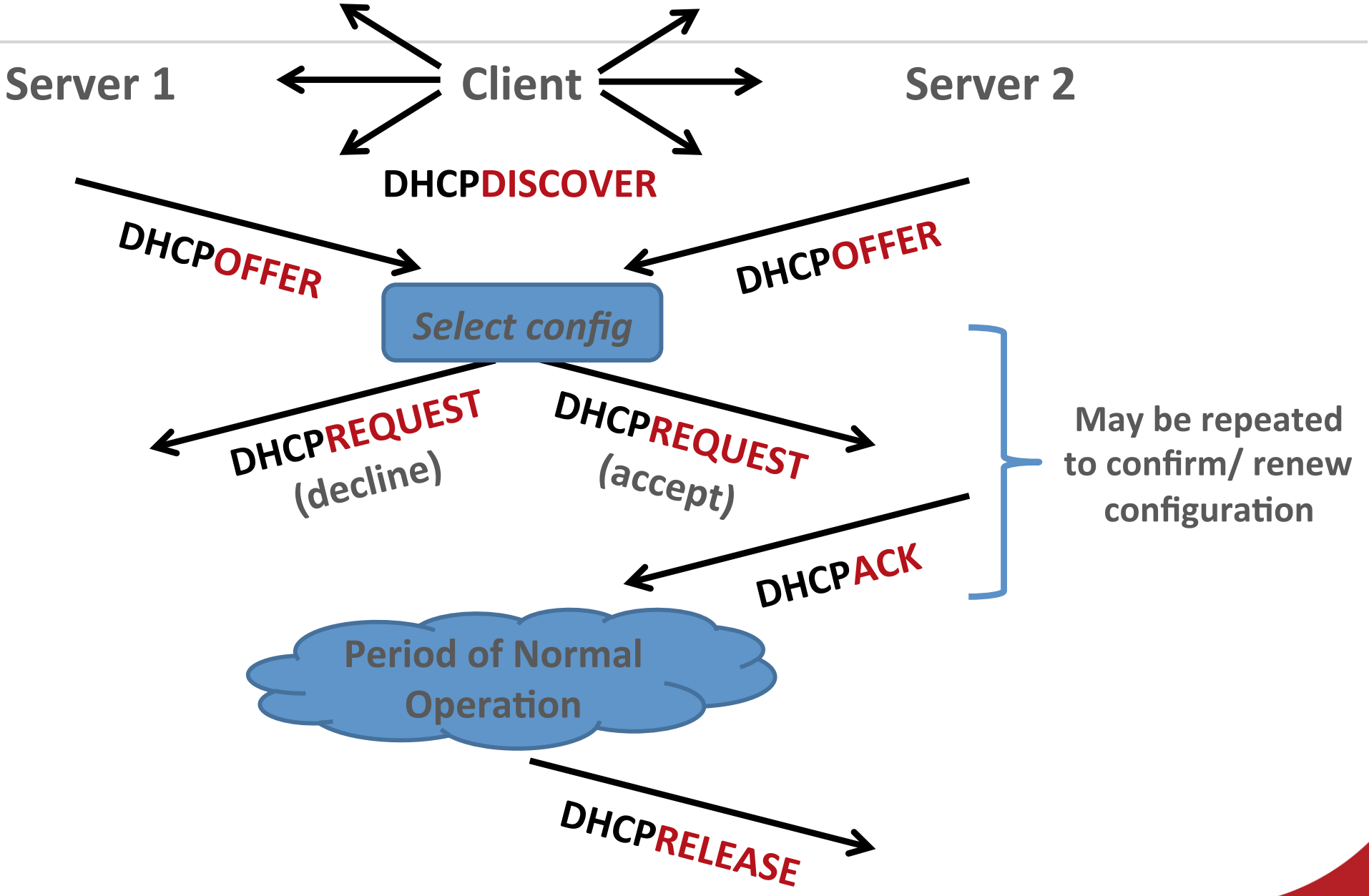
Message Types

Lancaster University

- DHCPDISCOVER: Broadcast by a client to find available DHCP servers
- DHCPOFFER: Response from a server to a DHCPDISCOVER and offering IP address and other parameters
- DHCPREQUEST: Message from a client to servers that does one of the following:
  - Requests the parameters offered by one of the servers and declines all other offers
    - Broadcast message
  - Verifies a previously allocated address after a system or network change (a reboot for example)
  - Requests the extension of a lease on a particular address
- DHCPACK: Acknowledgement from server to client with parameters, including IP address
- DHCPNACK: Negative acknowledgement from server to client, indicating that the client's lease has expired or that a requested IP address is incorrect
- DHCPDECLINE: Message from client to server indicating that the offered address is already in use
- DHCPRELEASE: Message from client to server canceling remainder of a lease and relinquishing network address
- DHCPINFORM: Message from a client that already has an IP address (manually configured for example), requesting further configuration parameters from the DHCP server

DCHP Protocol in Use

Lancaster University



Cisco DHCP Configuration

Lancaster University

```
! Hold some addresses back for static allocation to servers
!
ip dhcp excluded-address 194.80.37.1 194.80.37.63
!
! Setup DHCP server incl. details it should pass to clients
!
ip dhcp pool dhcp-37
   network 194.80.37.0 255.255.255.128
   default-router 194.80.37.1
   dns-server 10.20.5.100 194.168.88.2
   domain-name cs-lab.co.uk
   accounting default
!
! Configuration for the matching network interface
!
interface GigabitEthernet0/1
 description cs-lab 194.80.37.0/25
 ip address 194.80.37.1 255.255.255.128
```

Just a convenient name

Configuring fixed address for client

Lancaster University

```
ip dhcp pool fixed-ip1.cs-lab.co.uk
   host 194.80.37.16 255.255.255.128
   client-identifier 018c.a982.b0b8.38
   client-name fixed-ip1
   default-router 194.80.37.1
   dns-server 10.20.5.100 194.168.88.2
   domain-name cs-lab.co.uk
   accounting default
!
ip dhcp pool fixed-ip2.cs-lab.co.uk
   host 194.80.37.18 255.255.255.128
   hardware-address 8ca9.82b0.b838
   client-name fixed-ip2
   default-router 194.80.37.1
   dns-server 10.20.5.100 194.168.88.2
   domain-name cs-lab.co.uk
   accounting default
```

Convenient name

If client sends ID instead of MAC address Normally interface type followed by MAC address e.g. 01 (Ethernet)

MAC Address

If client sends MAC address

DHCP Pros

Lancaster University

- Relieves the network administrator of manual configuration
- Device can be moved from network to network and automatically obtain valid configuration parameters for the current network
- IP addresses are only allocated when needed
  - It is possible to re-use IP addresses after lease
    - Especially considering mobile clients

  ➔ Reduction in the total number of addresses in use

DHCP Limitations

Lancaster University

- Server Issues
  - A machine to run the DHCP server continually is required
    - When DHCP server is unavailable, client is unable to access the enterprise's network
- Security Problems
  - Uses UDP, an unreliable and insecure protocol
  - DHCP is an unauthenticated protocol
    - When connecting to a network, the user is not required to provide credentials in order to obtain a lease
    - Malicious users with physical access to the DHCP-enabled network can instigate a denial-of-service attack on DHCP servers by requesting many leases from the server, thereby depleting the number of leases that are available to other DHCP clients
- DNS cannot be used for DHCP configured hosts

Lancaster University

IPv4 Address Problem

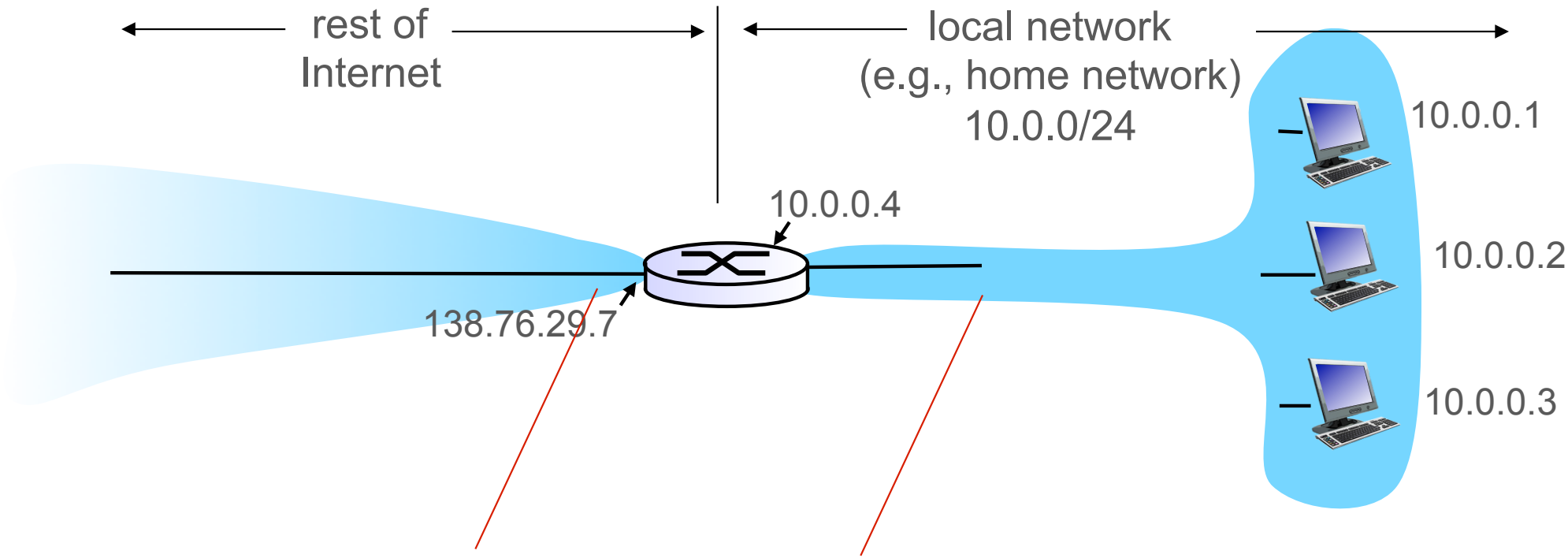
...or what to do when running out of address space

What is the Issues

Lancaster University

- Rapidly running out of IP addresses
  - Hierarchical allocation (waste)
  - IANA / RIPE have allocated all main blocks
- Solutions include
  - Network Address Translation (NAT)
  - IP version 6 (IPv6) – bigger addresses!

NAT: network address translation



all datagrams *leaving* local network have *same* single source NAT IP address: 138.76.29.7, different source port numbers

datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

NAT Basic Idea

Lancaster University

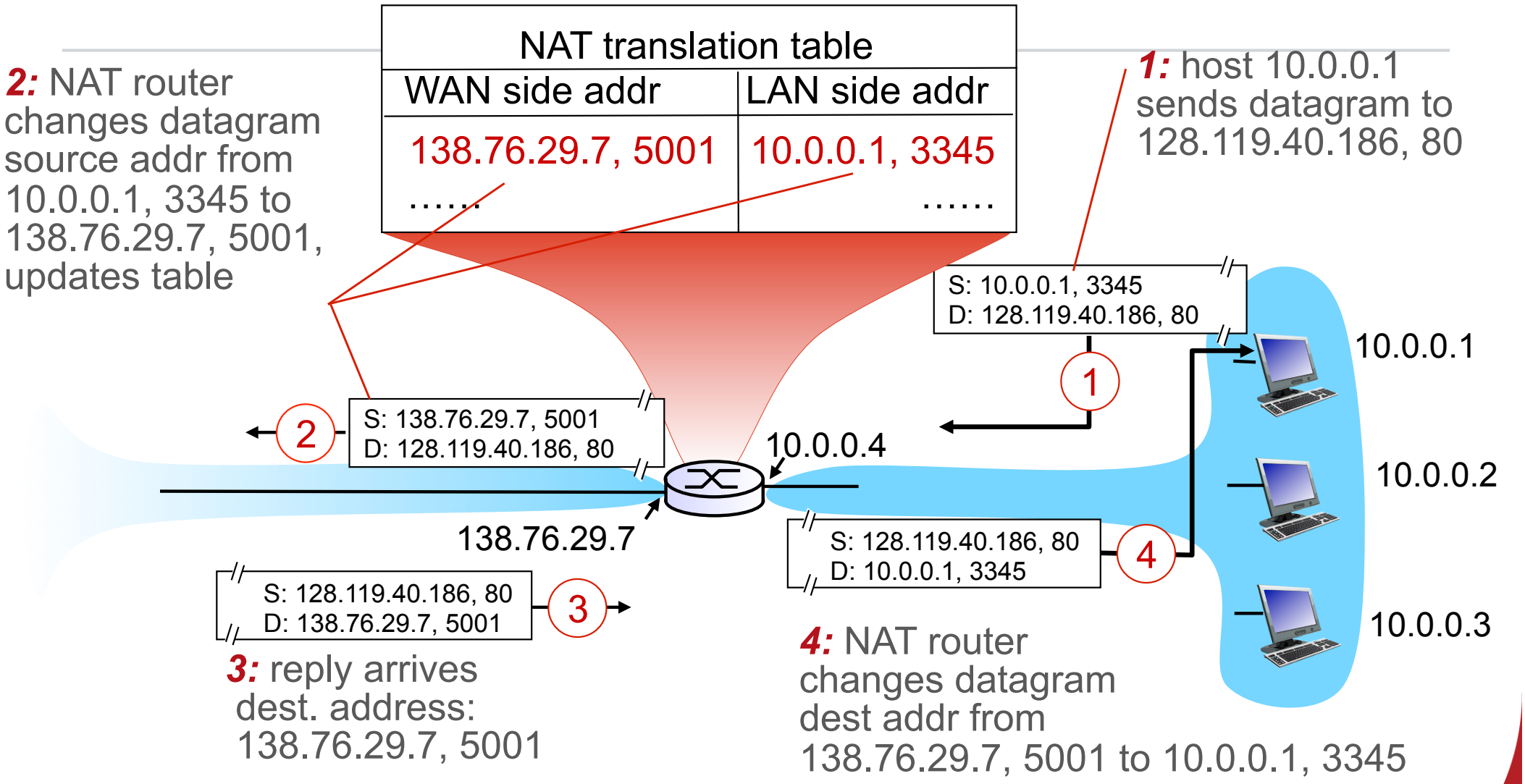
- Local network uses just *one IP address* as far as outside world is concerned:
  - Range of addresses not needed from ISP
    - Just one IP address for all devices
  - Can change addresses of devices in local network without notifying outside world
  - Can change ISP without changing addresses of devices in local network
  - Devices inside local net not explicitly addressable, visible by outside world (a security plus)

NAT Implementation

Lancaster University

- NAT router functionality
  - Outgoing datagrams:
    - replace (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
      - . . . remote clients/servers will respond using (NAT IP address, new port #) as destination addr
    - remember (in NAT translation table) every (source IP address, port #) to (NAT IP address, new port #) translation pair
  - Incoming datagrams:
    - replace (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

NAT Example


NAT translation table
WAN side addr
LAN side addr
138.76.29.7, 5001
10.0.0.1, 3345
.....
.....
2: NAT router changes datagram source addr from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table
1: host 10.0.0.1 sends datagram to 128.119.40.186, 80
S: 10.0.0.1, 3345
D: 128.119.40.186, 80
1
10.0.0.1
2
S: 138.76.29.7, 5001
D: 128.119.40.186, 80
10.0.0.4
10.0.0.2
138.76.29.7
S: 128.119.40.186, 80
D: 10.0.0.1, 3345
4
S: 128.119.40.186, 80
D: 138.76.29.7, 5001
3
10.0.0.3
3: reply arrives dest. address: 138.76.29.7, 5001
4: NAT router changes datagram dest addr from 138.76.29.7, 5001 to 10.0.0.1, 3345

NAT Summary

Lancaster University

- NAT allows to assign addresses locally and circumvent IPv4 address shortage
  - 16-bit port-number field:
    - 60,000 simultaneous connections with a single LAN-side address!
- NAT is controversial:
  - routers should only process up to layer 3
  - violates end-to-end argument
    - NAT possibility must be taken into account by app designers, e.g., P2P applications
  - address shortage should instead be solved by IPv6

Summary

Lancaster University

- MPLS
  - How to establish a path from source to destination
    - Establish routes for flows
  - Relaxes strict packet switching paradigm of the Internet
  - Digital Transmission
    - Basics of how bits are transmitted over the network
- DHCP
  - Dynamic network address configuration
    - Less administrative overhead
  - Protocol structure and message types
    - Step-by-step how DCHP works
  - Advantages & limitations
- NAT
  - IP Address Problem
  - NAT .Implementation and Process
    - Step-by-step how NAT works
  - Advantages & limitations

Questions?

Lancaster University

Questions

Lancaster University

- Multiple Choice
  - MPLS routing bases routing path on
    - A) Destination address only B) IP and Data Link address C) Source and Destination Address D) Static Routing Tables
  - What is the purpose of DHCPREQUEST ?
    - I) Requests the parameters offered a server, II) Declines offers of a server, III) Verifies a previously allocated address, IV) Requests extension of lease on a particular address
- Explain advantages and disadvantages of MPLS compared to standard IP routing.
- What is the idea behind NAT and what are its advantages?

Lancaster University

SCC203 Computer Networks: Transport Layer - UDP

Dr Andreas Mauthe (a.mauthe@lancaster.ac.uk)
Dr Andrew Scott (a.scott@lancaster.ac.uk)

Contents

Lancaster University

- Outline
  - Transport Layer
    - Transport Layer Services
    - Transport Layer Functions
  - Internet Transport Layer
  - User Datagram Protocol – UDP
    - UDP Features
    - Sockets
  - Revision
- Objectives
  - To get familiar with the characteristics and functionality of the Transport Layer
    - You should know the services and functionality of the Transport layer
    - You should be able to distinguish between OSI and Internet Transport Layer concepts
  - To understand how UDP works and get familiar of central programming concepts
    - You should know the features of UDP.
    - You should become familiar with Sockets as central programming abstraction

Lancaster University

Reminder

What you should remember from the last lecture

Questions

Lancaster University

- Multiple Choice
  - MPLS routing bases routing path on
    - A) Destination address only B) IP and Data Link address C) Source and Destination Address D) Static Routing Tables
  - What is the purpose of DHCPREQUEST ?
    - I) Requests the parameters offered by a server, II) Declines offers of a server, III) Verifies a previously allocated address, IV) Requests extension of lease on a particular address
- Explain advantages and disadvantages of MPLS compared to standard IP routing.
- What is the idea behind NAT and what are its advantages?

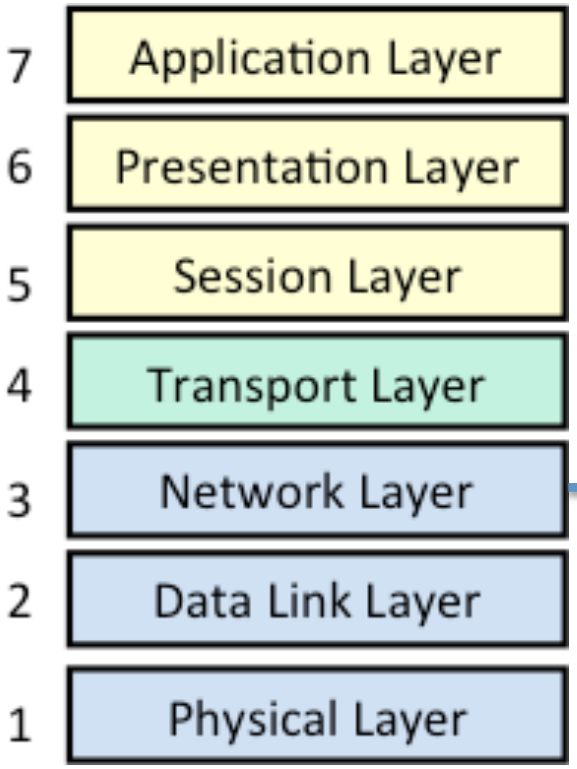
Lancaster University

Transport Layer

Above the Network: end-to-end communication

Looking back, Looking forward

Lancaster University

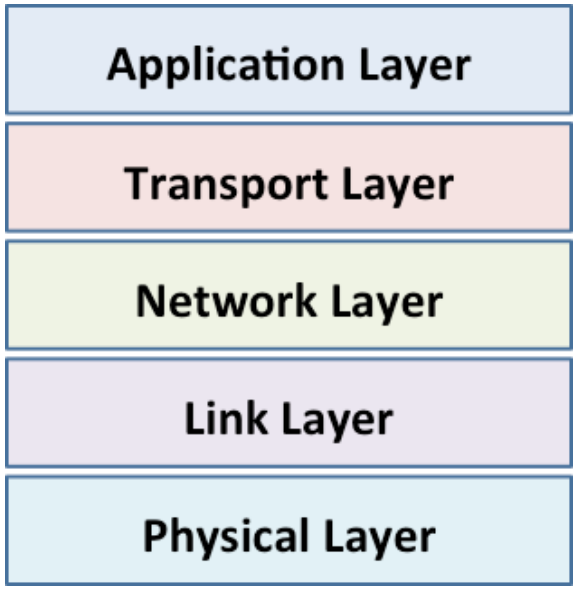


What's to come

- Not quite finished with Network Layer
  - Further routing
  - Advanced issues
- Transport Layer
  - The End-to-end argument

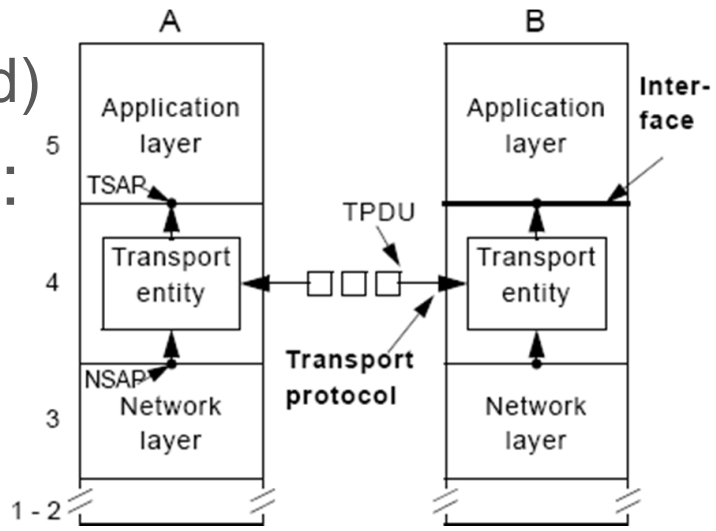
We covered

- Physical layer, Link Layer, Network Layer
- Communication modes and basic concepts
- Addressing issues
- Etc.



Transport Layer Function

- Task
  - To provide data transport between
    - Processes (process-to-process)
    - At different end-systems (end-to-end)
  - Requirements: to provide service:
    - Reliably
    - Efficiently
    - At low cost
    - Independently
      - From particularities of specific network and lower layers



Transport Layer Service

Lancaster University

- Service Types
  - Connection oriented service
    - 3 phases: connection set-up, data transfer, disconnect
  - Connectionless service
    - Transfer of isolated units
- Task
  - To isolate upper layers from technology, design and imperfections of subnet
  - Creates major boundary between provider and user of (reliable) data transmission service
- Aim:
  - To Improve the Network Service quality users and layers want to get from the network layer, e.g.:
    - Reliable service
    - Necessary time guarantees
- Implementation: transport entity
  - Software and/or hardware?
  - Software part usually contained within the kernel (process, library)

Reminder: Network Service

Lancaster University

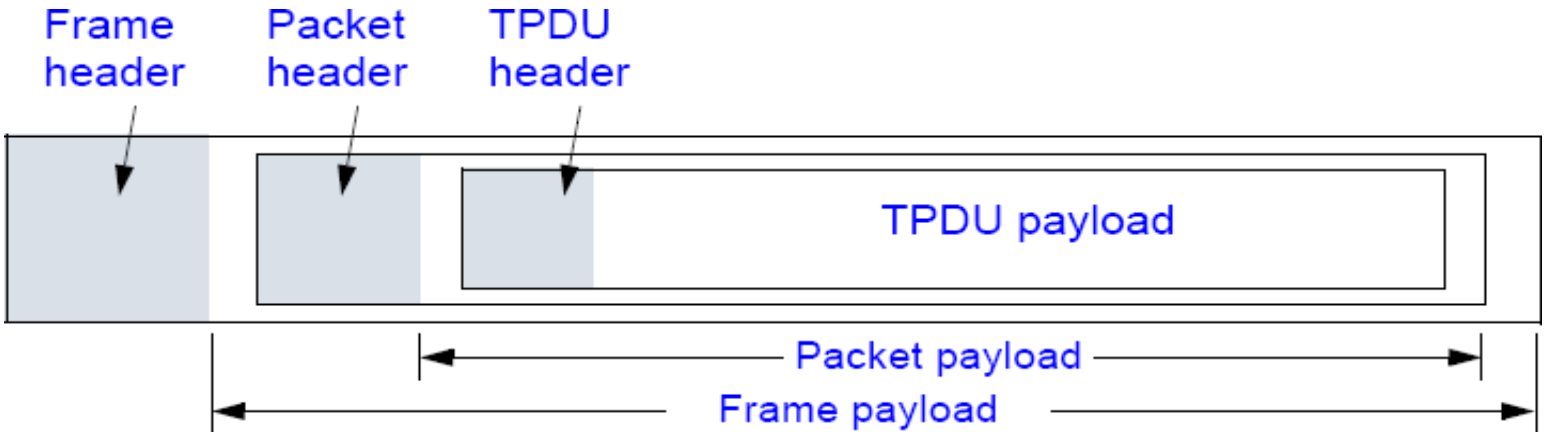
- Not to be self-governed or influenced by the user
- Independent from application & user
- Enables compatibility between applications, provides for example
  - Only connection oriented communications
    or
  - Only unreliable data transfer

Reminder: Layer Relationship

Lancaster University

Layer	Data Unit
Transport	TPDU/ Message
Network	Packet
Data Link	Frame
Physical	Bit Stream (bits & bytes)

- TBDU: Transport Protocol Data Unit
  - Part of Frame



Service Primitives

Lancaster University

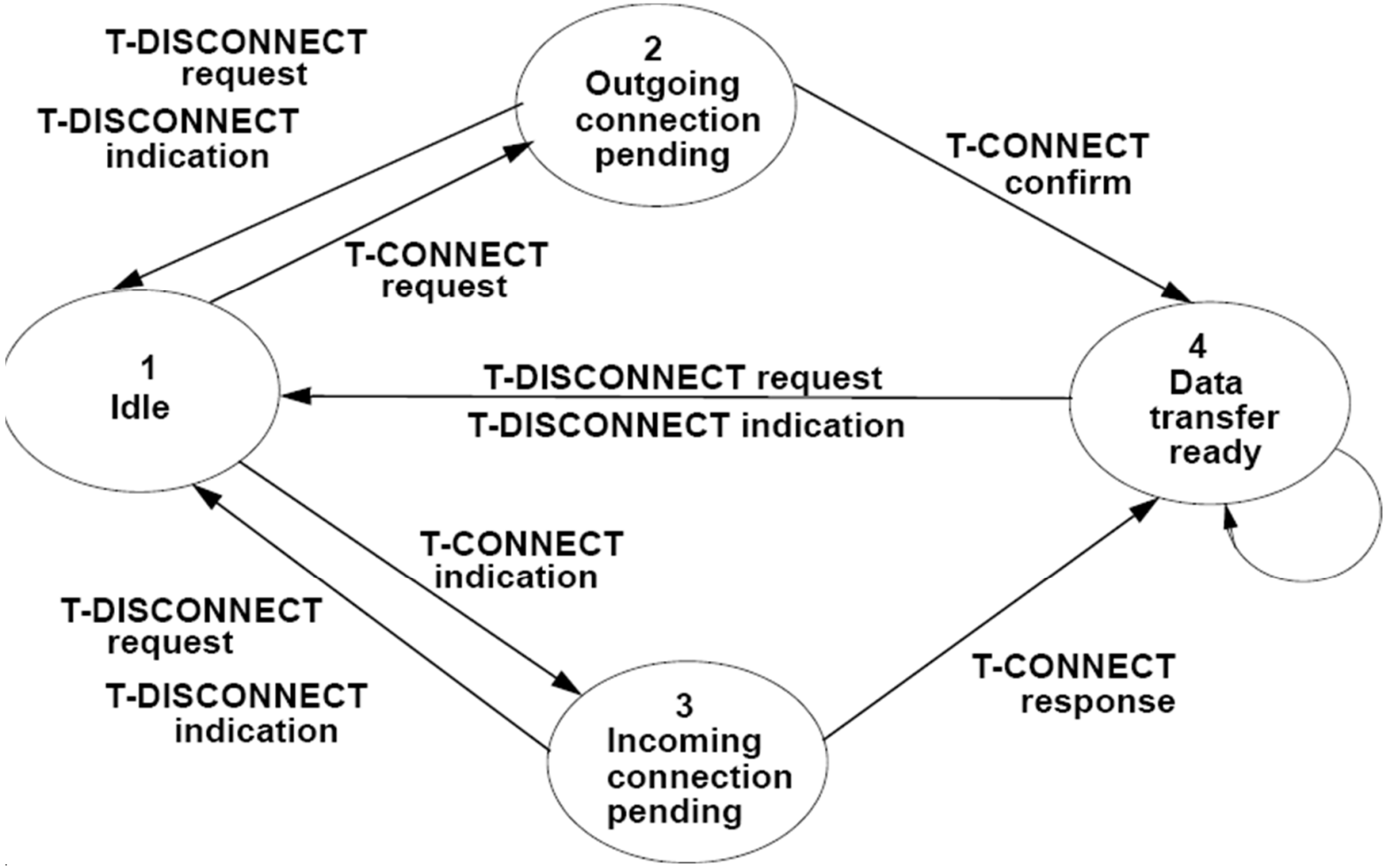
- Simple Transport Service

Primitive	Packet Sent	Meaning
Listen	(none)	Block until some process tries to connect
Connect	CONNECTION REQ	Actively attempt to establish a connection
SEND	DATA	Send Information
RECEIVE	(none)	Block until a DATA packet arrives
DISCONNECT	DISCONNECTION REQ	Request to release the connection

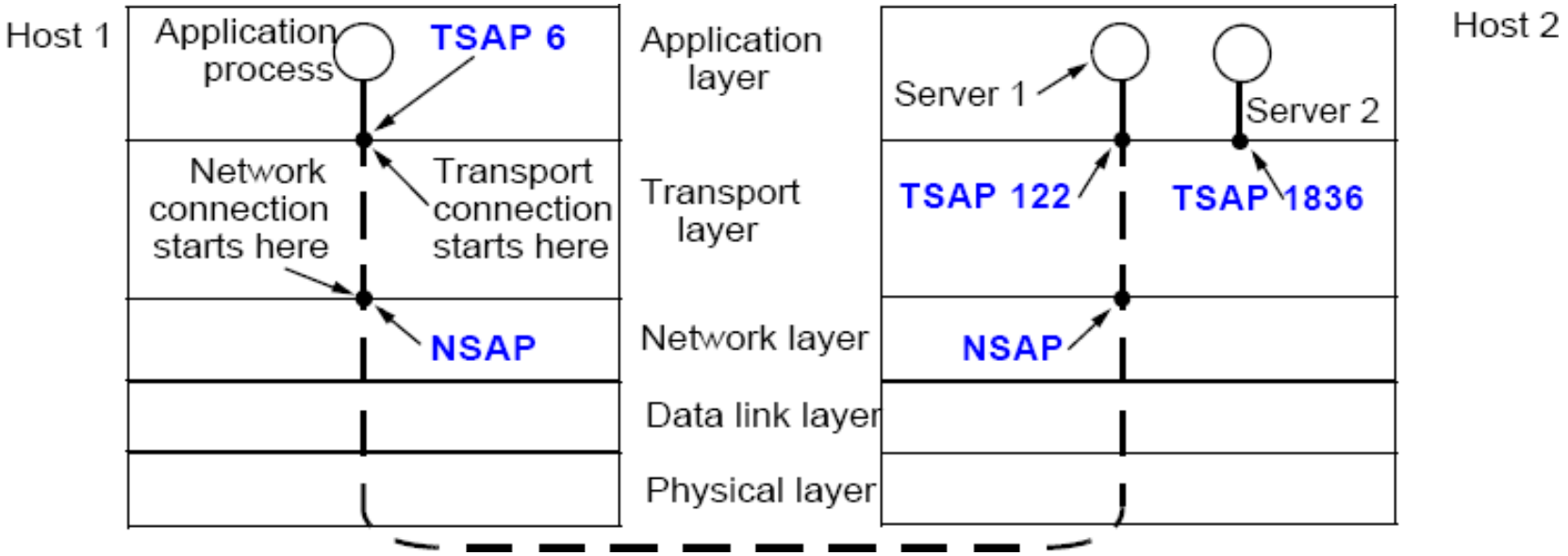
Connection Oriented Service

Lancaster University

- State Transition Diagram



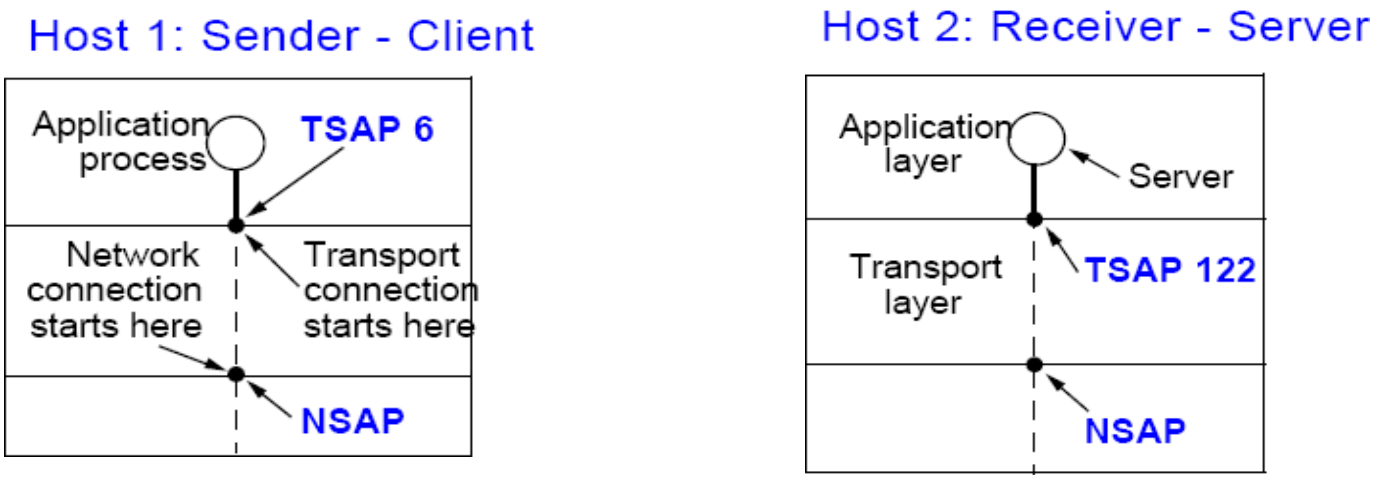
Addressing at the Transport Layer



- Transport Layer addresses have to identify communicating process
  - Sender (process) wants to address receiver (process)
    - Connection setup
    - Individual message exchange
  - Receiver (process) can be approached by the sender (process)
- Definition
  - *OSI/ISO*: Transport Address is a Transport Service Access Point (TSAP)
  - *Internet*: **Port**

Procedure

Lancaster University



- Server (Service Provider)
  - 1.) Connects itself to TSAP 122
    - Waits for service request (polling, signaling..)
  - 4.) Addresses Server
    - Incoming CONNECT REQ
  - 5.).....
- Client
  - 2.) Initiates communication via TSAP 6 to destination TSAP 122
    - CONNECT REQ
  - 3.) Transport System Client (Host 1)
    - Identifies NSAP, initiates communication at Network Layer, communicates with TL at server, informs TSAP 122 about connection request

How to find TSAP

Lancaster University

- Implicit knowledge
  - Services that are well known and often used have pre-defined TSAPs as "well-known ports" of a transport protocol
    - e.g., stored in /etc/services file at UNIX systems
      - "Time of day" service
  - Pros & Cons

    + works well for small number of stable services

    - not suitable for user specific processes

    - existing for short time, no known TSAP address

    - waste of resources; seldom used servers active and listening
- Initial connection protocol
  - Process server acting as proxy for less often used servers process server listens to a set of ports at same time
    - Waits for connection requests
    - Creates the appropriate service provider process
    - Transfers connection and desired service waits for further requests
  - Pros & Cons

    + works well for servers which can be created on demands

    - not suitable if service exists independently of process server at another machine (e.g., file server)

How to find TSAP (cont.)

Lancaster University

- Name Server
  - Server process already exists
    - Client addresses Name server (establishing connection)
    - Client specifies the service as an ASCII data set example “name of day”
    - Name server supplies TSAP
    - Client disconnects from name server
    - Client addresses TSAP provided by name server
    - etc.
  - Pros & Cons
    + central knowledge
    - overhead (process & infrastructure)

Transport over Network Layer

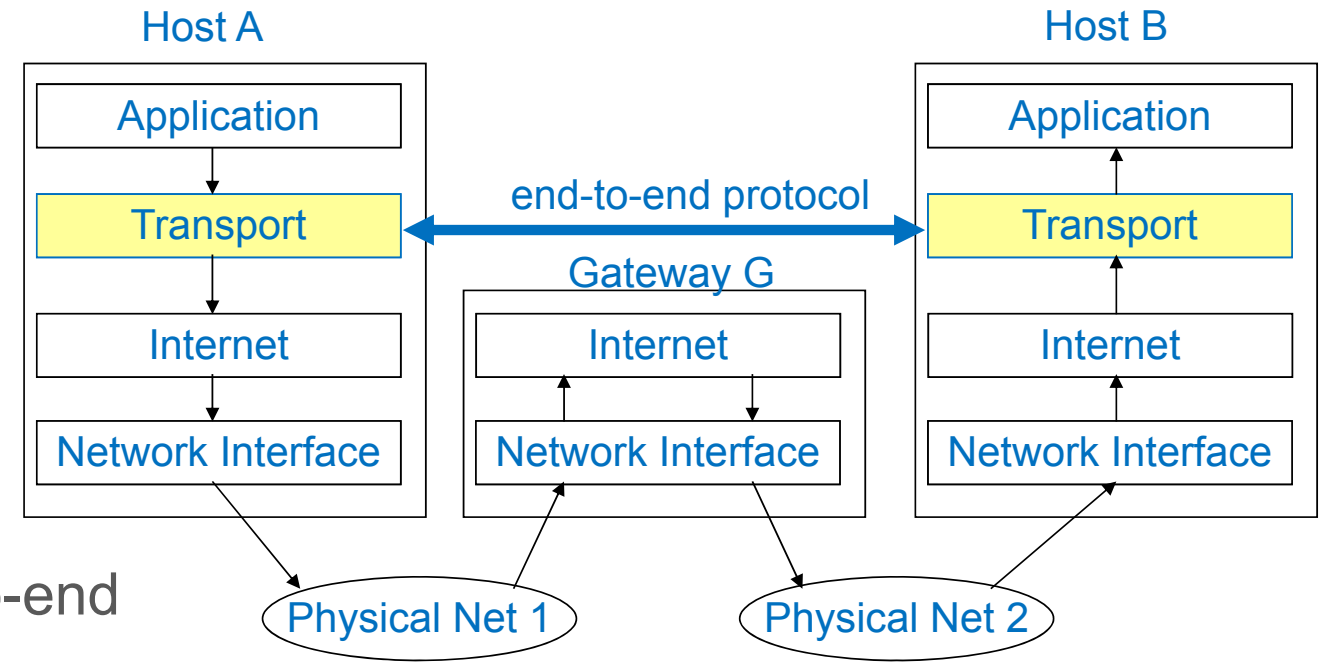
Lancaster University

- Multiplexing/ de-multiplexing
  - Grouping of transport connections by destination address
    - Each group is mapped to the minimum number of network connections
      - Too many L4-T connections per L3-N connection possibly poor throughput
      - Too few T connections per N connection possibly transfer costs too high
  - Motivation
    - Minimizing costs when num. of connections/ connection time represents the main cost factor
- Splitting/ recombination
  - Distributing TPDUs onto the various network connections
    - Usual algorithm: Round Robin
  - Motivation:
    - Implementation of Transport connections with high bandwidth requirements
  - Also known as “upward” multiplexing

Transport Layer in the Internet

What is above IP

Internet Transport Layer



- Lowest level end-to-end protocol
- Header generated by sender is interpreted only by destination routers / gateways view transport header as part of the payload
- Can add extra functionality to the best effort packet delivery service provided by IP
- Can make up for shortcomings of core network

Some Transport Layer Functions

Lancaster University

- Multiplexing/de-multiplexing data for multiple applications
  - Uses “port” abstraction
- Connection establishment
  - Logical end-to-end connection
- Error control
  - Hides unreliability of network layer from applications
    - Corruption, loss, duplication, reordering
- End-to-end flow control
  - To avoid flooding the receiver
- Congestion control
  - To avoid flooding the network

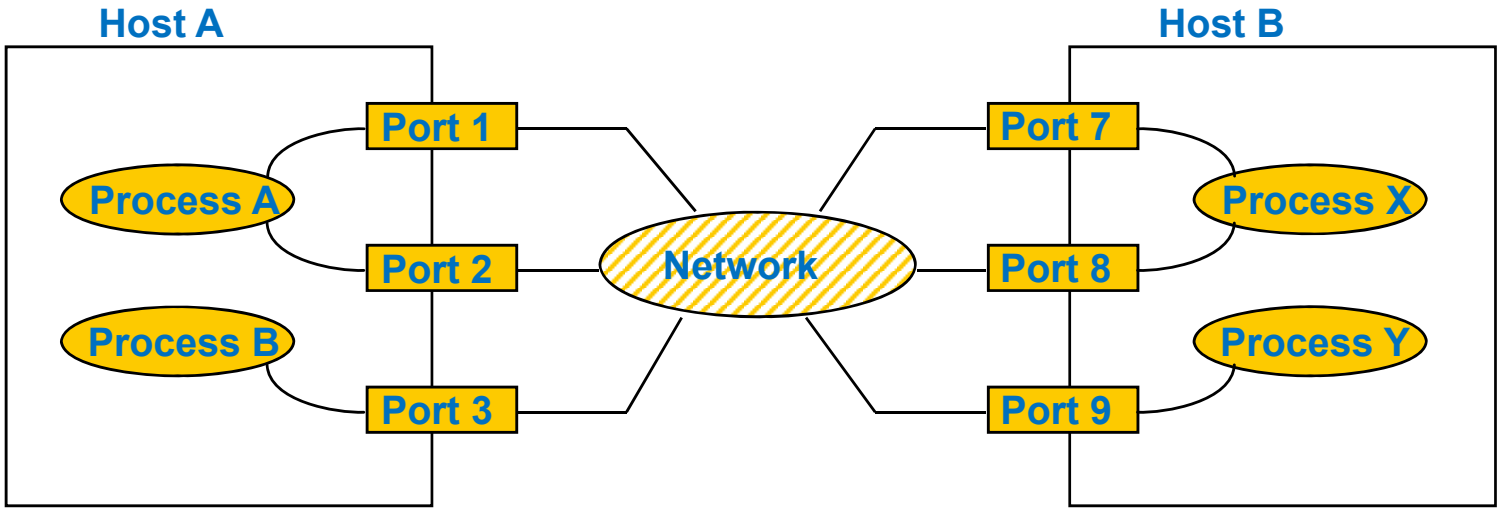
Addressing Abstraction: Port

Lancaster University

- Why not using Process ID?
  - Processes are generated/terminated dynamically
    - i.e. process number rarely know
  - Relationship (Service, Process) is not one to one and is not fixed
    - 1 process can supply multiple services
    - Various processes can provide same service
  - Not possible to replace processes receiving datagrams without informing all senders (e.g. on reboot)

  ➔ need to identify destination
    - Based on the implemented function
    - Without knowing the process that implements it
- Port
  - To communicate process needs to know
    - IP address
    - Port Number

Addressing Abstraction: Port (cont.)



- Protocol ports
  - Concept of abstract communication endpoint
  - Identified by a positive integer
  - Local operating system provides interface mechanism that processes use to specify a port or access it ports are in general buffered
  - To communicate with a “foreign” port
    - Sender needs to know
    - IP address of the destination device
    - Protocol port number of the destination within that device

Reserved Port Numbers

Lancaster University

#	Keyword	UNIX	Description
0			Reserved
1	TCPMUX		TCP Multiplex
5	RJE		Remote Job Entry
7	ECHO	echo	ECHO
9	DISCARD	discard	Discard
11	USERS	systat	Active Users
13	DAYTIME	daytime	Daytime
15		netstat	Network status program
17	QOUTE	qotd	Quote of the day
19	CHARGEN	chargen	Character Generator
20	FTP-DATA	FTP-DATA	FILE TRANSFER PROTOCOL (DATA)
21	FTP	FTP	FILE TRANSFER PROTOCOL
23	TELNET	TELNET	TERMINAL CONNECTIONS
25	SMTP	SMTP	SIMPLE MAIL TRANSFER PROTOCOL
37	TIME	time	Time
42	NAMESERVER	name	Host Name Server

TCP Port Assignments

Reserved Port Numbers (cont.)

Lancaster University

#	Keyword	UNIX	Description
43	NICNAME	whois	Who is
53	DOMAIN	nameserver	Domain Name Server
		rje	Any private remote job entry service
79	FINGER	finger	Finger
80	HTTP	HTTP	WWW
101	HOSTNAME	hostname	NIC Host Name Server
102	ISO-TSAP	Iso-tsap	ISO TSAP
103	X.400	x400	X.400 Mail Service
104	X400-SND	X400-snd	X.400 Mail Sending
110	POP3	POP3	REMOTE EMAIL ACCESS
111	SUN RPC	sunrpc	SUN Remote Procedure Call
113	AUTH	auth	Authentication Service
117	UUCP-PATH	Uucp-path	UUCP Path Service
119	NNTP	nntp	USENET News Transfer Protocol
129	PWDGEN		Password Generator Protocol
139	NETBIOS-SSN		NETBIOS Session Protocol
160 - 1023	Reserved		

Lancaster University

User Datagram Protocol - UDP

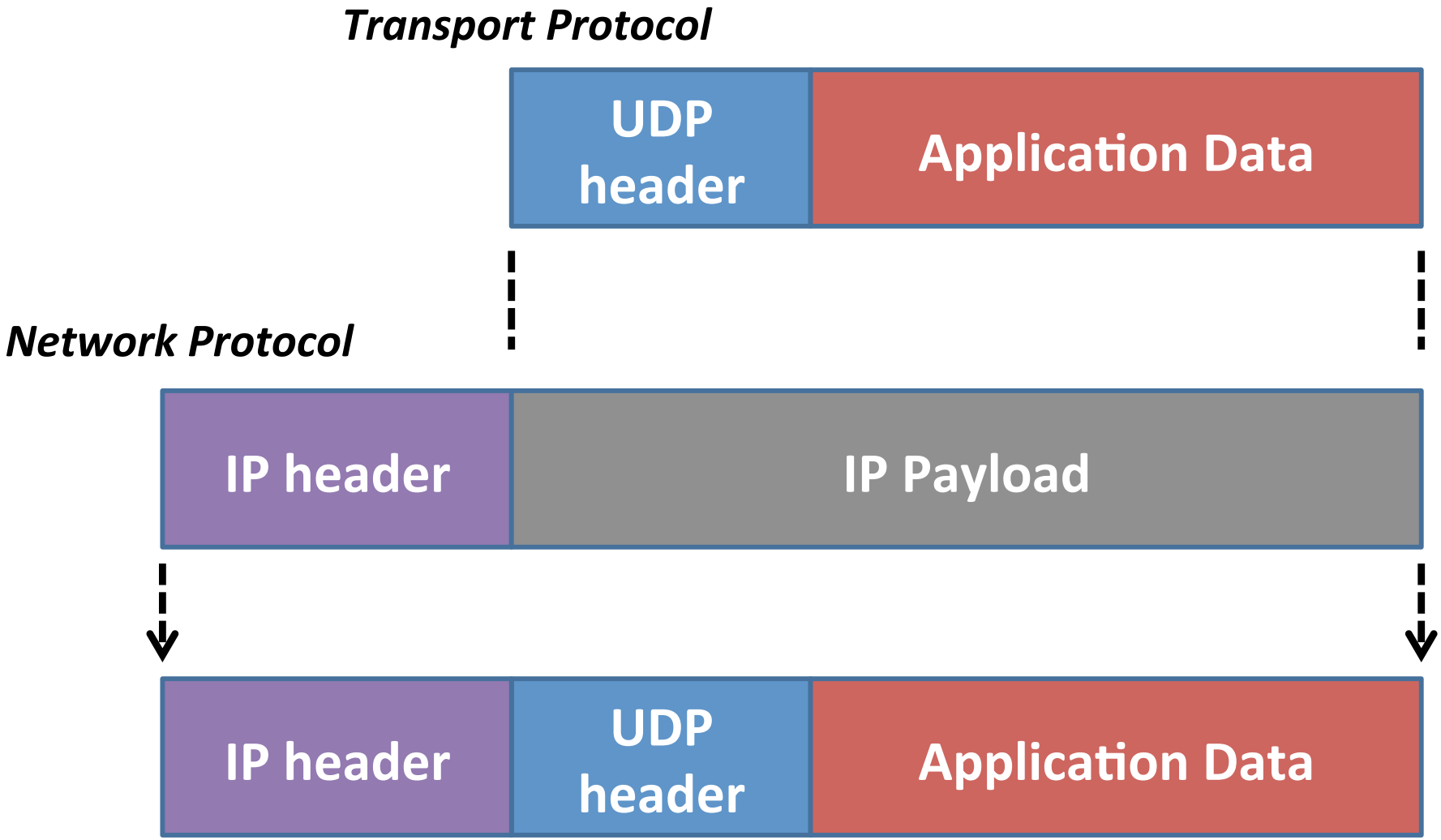
Simple Internet Transport Protocol

UDP Overview

Lancaster University

- Simple transport protocol
  - Best effort message delivery over IP
    - Specification RFC768
  - Unreliable
  - Connectionless
  - Message-oriented
- UDP is "*mostly IP with a short transport header*"
  - Source and destination port
  - Ports allow for dispatching of messages to receiver process

UDP and IP


Transport Protocol
UDP header
Application Data
Network Protocol
IP header
IP Payload
IP header
UDP header
Application Data

UDP Characteristics

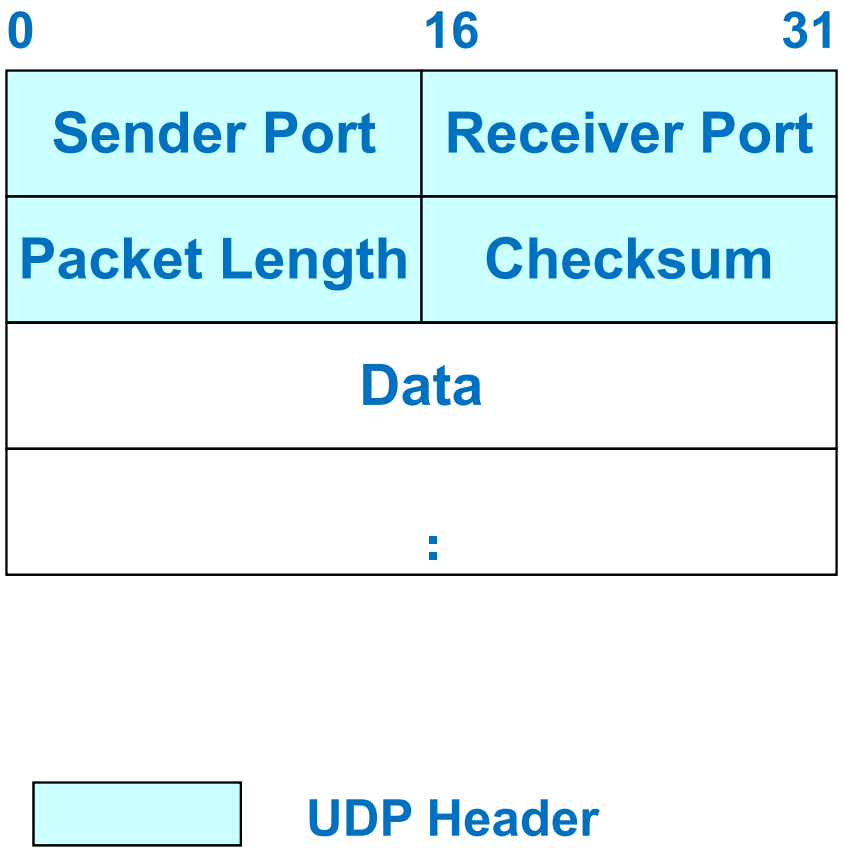
Lancaster University

- No connection establishment
- No QoS guarantees
  - No bandwidth or other performance guarantees
- No flow control
  - Application may transmit
    - As fast as it can/wants to and
    - As fast as the network permits
- No error control or retransmission
  - No guarantee about packet sequencing
  - Packet delivery to receiver not ensured
  - Possibility of duplicated packets
- May be used with broadcast / multicast and streaming

UDP Message Format

Lancaster University

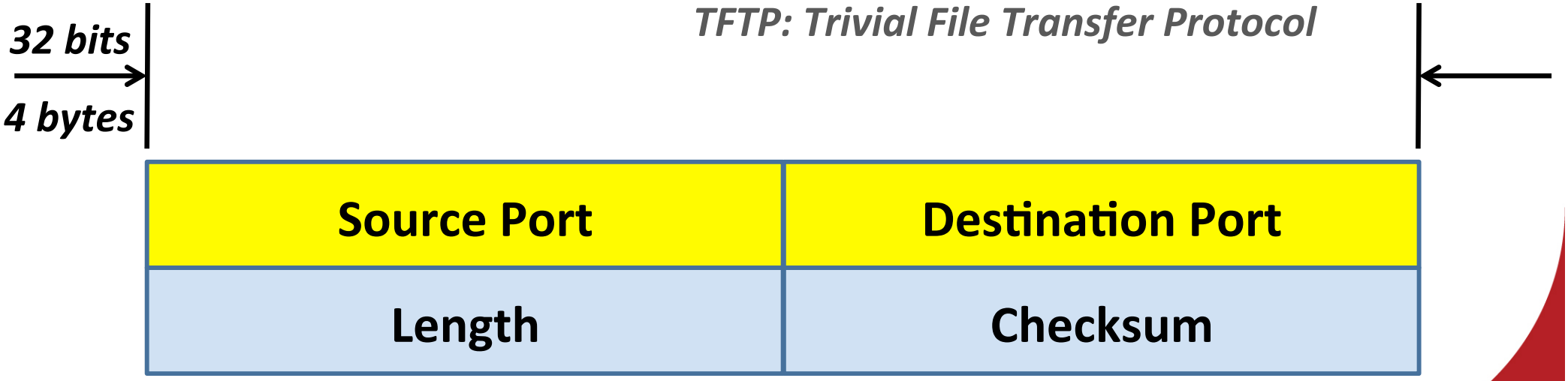
- Sender Port
  - 16 bit sender identification
    - Optional
    - use: possible response
- Receiver Port
  - Receiver identification
- Packet length
  - In bytes
    - Including UDP header)
  - minimum: 8 (byte)
    - i.e. header without data
- Checksum
  - Of header and data for error detection
    - Use of checksum optional



UDP Ports

Lancaster University

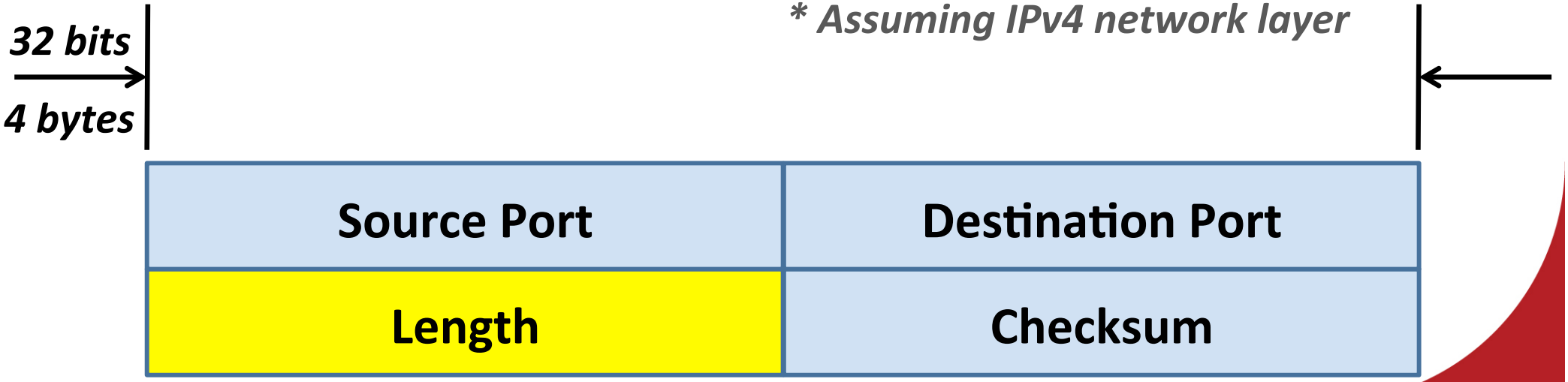
- Source and destination ports (both 16 bits)
- Example,
  - Source: ‘Random’ user port > 49151
  - Dest: ‘Well known’ destination port, such as 69 (TFTP)



UDP Length Limits

Lancaster University

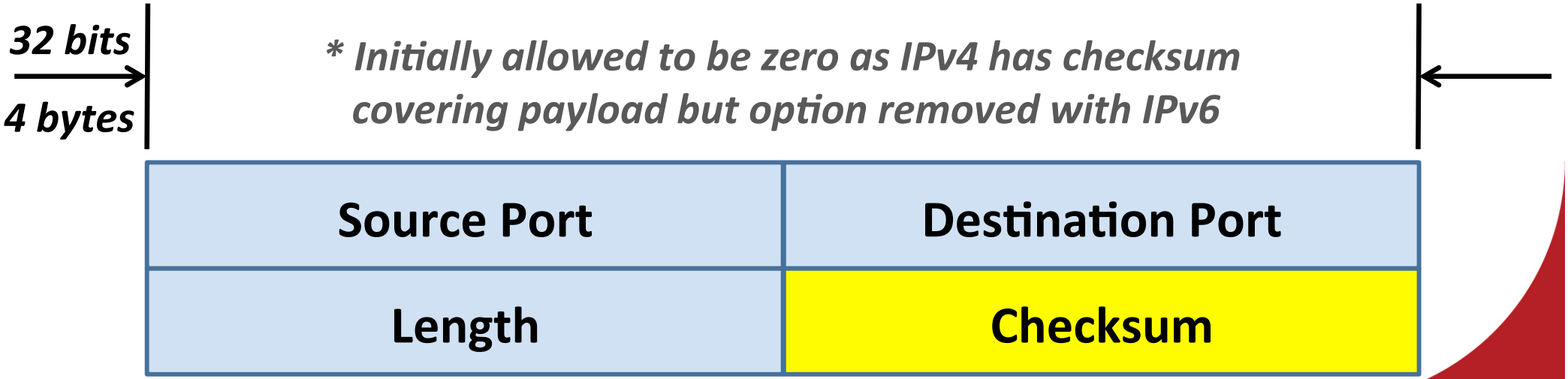
- Length in bytes (UDP header and data)
  - Minimum is 8 bytes (length of UDP header)
  - Maximum is
    - 65,535 – (UDP header) – (Minimum IP* header)
    - 65,535 – 8 – 20 = 65,507 bytes



UDP Checksum

Lancaster University

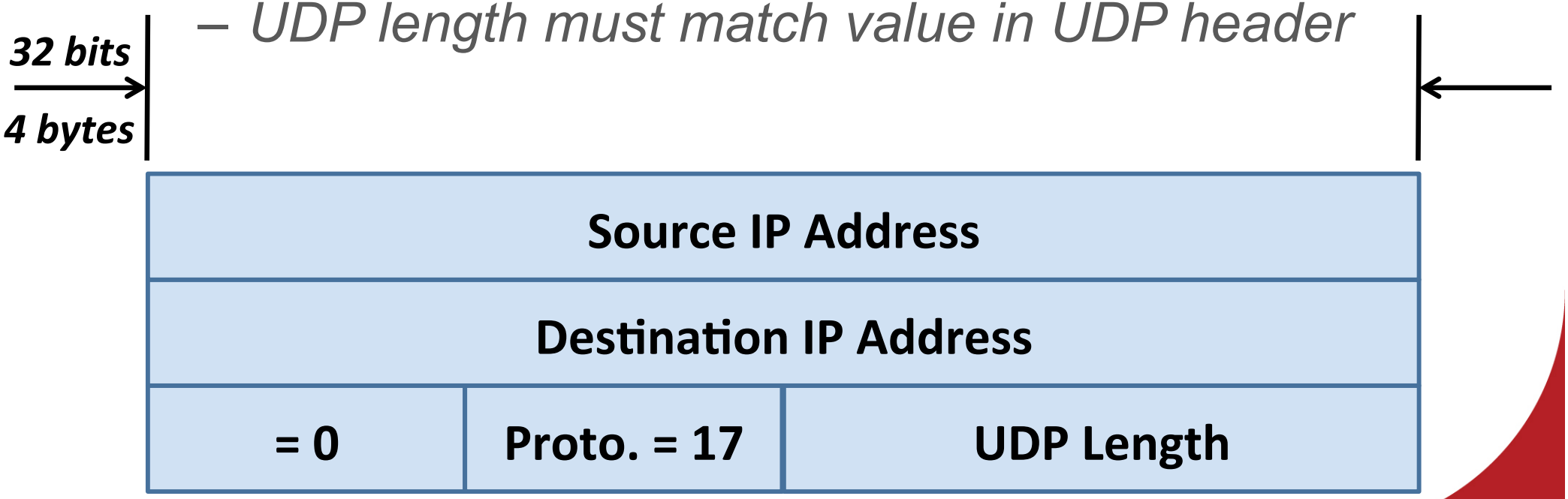
- Basically a 16 bit sum of words in packet
  - *Note:* we'll see same algorithm in detail when we look at IP
- Covers network-layer and UDP headers + payload
  - But UDP layer doesn't know internals of network layer…



UDP IPv4 Pseudo Header

Lancaster University

- UDP checksum based on ***pseudo header*** not real one
  - *Protocol must = 17 (UDP)*
  - *Source and dest. addresses must match real IP header*
  - *UDP length must match value in UDP header*



UDP Header Implementation

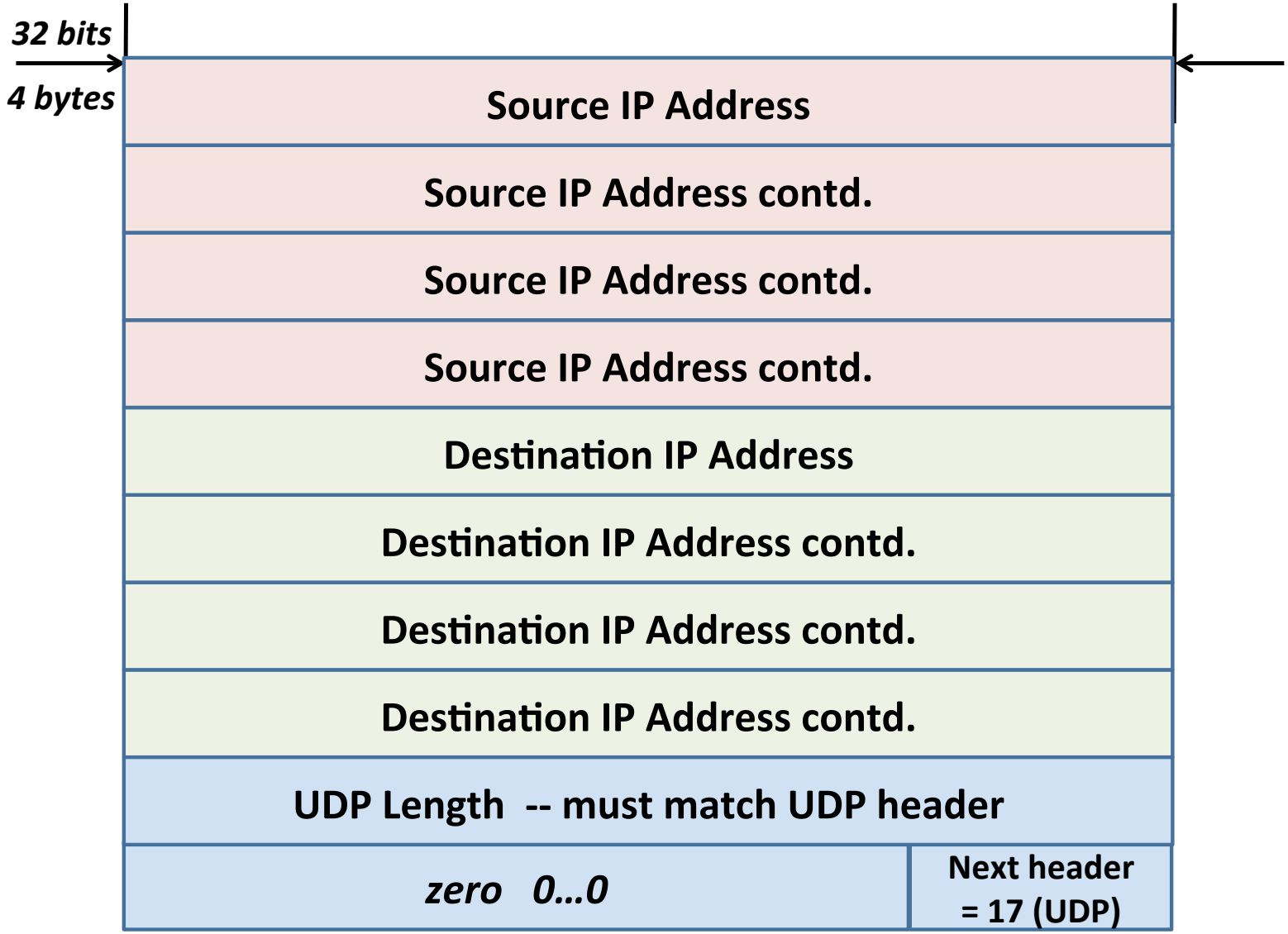
Lancaster University

```
struct udphdr {
    uint16_t        srcport;
    uint16_t        dstport;
    uint16_t        length;
    uint16_t        checksum;
} __attribute__((__packed__));
```

```
struct v4pseudo_hdr {
    uint32_t        srcip;
    uint32_t        dstip;
    uint8_t         zero;           // Always set to zero
    uint8_t         protocol;       // Decimal 17 (UDP)
    uint16_t        udp_length;
} __attribute__((__packed__));
```

UDP Pseudo Header

Lancaster University



Lancaster University

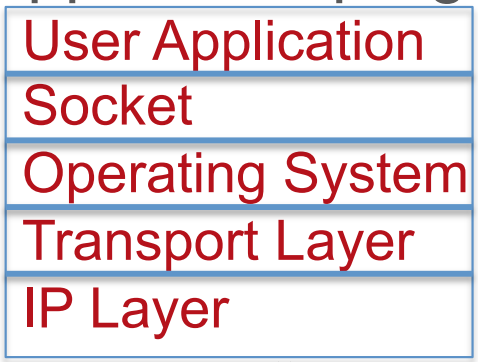
Sockets

... for self-study

Socket Basics

Lancaster University

- Interface that the OS provides to its networking subsystem
  - Used by application layer
  - Application programs "*plugs in*to" a socke*t*



- User sees "descriptor" - integer index or object handle
  - Like: FILE *, or file index from open()
  - Returned by socket() call
  - Programmer cares about Application Programming Interface (API)

Socket Basics (cont.)

Lancaster University

- End point determined by two things:
  - Host address: IP address is Network Layer
  - Port number: Is Transport Layer
- Two end-points determine a connection: socket pair
  - ex: `206.62.226.35,p21 + 198.69.10.2,p1500`
  - ex: `206.62.226.35,p21 + 198.69.10.2,p1499`

BSD Socket API

Lancaster University

- Developed for Berkeley Software Distribution Unix
  - *Basis for POSIX and Windows APIs*
- Extensions/ wrapper functions added for IPv6
  - *Allows applications to be protocol agnostic*
  - *Replaced old gethostbyname( ) and gethostbyaddr( ) with getnameinfo( ) and getaddrinfo( )*

Socket Functions

Lancaster University

- Functions pass address from user to OS

  `bind()`

  `connect()`

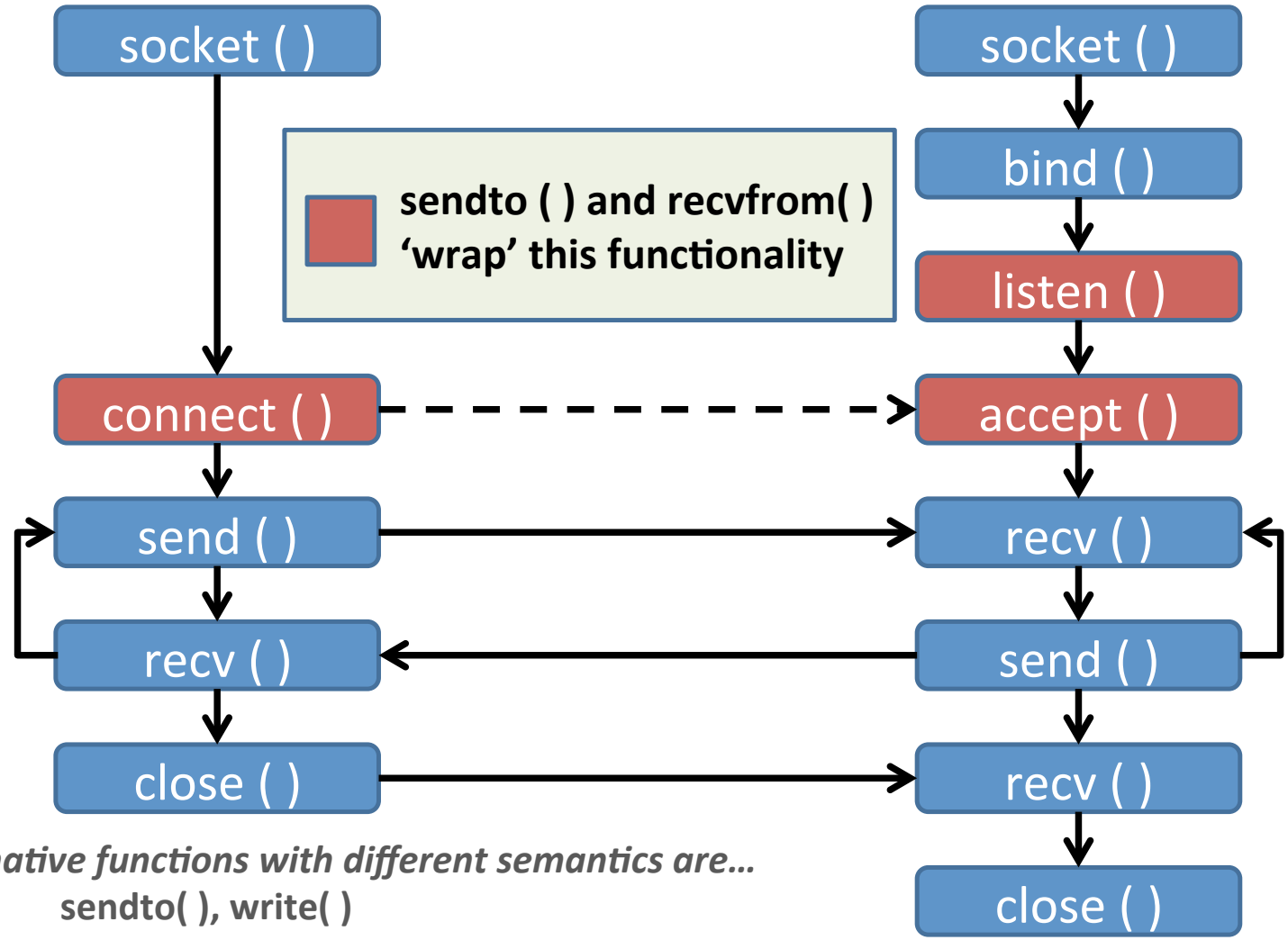
  `sendto()`

- Functions pass address from OS to user

  `accept()`

  `recvfrom()`

Socket API



Alternative functions with different semantics are…

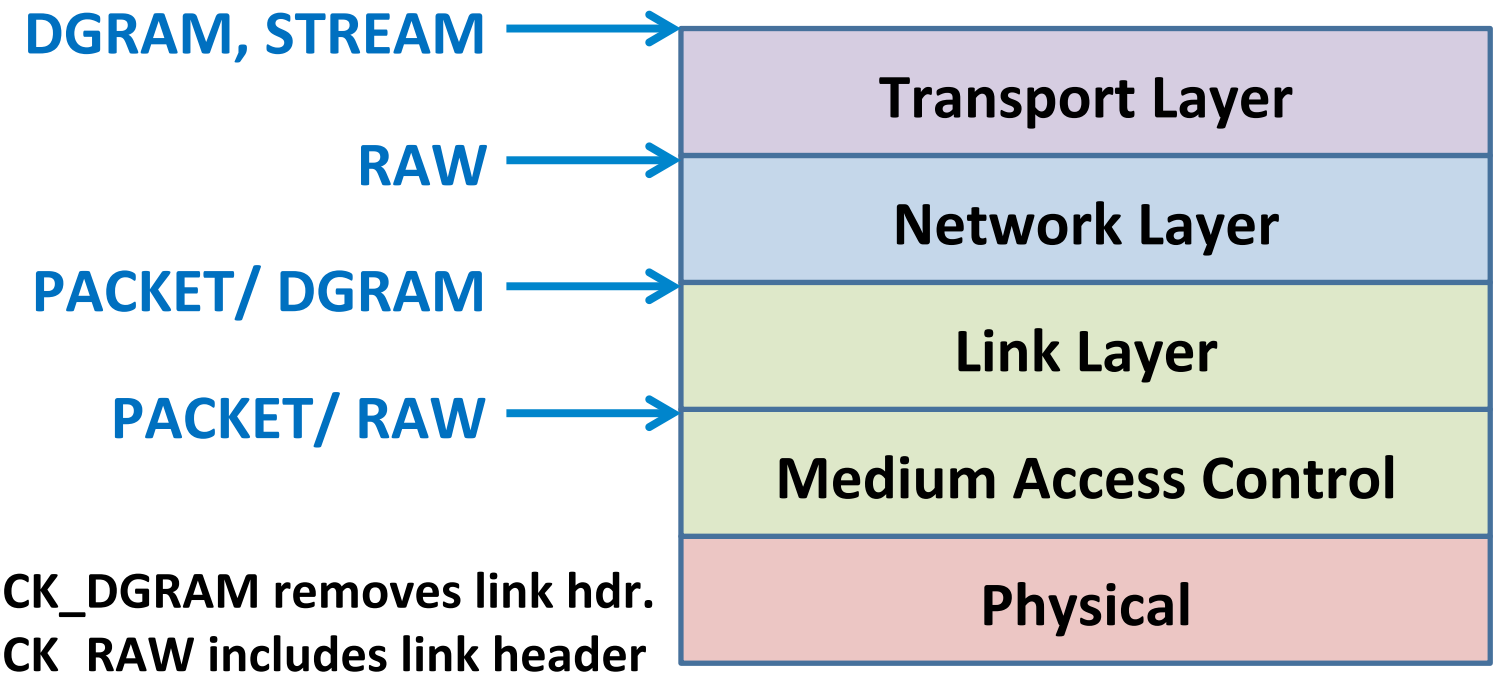
Send: sendto( ), write( )

Recv: readfrom( ), read( )

Close: shutdown( )

Socket Types

```
skt = socket (AF_UNSPEC, SOCK_DGRAM, 0);
skt = socket (AF_INET,   SOCK_RAW,   UDP); // incl IP hdr
Skt = socket (AF_PACKET, SOCK_DGRAM, htons(ETH_P_IP) );
skt = socket (AF_PACKET, SOCK_RAW,   htons(ETH_P_ALL));
```



Note:

AF_PACKET/SOCK_DGRAM removes link hdr.
AF_PACKET/SOCK_RAW includes link header

AF_UNSPEC now used to accept IPv4 or IPv6, AF_INET and AF_INET6 respectively

Network Byte Order

Lancaster University

- Different architectures use different byte orders
  - *Need common format for all network transmissions*
- Internet uses Most Significant Byte first
  - *x86 uses Least Significant Byte first*

```
SYNOPSIS
       #include <arpa/inet.h>  // Some systems use <netinet/in.h>

       uint32_t htonl( uint32_t hostlong );

       uint16_t htons( uint16_t hostshort );

       uint32_t ntohl( uint32_t netlong );

       uint16_t ntohs( uint16_t netshort );
```

Summary

Lancaster University

- Network Address Translation
  - Means of extending IP address range
    - Allows to hide (sub-)network structure
    - Violates end-to-end argument
- Transport Layer
    - Service abstraction and functionality
    - Difference between Internet and OSI view on Transport Layer
- UPD
  - Basic idea & functionality

Questions

Lancaster University

Questions??

Lancaster University

- Multiple Choice
  - Multiplexing at the Transport Layer refers to
    - A) Distributing TPDUs over multiple network connections B) Multi-homing devices C) Grouping of transport connections by destination address D) Providing connection less service over Network
  - Which of these is true for UDP?
    - I) is connection-oriented , II) is unreliable, III) is message oriented , IV) provides bandwidth guarantees
- What is the function of the Transport Layer? Which Transport Services are provided? Describe two types of Transport Services and name corresponding protocols and applications making use of the different service types.
- What are the main differences between the Network and the Transport Layer? Why are these two layers necessary?

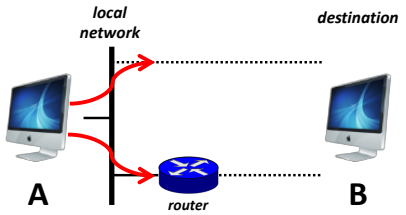
Internet Forwarding

Dr Andrew Scott

a.scott@lancaster.ac.uk

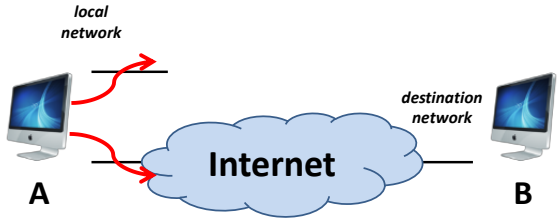
Getting from A to B

- Given a packet destined for host B
  - Is B on same physical link? *…send direct*
  - Is B on a remote network? *…send to router*



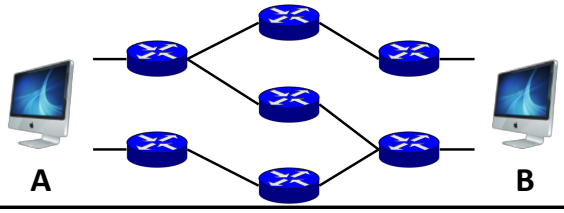
More Generally…

- PC A may have many interfaces, thus options
- If we send to a gateway router… what next?
- How should packet be *forwarded* toward B?



Getting from A to B

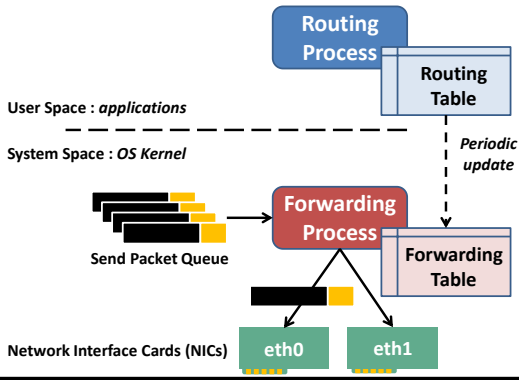
- Internet formed from a set of routers
  - No global view (at IP level)
- Routers conspire to deliver packets to destination
  - Each pushing packets closer to their destination



Internet 'Routing'

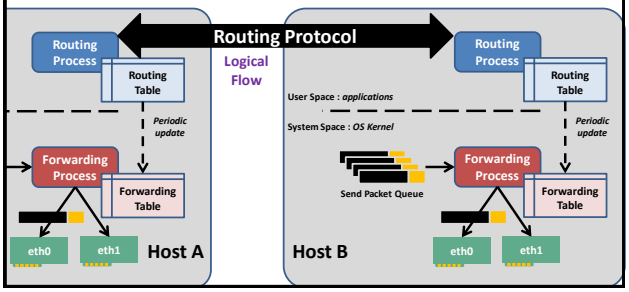
- Two <u>distinct</u> parts to process
  - Routing
    - **Application level process** to determine routes
  - Forwarding
    - **System level process** directing packets according to learned routes

Routing and Forwarding



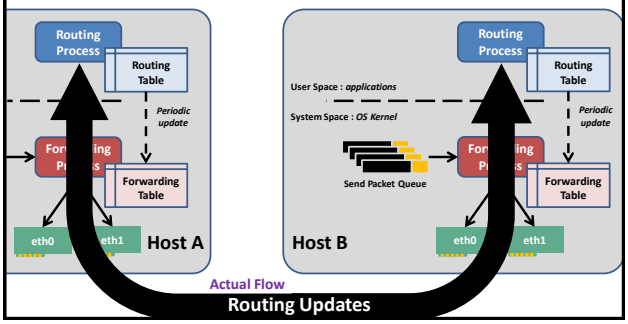
Routing and Forwarding

- Routing processes exchange route updates
  - Details later inform forwarding decisions



Beware: Updates treated as any other data

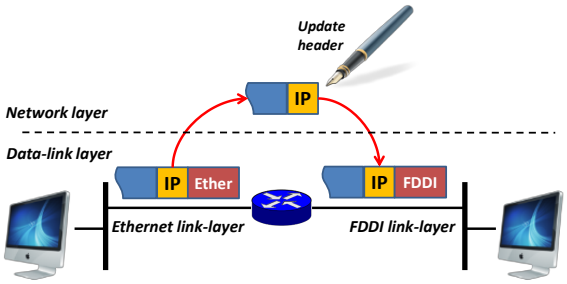
- Errors in forwarding tables can block updates



IP Forwarding

- Kernel function operating at *Network layer*
- Involves
  - Selecting best of known *path* to destination
  - Updating IP header
- Independent of *Link-layer*
  - Can forward between different types of network

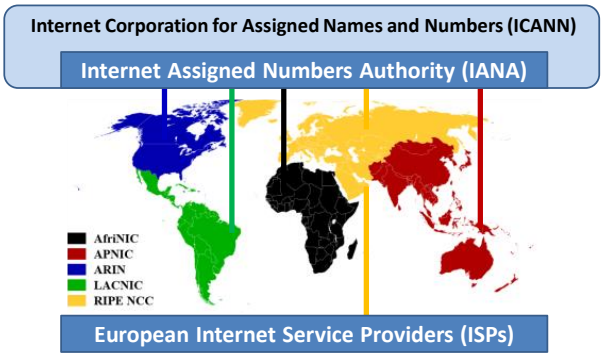
IP Forwarding



IPv4 Address Format

- 32 bit address
- Typically represented in dotted-quad form
  - 194.80.37.5
  - Each part represents 8 bit (byte), thus 0 – 255

Internet Address Allocation



IP Addresses

- Hierarchical
  - Not flat like Ethernet/ MAC address
- Topological
  - Reflects network structure
  - Not geographic
    - Though topology might be constrained by geography

Traditional IPv4 Allocation

- Organisations allocated address blocks
  - Contiguous set of addresses based on org. size

	8 bits	8 bits	8 bits	8 bits
Class A	Network	Host		
Class B	Network		Host	
Class C	Network			Host
Class D - multicast				
Class E - experimental				

Must be able to distinguish class by content of first byte…

Class Bit-prefixes

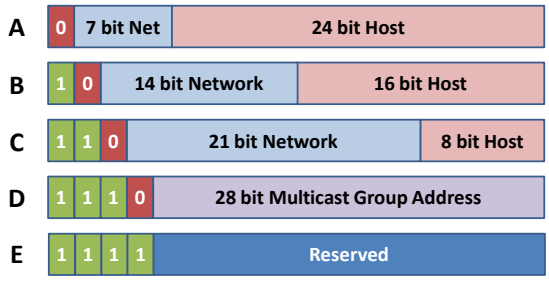
- First four bits give indication of class

Class	Bit prefix	First byte
A	0	1 – 126
B	10	128 – 191
C	110	192 – 223
D	1110	224 - 239
E	1111	240 - 255

0 and 127 are reserved, e.g.,

Address **127.0.0.1** is the ***loopback interface*** used for host internal traffic: data is 'passed down stack' and received by same host as though it had come across network

Usable Network and Host Bits



Available Networks

- Class A network addresses must start with 0
  - This halves range (from 8bits to 7)
  - Networks 0 and 127 are reserved giving range -2
    - So range is 1 to 126, *…use similar method for Class B + C*

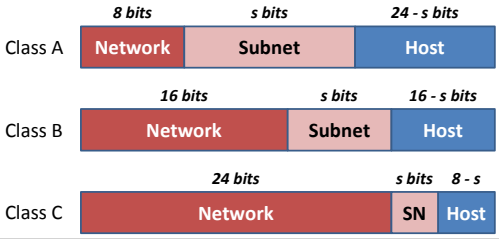
	8 bits	8 bits	8 bits	8 bits	Number of Networks
Class A	1 – 126	1 – 16,777,214			$2^7 - 2 =$ **126**
Class B	128 – 191	…	1 – 65,5354		$2^{6+8} = 2^{14} =$ **16,192**
Class C	192 – 223	…		1 - 254	$2^{5+16} = 2^{21} =$ **2,097,152**
Class D	224 – 239	1 – 16,777,214			
Class E	240 – 255	1 – 16,777,214			**Total networks: 2,113,470**

IP Addresses

- Network addresses allocated to organisations by their *Internet Service Provider* (ISP)
  - Used to correctly forward traffic – *how we know destination*
  - Fixed and cannot be changed by organisation
- Host addresses 'belong' to organisation
  - Allocate/ change them at will

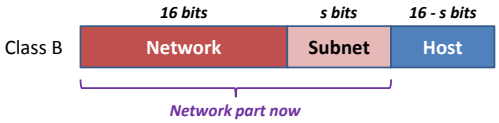
Internal IP Sub-netting

- Within organisation
  - Borrow host bits to form internal (sub-)networks
  - Network part is fixed – *what ISP allocated to org.*



Sub-netting

- Notice we can no longer determine network-host boundary by knowing address class
  - Subnet address only makes sense within organisation
- ISP will still treat this as 16bit network address
  - The network address it allocated to organisation

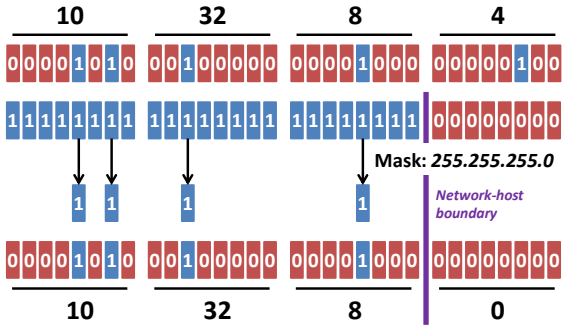


Variable Length Subnet Masking (VLSM)

- *Subnet mask* is used to split (sub-)net and host parts
- Mask must have set of ones followed by set of zeros
  - Once a zero bit appears, all remaining bits must be zero

- Any IP address **AND**'d with its subnet mask gives network address on which it resides

Subnet Masks



Subnet Masks

	Network			Host
Address	192	168	88	32
Subnet mask	255	255	255	0
Result	192	168	88	0

	Network		Host	
Address	192	168	88	32
Subnet mask	255	255	0	0
Result	192	168	0	0

Masks and Subnets

		128	64	32	16	8	4	2	1
Address	194.80.37.	0	0	0	0	0	0	0	0
Mask	255.255.255.	0							

255.255.255.0	← *Subnet mask*
194.80.37.**0**	← **Single available subnet**

Masks and Subnets

		128	64	32	16	8	4	2	1
Address	194.80.37.	X	0	0	0	0	0	0	0
Mask	255.255.255.	128	*Host bits*						

255.255.255.0	255.255.255.128	← *Subnet mask*
194.80.37.**0**	194.80.37.**0**	← **128 bit unset in address**
	194.80.37.**128**	← **128 bit set in address**

Masks and Subnets

		128	64	32	16	8	4	2	1
Address	194.80.37.	X	X	0	0	0	0	0	0
Mask	255.255.255.	192	*Host bits*						

255.255.255.0	255.255.255.128	255.255.255.192	← *Subnet mask*
194.80.37.**0**	194.80.37.**0**	194.80.37.**0**	
		194.80.37.**64**	
	194.80.37.**128**	194.80.37.**128**	
		194.80.37.**192**	**(128 + 64)**

Masks and Subnets

		128	64	32	16	8	4	2	1
Address	194.80.37.	X	X	X	0	0	0	0	0
Mask	255.255.255.	224	*Host bits*						

255.255.255.0	255.255.255.128	255.255.255.192	255.255.255.224
194.80.37.**0**	194.80.37.**0**	194.80.37.**0**	194.80.37.**0**
			194.80.37.**32**
		194.80.37.**64**	194.80.37.**64**
			194.80.37.**96**
	194.80.37.**128**	194.80.37.**128**	194.80.37.**128**
			194.80.37.**160**
		194.80.37.**192**	194.80.37.**192**
			194.80.37.**224**

Host Addresses per Subnet

- For a subnet of *n* bits
  - Number of available host addresses is $2^n - 2$
- Why minus 2?

  Two reserved addresses per subnet :
  - All zeros is reserved *(it's the network number)*
  - All ones is subnet broadcast address

Host Addresses per Subnet, *example*

- Why minus 2?

  Two reserved addresses per subnet :
  - All zeros is reserved *(it's the network number)*
  - All ones is subnet broadcast address

Lancaster University Class B: 148.88.0.0	
Subnet mask	255.255.0.0
Network address	148.88.0.0
Subnet Broadcast address	148.88.255.255

Subnet Address Ranges

		128	64	32	16	8	4	2	1
Mask	255.255.255.	1	1	1	0	0	0	0	0
	255.255.255.	224							
Subnet	194.80.37.	0	0	0	0	0	0	0	0
	194.80.37.	0							
Broadcast	194.80.37.	31			1	1	1	1	1
		(0 + 31 = 31)			16 + 8 + 4 + 2 + 1 = 31				

Subnet	Lowest IP	Highest IP	Broadcast
194.80.37.0	194.80.37.**1**	194.80.37.**30**	194.80.37.**31**
	Subnet + 1	***Broadcast - 1***	

Subnet Address Ranges

		128	64	32	16	8	4	2	1
Mask	255.255.255.	1	1	1	0	0	0	0	0
	255.255.255.	224							
Subnet	194.80.37.	0	0	1	0	0	0	0	0
	194.80.37.	32							
Broadcast	194.80.37.	63			1	1	1	1	1
		(32 + 31 = 63)			16 + 8 + 4 + 2 + 1 = 31				

Subnet	Lowest IP	Highest IP	Broadcast
194.80.37.0	194.80.37.**1**	194.80.37.**30**	194.80.37.**31**
194.80.37.32	194.80.37.**33**	194.80.37.**62**	194.80.37.**63**
	Subnet + 1	***Broadcast - 1***	

Subnet Address Ranges

		128	64	32	16	8	4	2	1
Address	194.80.37.	1	1	1	0	0	0	0	0
Mask	255.255.255.	224							

Subnet	Lowest IP	Highest IP	Broadcast
194.80.37.0	194.80.37.**1**	194.80.37.**30**	194.80.37.**31**
194.80.37.32	194.80.37.**33**	194.80.37.**62**	194.80.37.**63**
194.80.37.64	194.80.37.**65**	194.80.37.**94**	194.80.37.**95**
194.80.37.96	194.80.37.**97**	194.80.37.**126**	194.80.37.**127**
194.80.37.128	194.80.37.**129**	194.80.37.**158**	194.80.37.**159**
194.80.37.160	194.80.37.**161**	194.80.37.**190**	194.80.37.**191**
194.80.37.192	194.80.37.**193**	194.80.37.**222**	194.80.37.**223**
194.80.37.224	194.80.37.**225**	194.80.37.**254**	194.80.37.**255**

Going the opposite way: *Super-netting*

- Grouping of multiple contiguous address ranges
- If allocated adjacent Class C blocks
  - Organisation could 'merge' them
    - Simplifies management of allocated addresses
    - Allows more flexibility when structuring internal subnets
- ISP will continue to treat them as independent
  - A super-netted address MUST NOT span network addresses not subject to same routing path

Super-netting

- 'Combine' adjacent subnet addresses
  - Locally, mask pushed into network part of address
  - Treat lower bits of network address as host bits

Extend host part up by (in this case) 1 bit ←

		128	64	32	16	8	4	2	Host...	
Net 1	194.80.	36								0
		0	0	1	0	0	1	0	0	
Net 2	194.80.	37								0
		0	0	1	0	0	1	0	1	
Mask	255.255.	255 254								0
		1	1	1	1	1	1	1		

Classless Inter-Domain Routing (CIDR)

- Two main aims
  - Reduce number of routing table entries
  - Slow IPv4 address depletion *(Note: now all allocated)*
- Logical step after Variable Length Subnet Masks
  - Allowed registries to allocate on any bit boundary
  - Can better reflect size of requesting organisation
- Introduced the */prefix-length* notation
  - Standard notation for IPv6

address / prefix-length

Alternative *Prefix* Notation

		128	64	32	16	8	4	2	1
Address	194.80.37.	1	1	1	0	0	0	0	0
Mask	255.255.255.	224							

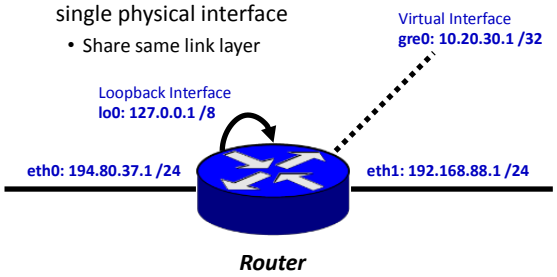
3 bytes, 24 bits — *+3 = 27 bits*

Address: 194.80.37.0 **Prefix:** 194.80.37.0 **/ 27**
Mask: 255.255.255.224

		128 + 64 = 192	128 + 64 + 32 = 224
255.255.255.0	255.255.255.128	255.255.255.192	255.255.255.224
/24	/25	/26	/27

Addresses are bound to Interfaces

- Addresses belong to interfaces not machines
  - Note: Can have multiple *virtual interfaces* on single physical interface
    - Share same link layer



Router

Configuring Addresses

- Each network interface needs to be configured with basic parameters:

eth0 :

Parameter	Value			
Network	194	80	37	0
Subnet Mask	255	255	255	0
Subnet Broadcast Address	194	80	37	255

Note that we only need IP address and Subnet mask/ prefix to work the others out

One Hop at a Time

- IP depends on underlying data-link protocol
- Data-link protocol can only address devices on same physical link/ network segment
- IP header holds endpoint addresses
  - Original source and final destination
- Data-link frames 'holding' packet sent hop-by-hop
  - Always sent to next-hop router, *until last subnet*

Data sent on wire

Forwarding Table

- Each device needs a forwarding table

This is one of the device's network interfaces
- can be interface name or address

Net. Address/ Prefix	Subnet Mask	Next Hop Router	Interface	Metric ('cost')
194.80.37.0	**255.255.255.0**	**192.168.88.1**	**eth0**	**0**

192.168.88.1 is router that can get our packets to network 194.80.37.0

This router *MUST* be on same (virtual) link/ subnet as eth0

Forwarding Table

- Many nodes have a ***default route*** prefix: 0.0.0.0 /0
  - Route of last resort, when no other route available
  - Default route typically toward our ISP's router

Net. Address/ Prefix	Subnet Mask	Next Hop Router	Interface	Metric ('cost')
0.0.0.0	**0.0.0.0**	**148.88.8.1**	**eth1**	**0**
194.80.37.0	**255.255.255.0**	**192.168.88.1**	**eth0**	**0**

Checking Host Table

U – route 'up'
G – gateway route
H – host route

```
root@acs:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         194.80.37.1     0.0.0.0         UG    0      0        0 eth0
link-local      *               255.255.0.0     U     1000   0        0 eth0
194.80.37.0     *               255.255.255.128 U     1      0        0 eth0
root@acs:~#
```

```
C:\> route print
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0      10.20.30.1    10.20.30.72      25
       10.20.30.0  255.255.255.128         On-link    10.20.30.72     281
       10.20.30.72 255.255.255.255         On-link    10.20.30.72     281
      10.20.30.127 255.255.255.255         On-link    10.20.30.72     281
        127.0.0.0        255.0.0.0         On-link     127.0.0.1     306
        127.0.0.1 255.255.255.255         On-link     127.0.0.1     306
  127.255.255.255 255.255.255.255         On-link     127.0.0.1     306
        224.0.0.0        240.0.0.0         On-link     127.0.0.1     306
        224.0.0.0        240.0.0.0         On-link    10.20.30.72     281
```

Determining Next Hop

- Given destination address 10.20.30.5 …

Network Address/ Prefix	Subnet Mask	Next Hop Router	Interface
194.80.37.0	255.255.255.0	192.168.88.1	eth0
10.20.30.0	255.255.255.0	192.168.88.2	eth0

- Test against each entry … *Bitwise AND*

Destination Address	& Subnet Mask	Prefix to test	= Prefix ?
10.20.30.5	10.20.30.0	194.80.37.0	No

Determining Next Hop

- Given destination address 10.20.30.5 …

Network Address/ Prefix	Subnet Mask	Next Hop Router	Interface
194.80.37.0	255.255.255.0	192.168.88.1	eth0
10.20.30.0	255.255.255.0	192.168.88.2	eth0

- Test against each entry … *Bitwise AND*

Destination Address	& Subnet Mask	Prefix to test	= Prefix ?
10.20.30.5	10.20.30.0	194.80.37.0	No
10.20.30.5	10.20.30.0	10.20.30.0	Yes

Send packet for 10.20.30.5 to next hop 192.168.88.2 via interface eth0

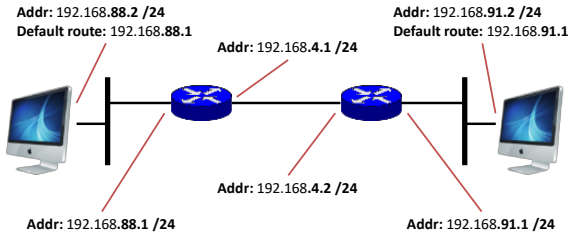
Internet Forwarding *continued*

Dr Andrew Scott

a.scott@lancaster.ac.uk

Forming a Network

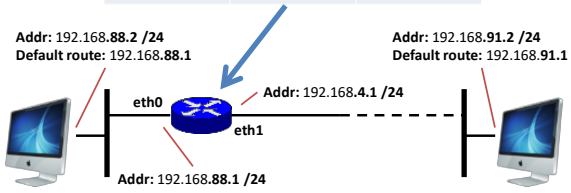
- Each interface has an address
- Default route is to router's interface on local subnet
- Don't typically have default routes within network



Try sending from 192.168.**88.2** to 192.168.**91.2**

Router's Forwarding Table

Prefix	Out Interface	Next Hop
192.168.88.0 /24	eth0	*link-local*
192.168.4.0 /24	eth1	*link-local*



*Sent to MAC addr. of 192.168.88.1 **

Src : 192.168.**88.2**
Dst : 192.168.**91.2**

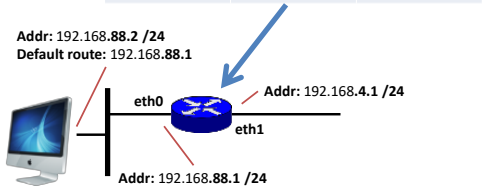
What happens?

* Can only go one hop/ link at a time

Error Returned

Router's Forwarding Table

Prefix	Out Interface	Next Hop
192.168.88.0 /24	eth0	*link-local*
192.168.4.0 /24	eth1	*link-local*



*Sent to MAC addr. of 192.168.88.1 **

Src : 192.168.**88.2**
Dst : 192.168.**91.2**

Router has no route to network

Src : 192.168.**88.1**
Dst : 192.168.**88.2**

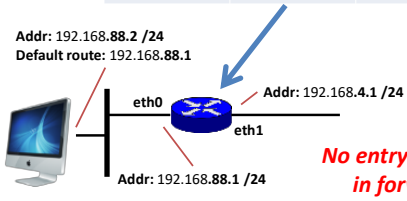
ICMP
Net. Unreachable

* Can only go one hop/ link at a time

Why did that fail?

Router's Forwarding Table

Prefix	Out Interface	Next Hop
192.168.88.0 /24	eth0	*link-local*
192.168.4.0 /24	eth1	*link-local*



No entry for 192.168.91.2 in forwarding table

*Sent to MAC addr. of 192.168.88.1 **

Src : 192.168.**88.2**
Dst : 192.168.**91.2**

Router has no route to network

Src : 192.168.**88.1**
Dst : 192.168.**88.2**

ICMP
Net. Unreachable

* Can only go one hop/ link at a time

Background

Updating Routing Table: Linux

```
root@as:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         194.80.37.1     0.0.0.0         UG    0      0        0 eth0
link-local      *               255.255.0.0     U     1000   0        0 eth0
194.80.37.0     *               255.255.255.128 U     1      0        0 eth0
root@as:~#
root@as:~# route add -net 11.22.33.0 netmask 255.255.255.0 gw 194.80.37.2
root@as:~#
root@as:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         194.80.37.1     0.0.0.0         UG    0      0        0 eth0
11.22.33.0      194.80.37.2     255.255.255.0   U     0      0        0 eth0
link-local      *               255.255.0.0     U     1000   0        0 eth0
194.80.37.0     *               255.255.255.128 U     1      0        0 eth0
root@as:~#
root@as:~# route del -net 11.22.33.0 netmask 255.255.255.0
```

Background: Updating Routing Table: Windows

```
C:\> route print
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0      10.20.30.1      10.20.30.72      25
       10.20.30.0  255.255.255.128         On-link      10.20.30.72     281
      10.20.30.72  255.255.255.255         On-link      10.20.30.72     281
     10.20.30.127  255.255.255.255         On-link      10.20.30.72     281
        127.0.0.0        255.0.0.0         On-link       127.0.0.1     306

C:\> route add 11.22.33.0 mask 255.255.255.0 10.30.201.127
 OK!
C:\> route print
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0      10.20.30.1      10.20.30.72      25
       10.20.30.0  255.255.255.128         On-link      10.20.30.72     281
      10.20.30.72  255.255.255.255         On-link      10.20.30.72     281
     10.20.30.127  255.255.255.255         On-link      10.20.30.72     281
       11.22.33.0    255.255.255.0         On-link      10.20.30.72      26
     11.22.33.255  255.255.255.255         On-link      10.20.30.72     281
        127.0.0.0        255.0.0.0         On-link       127.0.0.1     306

C:\> route delete 11.22.33.0
 OK!
```

Multiple Next Hop Matches

- Likely to be more than one matching entry
  - If only due to default route
    - Which always matches
- **Always select longest matching prefix**
  - Most specific entry
  - Should be best path to destination

Longest Prefix Match

- Apply mask to address and compare with prefix

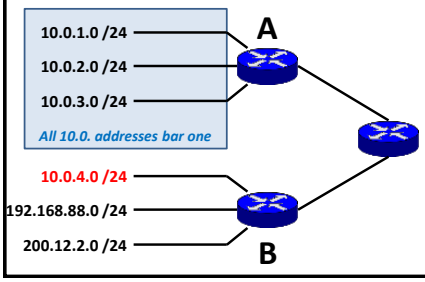
Destination Address =	10	32	5	9	
Prefix				**Mask**				**Masked Address**				
10	32	0	0	255	255	0	0	10	32	0	0	✔
10	32	5	0	255	255	255	0	10	32	5	0	✔
192	168	88	6	255	255	255	0	10	32	5	0	✘

= ? = Prefix ?

- Options are:
  - 10.32. shortest of two
  - 10.32.5. longest of two ← *we select this*

Why Multiple Matches?

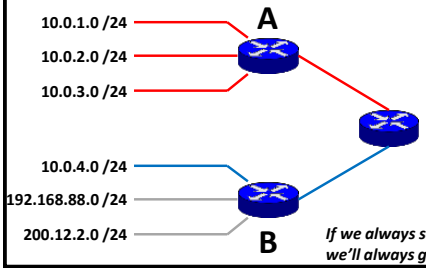
- Two equivalent scenarios :
  1. Fully enumerated table



Network Address	N.H.
10.0.1.0 /24	A
10.0.2.0 /24	A
10.0.3.0 /24	A
10.0.4.0 /24	B
192.168.88.0 /24	B
200.12.2.0 /24	B

Why Multiple Matches?

- Two equivalent scenarios :
  2. 10.0.0.0/16 with **exception** for 10.0.4.0/24



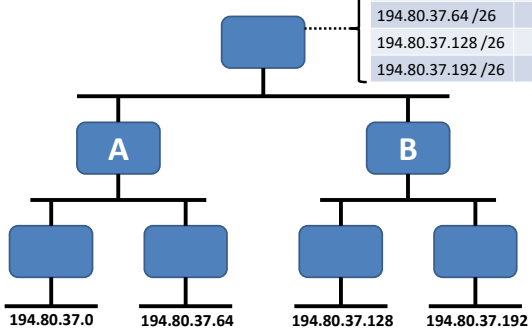
Simpler routing table
(fewer to check = faster)

Network Address	N.H.
10.0.0.0 /16	A
10.0.4.0 /24	B
192.168.88.0 /24	B
200.12.2.0 /24	B

If we always select Longest Prefix Match
we'll always get to correct destination

Building Tables

Forwarding Table	
Subnet	**Nxt Hop**
194.80.37.0 /26	A
194.80.37.64 /26	A
194.80.37.128 /26	B
194.80.37.192 /26	B



Route Summarisation

- Looking at that last example:

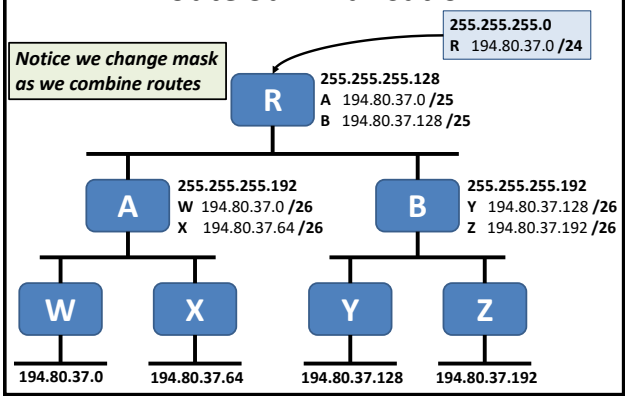
Initial Forwarding Table	
Subnet	Nxt Hop
194.80.37.0	A
194.80.37.64	A
194.80.37.128	B
194.80.37.192	B

- We can combine these routes
  - Two routes differ only in last bit
    - 194.80.37.0 + 194.80.37.64
    - 194.80.38.128 + 194.80.37.192
  - Each pair has the same next hop

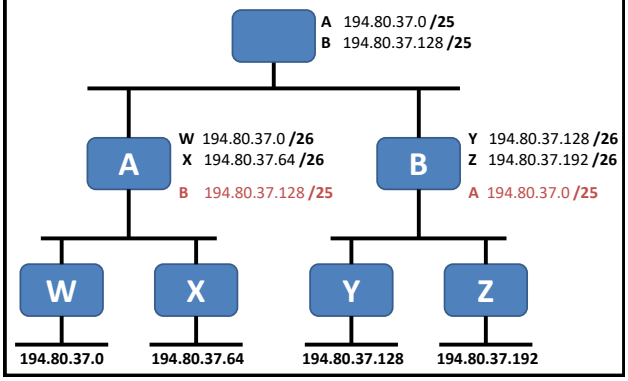
New Forwarding Table	
Subnet	Nxt Hop
194.80.37.0	A
194.80.37.128	B

** aka. Route Aggregation*
– though this is really a specific wide area mechanism

Route Summarisation

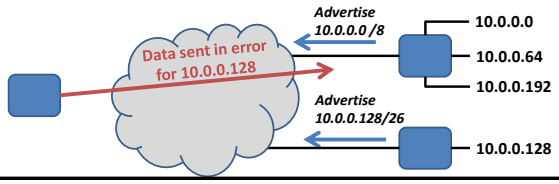


Completing the tables



Warning

- If a summary route covers 'holes' in address range
  - Router must *black-hole* any traffic erroneously sent to it
    - i.e. if a packet arrives for a 'host' somewhere in the hole
  - It MUST NOT return traffic and risk a loop



Reducing Table Sizes

- Size reduced by
  - Classless Inter-Domain Routing (CIDR)
    - Variable length subnet masks
    - Route summarisation allows main route + exceptions
  - Default routes
    - Can collapse all but special cases into single route entry
- Core network is a *'default free zone'*
  - Default pushes traffic toward core network
  - In core there is no default (central point/ root)

Handling Broadcasts

- Two forms of IPv4 broadcast address
  - 255.255.255.255 link-local broadcast
    - Must never be forwarded → *NO GLOBAL BROADCAST*
  - 194.80.37.**255** subnet (aka directed) broadcast address
    - More generally : <network prefix><**all 1s**>
    - Forwarded as normal packet until subnet
      - then broadcast on physical links forming subnet
- Broadcast packet MUST never be forwarded out of interface on which it arrived – *to avoid loops*

IP Delivery: *Summary*

- Direct delivery -- *same link*
  - Use ARP to obtain MAC address for destination *
  - Send over connected link to host's MAC address
- Remote delivery -- *off link*
  - Find next hop in table using longest prefix match
  - Use ARP to obtain MAC address of router *
  - Send over connected link to router's MAC address

* If same ARP translation recently done, mapping should already be in ARP cache

IP Delivery: Summary

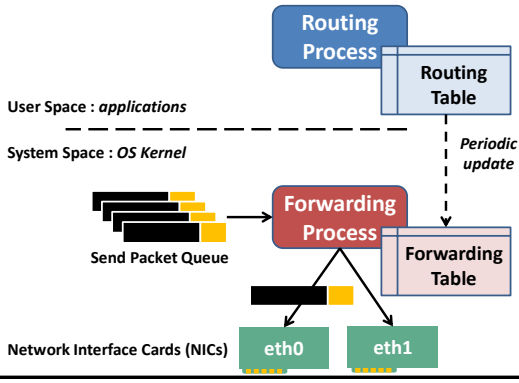
- Packet sent hop-by-hop
  - Remote delivery until router sees destination on same link
  - Then uses direct delivery to final destination
- Remember a router can be final destination
  - Remote console or other management connection

Route Information Protocol (RIP)

Dr Andrew Scott

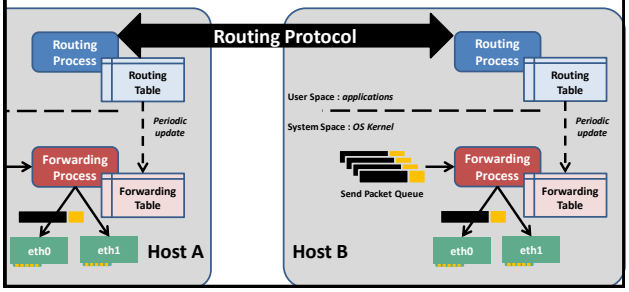
a.scott@lancaster.ac.uk

Routing and Forwarding



Routing and Forwarding

- Routing processes exchange route updates
  - Details later inform forwarding decisions



Configuring Routes

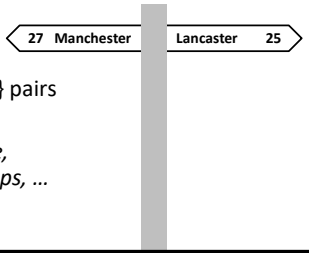
- Two approaches
  - Non-Adaptive/ Static
    - Always used to add local (to device) interfaces to table
    - Manually configure any additional routes
    - Relies on administrator establishing consistent routes …on devices across networks under their control
  - Adaptive/ Dynamic
    - Routers run distributed algorithm
    - Aim to establish best path between any two points
    - Depends on ***Routing Protocols***

Routing Protocols

- Local Area Networks *Interior Routing*
  - Distance Vector
  - Link State
- Wide Area Networks *Exterior Routing*
  - Path Vector

Distance Vector

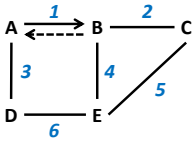
- Implement the Bellman-Ford algorithm
  - A distributed shortest-path algorithm
- Routers exchange
  - {Destination, Metric} pairs
  - Metric: *cost, distance, number of hops, …*



AN EXAMPLE NETWORK…

Distance Vector

- Graph with
  - 5 ***nodes*** A – E, each with a table of distance vectors
  - 6 ***edges***/ links 1 – 6, each with a known traversal cost

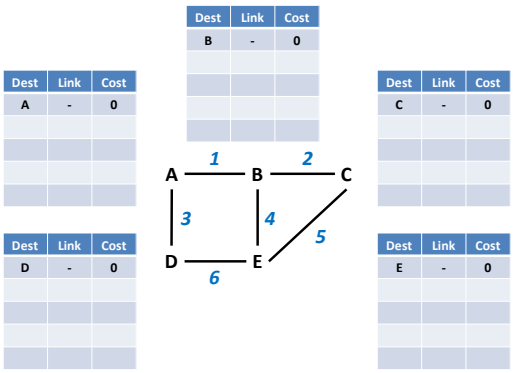


Dest	Link	Cost

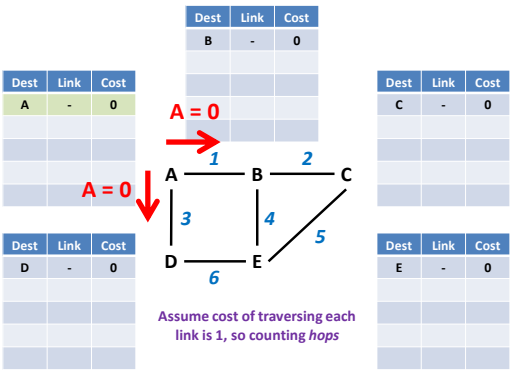
Strictly, traversal cost can be different in each direction
- should be two edges between nodes, as seen with A – B

Tend to assume bidirectional links with same cost in both directions

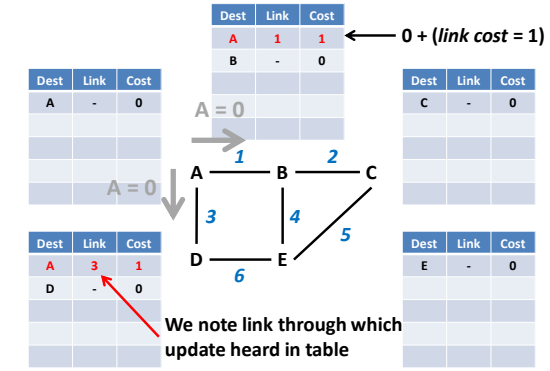
Distance Vector: *Initial State*



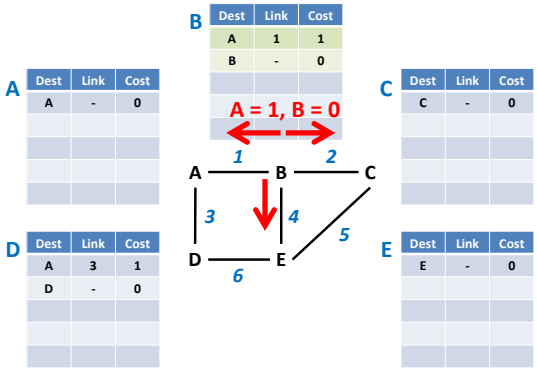
Nodes Broadcast Distance Vectors



Nodes Add Link Cost and Update Table



Other Nodes Broadcast Known Vectors



Add Link Costs and Update Tables

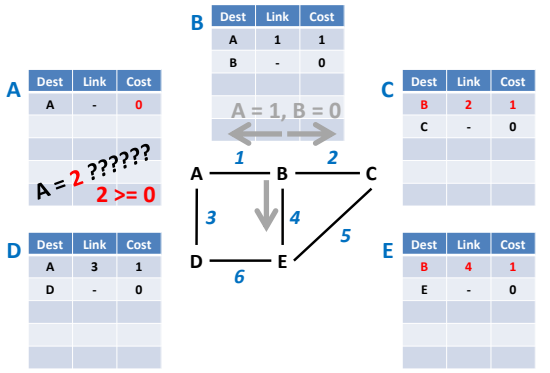
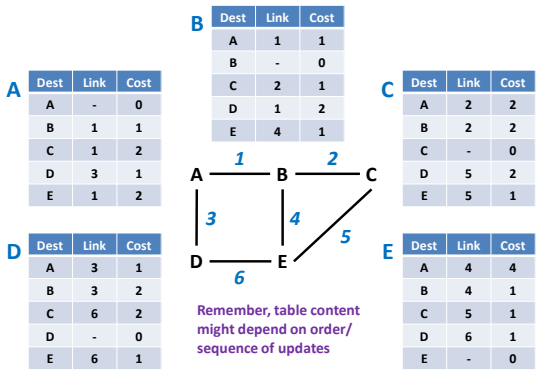


Table Updates

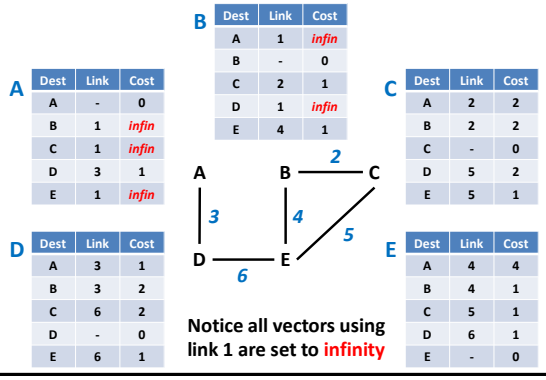
- Didn't update table A because (received + link) cost to A **>=** that already in table
- Due to that *equals*
  - Table content may depend on order of updates
  - So, path from ***X to Y*** may not be same as ***Y to X***
    - We would call this an *asymmetric* route

System Stabilises



WHEN THINGS GO WRONG…

Link 1 Fails: *infinite cost*

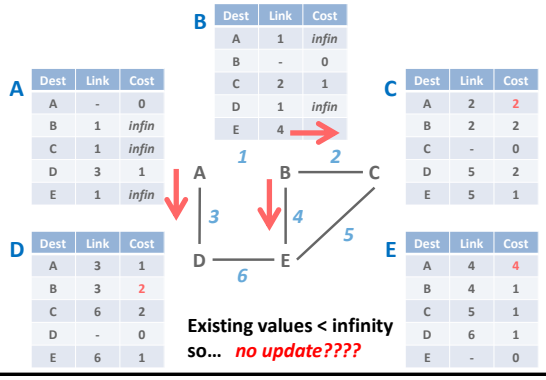


Notice all vectors using link 1 are set to infinity

A and B Immediately Send Updates

Notice all vectors using link 1 are set to infinity

A and B Immediately Send Updates



Existing values < infinity so… *no update????*

Dealing With New Information

- Normally ignore updates for vectors where we have existing entry <= (*received value* + *link cost*)
- We've received an **increased value** **via recorded link**
  - For example,
    update (from B to E) via link 4 says cost to A is infinite

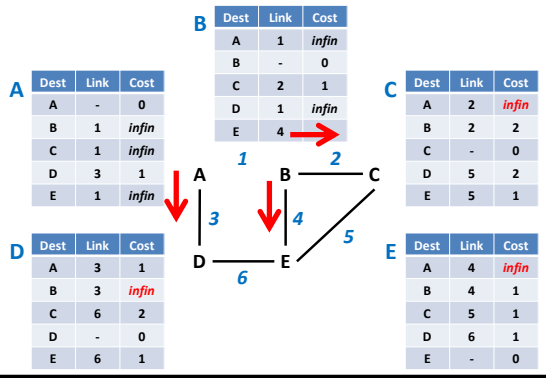
    E's table says learned cost to A via link 4

    → So we **MUST** update cost to A

Dest	Link	Cost
A	4	4
B	4	1
C	5	1
D	6	1
E	-	0

Note, this should be case for any increased cost, not just infinity due to link failures

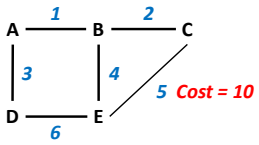
Propagating Link Failure Information



2 Update Cycles Insert Alternate Links

Links With Higher Cost

- Link cost could be a good way of reflecting
  - Financial cost of using link
  - Relative bandwidth/ capacity of link

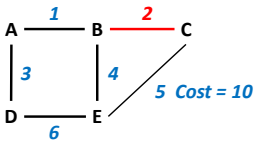


Lets see costs and how information would be forwarded…
- assume all other links continue to have cost = 1

Consider Routes to C

- *This table is a convenience*
  - *Remember each node keeps its own local table/ view as we saw earlier*

To C From	Link	Cost
A	1	2
B	2	1
C	-	0
D	3	3
E	4	2

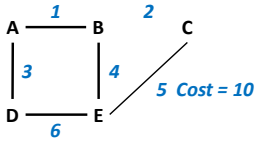


But what if link 2 fails…?

Link 2 Fails

- Two things will happen
  - Normal periodic updates
  - Failure triggers B to send update early

To C From	Link	Cost
A	1	2
B	2	*infinity*
C	-	0
D	3	3
E	4	2

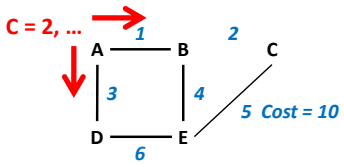


But, let's say A happens to do a periodic update first…

Assume A Sends *C = 2, …* first

- On receipt
  - D sees: C = 2 + 1 = 3 *…no change*
  - B sees: C = 2 + 1 = **3** ***< infinity***

To C From	Link	Cost
A	1	2
B	2	*infinity*
C	-	0
D	3	3
E	4	2

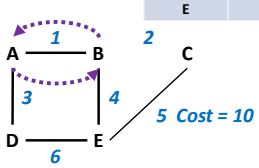


So B updates table…

Errrr…

- A still believes C is via B (link 1)
- B now believes C is via A

To C From	Link	Cost
A	1	2
B	1	3
C	-	0
D	3	3
E	4	2

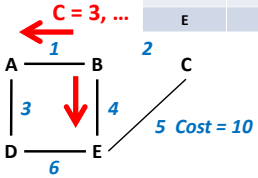


Now B send its update…

Now B Sends Updates

- A and E see increase from B
  - A and E update their tables

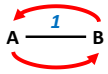
To C From	Link	Cost
A	1	4
B	1	3
C	-	0
D	3	3
E	4	4



But What About Link 5?

- A believes C is via B
- B believes C is via A
- All costs are less than 10 so link 5 cannot be used
- Note we have a nasty routing loop to C
  - Any packet sent will cycle until its TTL hits 0 and expires

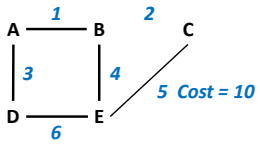
To C From	Link	Cost
A	1	4
B	1	3
C	-	0
D	3	3
E	4	4



Eventually Get Something Like…

- Remember
  - Table content depends on order of received updates

To C From	Link	Cost
A	1	12
B	1	11
C	-	0
D	3	11
E	4	12



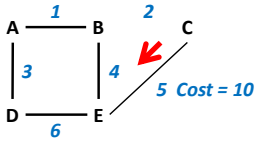
It takes 5 full rounds of updates to get to this point!

…and there's still a loop!

When C Sends Next Update…

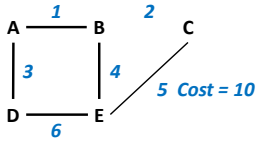
- C sends C = 0 to E
  0 + 10 = 10, which < 12
- E can now use link 5 to C

To C From	Link	Cost
A	1	12
B	1	11
C	-	0
D	3	11
E	4	12



To Give…

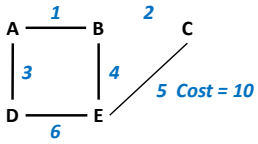
To C From	Link	Cost
A	1	12
B	1	11
C	-	0
D	3	11
E	5	10



Next round of updates cause other nodes to converge to use link E - C

We Have a Working System Again

To C From	Link	Cost
A	1	12
B	4	11
C	-	0
D	6	11
E	5	10



The 'Bouncing Effect'

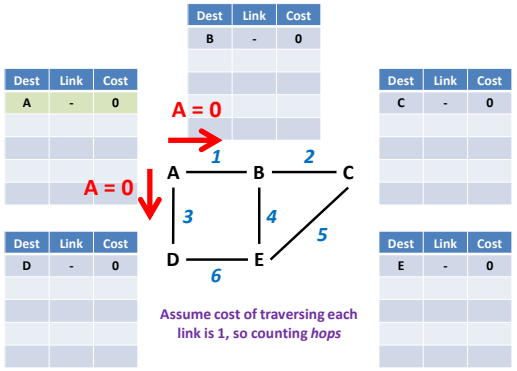
- Each full round adds twice link cost to related entries
  - For example, A → B → A
    A sends cost *n* to B, B adds link cost *c* to give $n + c$
    B then sends this back to A, which adds *c* giving $n = n + 2c$
- Looping packets can cause congestion
  - This can cause packet loss
  - Can easily end up discarding the routing packets needed to fix problem

Route Information Protocol (RIP) continued

Dr Andrew Scott

a.scott@lancaster.ac.uk

Recap: *Distance Vector* protocols

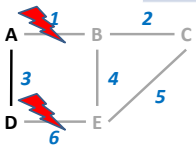


WHAT IF TWO LINKS FAIL?

Two Link Failure: from *D's perspective*

- Let's focus on A and D
  - B, C, and E will do their own thing for a while

From D to	Link	Cost
D	local	0
A	3	1
B	6	infinity
E	6	infinity
C	6	infinity



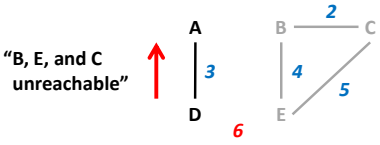
Remember, this combined table is a convenience – each router will maintain its own local routing table

TWO SCENARIOS…

Two Link Failure: *scenario 1*

- D **sees link failure**
  - Sends update to A:

From D to	Link	Cost
D	local	0
A	3	1
B	6	infinity
E	6	infinity
C	6	infinity

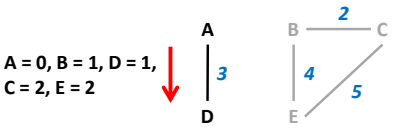


A updates table -- everything works ☺

Two Link Failure: *scenario 2*

- A sends update **before noticing link failure**

From D to	Link	Cost
D	local	0
A	3	1
B	3	2
E	3	3
C	3	3



D updates table and we have another loop, between A and D

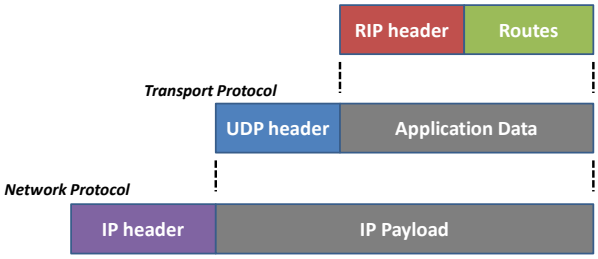
Count to Infinity

- Notice **there is no good information in system**
  - A and D are isolated
- Their entries increase by twice link cost with each full round of updates
- This continues until they reach infinity
  - Routes only then considered unreachable
  - So, we'd like *'infinity'* to be small

…but must be greater than maximum cost we'll ever see

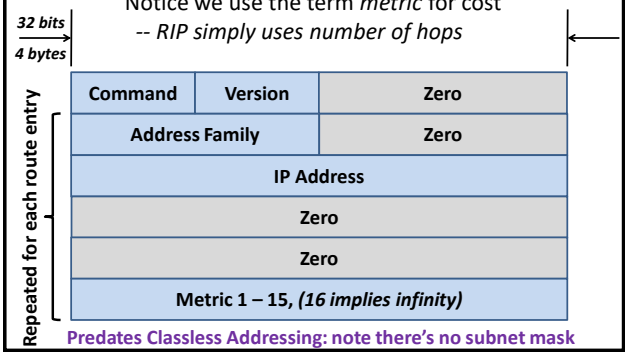
Route Information Protocol (RIP)

- Simple implementation of Distance Vector
  - Version 1 predates classless addressing



RIPv1 Route Announcement

Notice we use the term *metric* for cost
-- RIP simply uses number of hops



Predates Classless Addressing: note there's no subnet mask

RIP Updates

- Updates sent at regular intervals *~30 seconds*
- Each table entry has *time to live* (TTL)
  - If no refresh within this time route set to infinity
    …node considered unreachable
  - Recommended TTL = 6 x update interval *(180 seconds)*
    - *Need one in every 6 updates to get through*
  - Too long → stale entries
    + more likely to see *bouncing* and *count to infinity*
  - Too short → kill valid routes due to transient network problems

RIP

- Addresses *bouncing* and *count to infinity* by:
  - Triggered Updates
  - Split Horizon

Triggered Updates

- Nodes send updates early if change in their route table
  - Due to **Link failure** or **increased link cost**
- Should reduce chance of
  - Nodes holding stale information
  - *Bouncing* and *count to infinity*
    - Both were caused by nodes sending stale information
      …so get new information out quickly

Split Horizon

- Simple observation
  - If A has route to D via node B
    …B must never try to reach D via A
  - A must not tell B it can reach D (*if can only do so via B*)
    - Unless it finds a better route (not via B)
- Requires different updates on each adjacent link
  - Omit entries reached via link on which update sent

Poisoned Reverse

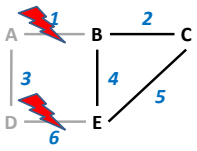
- Split Horizon with Poisoned Reverse
  - A more aggressive variant of Split Horizon
- Instead of omitting entries (back to source)
  - Send them, but list them with infinite cost

  i.e. explicitly announce them as unreachable
- Aim to quickly kill stale entries in adjacent devices
  - Works with one- and two-hop loops

Multi-hop Loops

- Now focus on B, C, and E
  - And the routes they hold

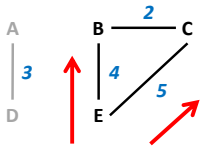
From	Link	Cost
B to D	4	2
C to D	5	2
E to D	6	1



E sees failure and sends updates

- Now focus on B, C, and E
  - And the routes they hold

From	Link	Cost
B to D	4	2
C to D	5	2
E to D	6	infinity

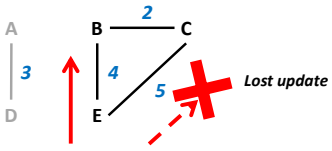


E to D unreachable
(cost infinity)

Assume that update to C lost

- B gets correct information
- C still holds stale route

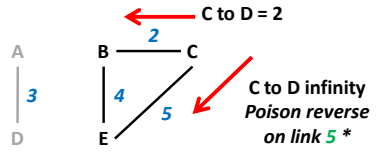
From	Link	Cost
B to D	4	infinity
C to D	5	2
E to D	6	infinity



E to D unreachable
(cost infinity)

C now does normal update

From	Link	Cost
B to D	4	infinity
C to D	5	2
E to D	6	infinity

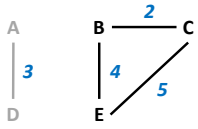


* We got route to D via link 5 so send infinity back

Now have another loop

- B believes C can reach D
- So we have…
  B → C → E → B → …

From	Link	Cost
B to D	2	3
C to D	5	2
E to D	4	4



Updates will count to infinity

RIPv1 Route Announcement

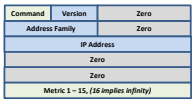
Notice we use the term *metric* for cost
-- *RIP simply uses number of hops*

32 bits →
4 bytes

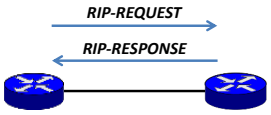
Command	Version	Zero
Address Family		Zero
IP Address		
Zero		
Zero		
Metric 1 – 15, *(16 implies infinity)*		

Repeated for each route entry

Solicited Updates

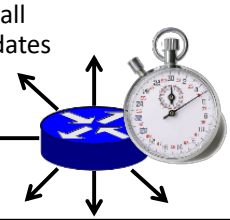


- Device sends *RIP-REQUEST*
  - Command 1 and metric set to 16
  - Version set appropriately for v1 or v2 reply
  - No other data
- Peer replies with one or more *RIP-RESPONSE* messages giving list of known routes



Un-solicited Updates

- Regularly broadcast *RIP-RESPONSE*
  - Also send on significant connectivity changes
    - Such as link failure
- Synchronisation a problem if all routers adopt exactly 30s updates
  - Can be surprising load on net
  - Add some variation
    - So typically send at 15 to 45s



RIPv1 Announcements

Note: this is a ***poisoned reverse***

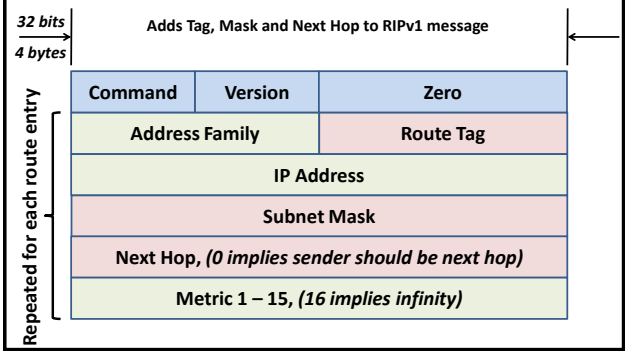
```
RIP command = 2, version = 1
AF_FAMILY: 2      IP_ADDRES: 10.20.101.00      Metric: 16
AF_FAMILY: 2      IP_ADDRES: 10.20.101.128     Metric: 1
AF_FAMILY: 2      IP_ADDRES: 148.88.00.00      Metric: 1
AF_FAMILY: 2      IP_ADDRES: 192.168.88.00     Metric: 1
AF_FAMILY: 2      IP_ADDRES: 194.80.36.00      Metric: 1
AF_FAMILY: 2      IP_ADDRES: 194.80.37.00      Metric: 1
```

A receiver should add one to the metrics (the link cost) and, if appropriate, update local routing table.

RIPv2

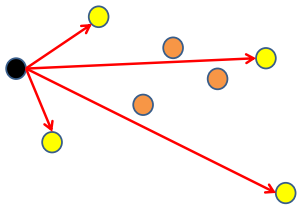
- Supports Classless Inter-Domain Routing (CIDR)
  - Exchanges subnet mask and next-hop information
  - Backward compatible format: *uses empty fields in RIPv1*
- Multicast based updates
  - Regular hosts not loaded handling broadcast updates
  - RIPv1 devices don't get updates they may misinterpret
- RIPv1 compatibility mode
  - Also broadcasts updates for any v1 nodes
  - Can accept RIPv1 updates

RIPv2 Route Announcement



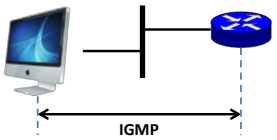
Multicast Based Updates

- Multicast
  - Deliver to defined group of hosts



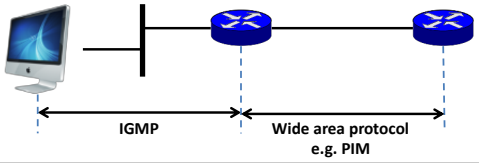
Internet Group Management Protocol

- Need a mechanism by which clients can register interest in groups/ data feeds
- Routers play key role in this
  - They're the gateway to remote off-subnet feeds



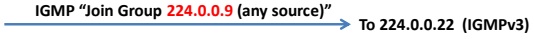
Internet Group Management Protocol

- IGMP used between device and first-hop router and between devices on same link
- Multicast protocols such as PIM used to control wide area group announcements and membership
  - PIM : Protocol Independent Multicast

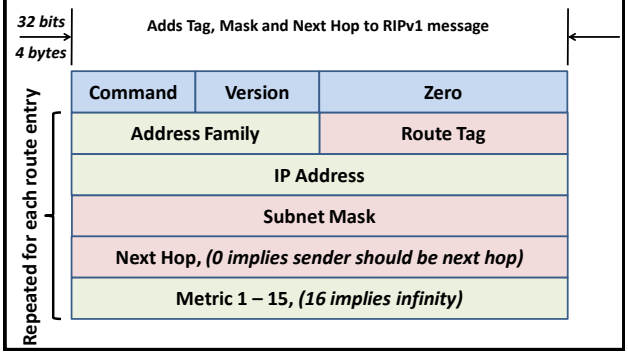


Clarification…

- Multicast aware devices join group 224.0.0.22 and listen for client join requests
  - Routers running RIPv2 do this
- To get RIP v2 data feed
  - Device needs to join group 224.0.0.9
- Sends join request to group 224.0.0.22
  - i.e. multicast management address



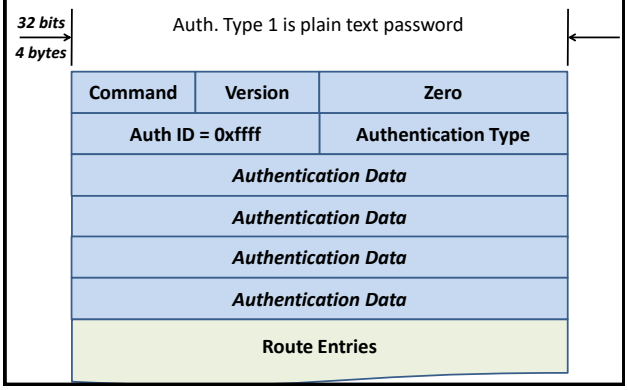
RIPv2 Route Announcement



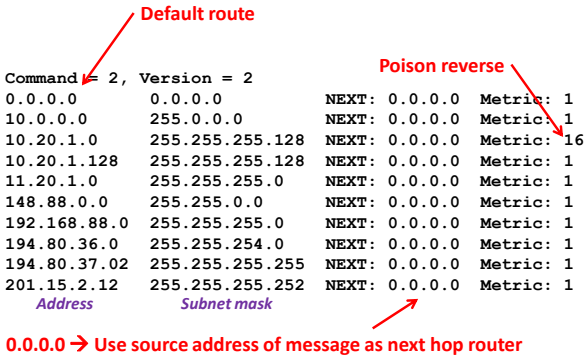
RIPv2: Why add next hop?

- RIPv1 used address of router that announced routes
- RIPv2 routers can announce next hops known to be better/ more direct than themselves
  - Can be useful in complex network structures
  - Not all nodes listen for every routing protocol in use

RIPv2 Authenticated Announcement



RIPv2 Sample Announcement

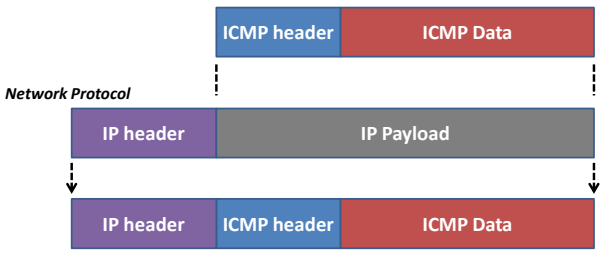


What if we use the wrong router?

- For example, if client only has a default route
  - May not be best next-hop router for destination
- Router
  1. Forwards packet
  2. Issues an ICMP Redirect to client
     - Informs client of better router for destination

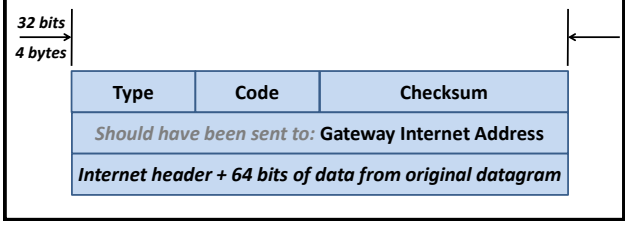
Internet Control Message Protocol

- Encapsulated in IP in similar manner to UDP
  - Note ICMP not considered a transport protocol



ICMP Type 5: Redirect

ICMP Type	Code	Description
5	**0**	Redirect datagrams for **network**
5	**1**	Redirect datagrams for **host**
5	**2**	Redirect datagrams for **type of service and network**
5	**3**	Redirect datagrams for **type of service and host**



Problems with RIP

- Rather simplistic, especially metric support
  - No support for multiple or dynamic metrics
    - Say, Hops + Link speed
- Robustness
  - Information (good or bad) freely distributed
    - Must trust neighbours – *doesn't work in wide area*
    - Only just adding support for encrypted updates

Problems with RIP

- Scalability
  - Doesn't scale to whole Internet
    - Use of 16 hops for infinity and slow convergence
    - Fixed update time and timeout prevent use on slow, high-loss nets
  - Limited to small scale/ local deployments
- Slow convergence
  - Updates sent node by node
    - Particular problem in large networks
  - Loops can (and likely will) occur during convergence

Problems with RIP

- Loop problems remain, despite
  - Triggered Updates
  - Split Horizon with Poisoned Reverse
- We only know cost to destination
  - Don't know the path → hard to make intelligent decisions
- Extensions proposed but add significant complexity
  - e.g. recursively check next hops to avoid loops

Reliable Transport Protocols

Dr Andrew Scott

a.scott@lancaster.ac.uk

UDP Not Designed to be Reliable

- 'Fire and forget' protocol
  - No direct response to any datagram sent
- Application must handle
  - Feedback between sender and receiver
  - Packet errors
  - Packet loss
- Ideally like to abstract this out of most applications

  → Reliable transport protocols

Reliable Transport

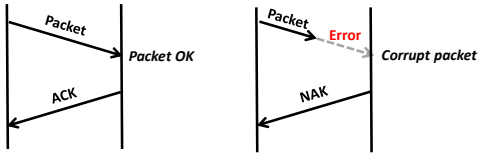
- IP offers unreliable, best-effort service
  - Reliable transp. protocol must handle possible errors
- Characteristics of unreliable channel determine complexity of a reliable protocol
  - Packet loss
  - Bit errors and data corruption
  - Packet reordering
  - Packet duplication?

RECOVERING FROM CORRUPTION...

Acknowledgements

- Receiver gets correct packet
  - Returns *Acknowledgement* (ACK)
- Receiver gets corrupt packet
  - Returns *Negative Acknowledgement* (NAK)

(Negative) Acknowledgements

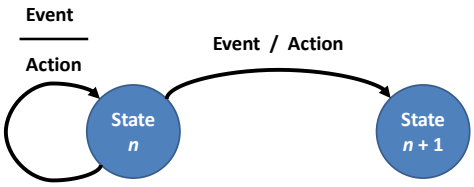


Automatic Repeat reQuest (ARQ)

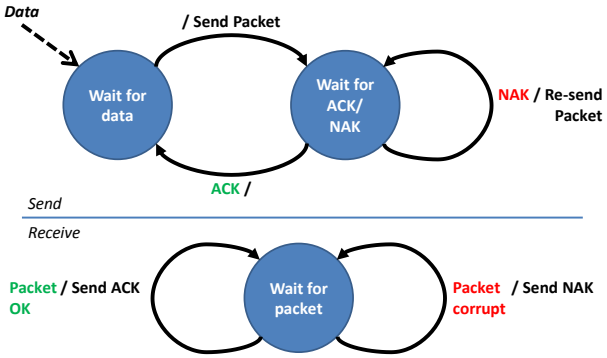
- ARQ Protocols have…
  - Error detection
    - Checksums, etc.
  - Receiver feedback
    - Such as ACK and NAK
  - Retransmission of corrupt packets
    - For now we assume packets cannot be lost

State Machines

- Good way of describing protocol operation
  - Two matched state machines
    - Sender and Receiver
    - Each like the following:

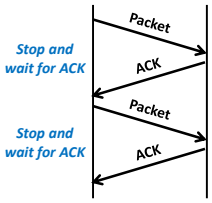


State Machine: *Send and Receive*



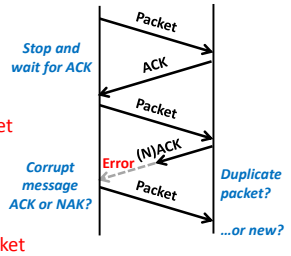
Stop and Wait Protocol

- Sender waits for ACK before sending next packet



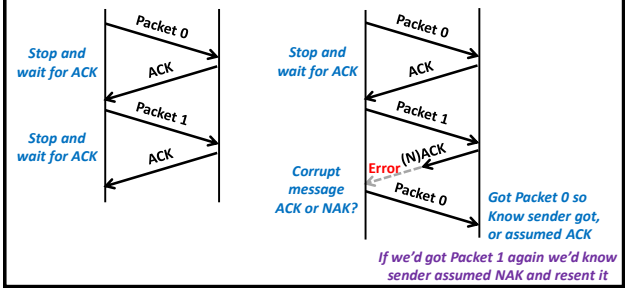
Stop and Wait: *corrupt (N)ACK*

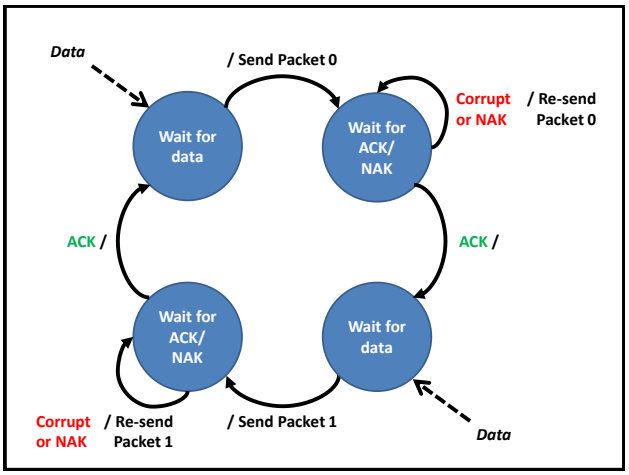
- What if response corrupt and can't tell whether ACK or NAK?
- **Assume ACK**, send new
  - If it was ACK – OK
  - If it was NAK – missing packet
- **Assume NAK**, resend old
  - If it was NAK – OK
  - If it was ACK – duplicate packet



Sequence Numbers

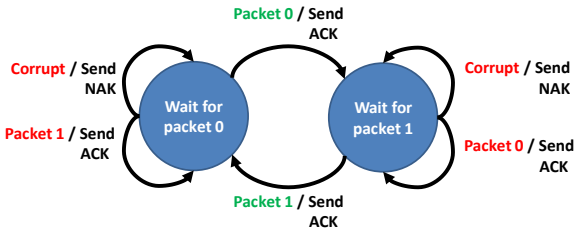
- A one bit sequence number allows both ends to spot peer 'misunderstanding' last message





Sequence Numbers

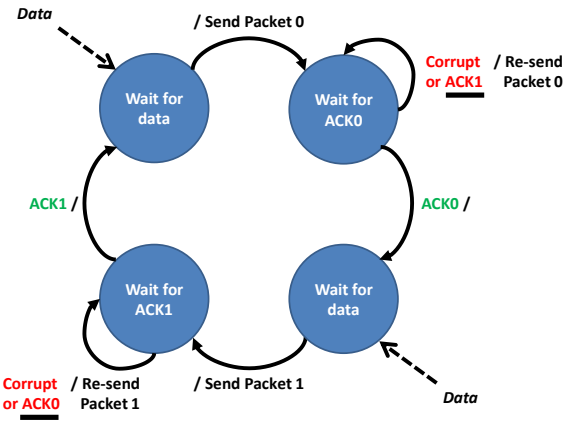
- Notice ACK/ NAK doesn't need seq. number
  - Only a non-corrupt ACK allows sender to continue



ACK*n* Sequence Numbers

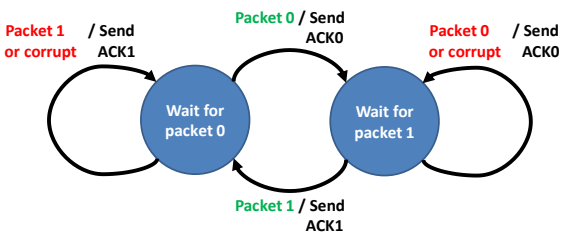
- Notice ACK and NAK don't need seq. number
  - Only a non-corrupt ACK allows sender to continue
- Alternative is to just have ACK*n* and no NAK
  - As in, ACK0, ACK1
  - ACK last packet sent, sender sends next packet
  - ACK previous packet, sender resends last packet
    - Thus a duplicate ACK*n* causes a resend

We can thus modify our state machine, replacing each NAK with an ACK*n*…



ACKn Sequence Numbers: receive

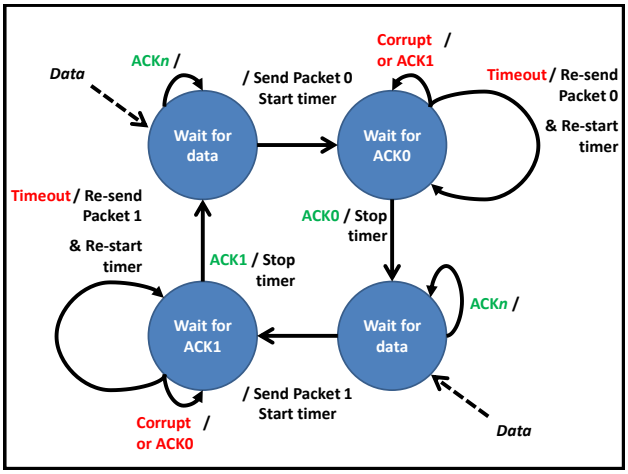
- As you can see, this simplifies the check
  - Anything other than the expected ACK causes a resend



HANDLING LOST PACKETS

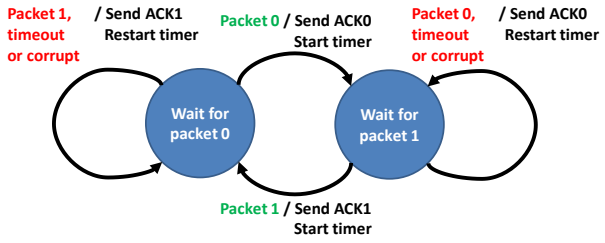
Handling Lost Packets

- Introduce idea of a *timeout*
  - Need to
    - Start timer when we send anything
    - Stop timer when we receive expected response
    - Re-start timer if timeout or unexpected response
- Resend last packet or ACK in event of a timeout
- If anything unexpected received, it is ignored
  - Again, the expected ACK causes us to move on



Adding Timeout: receive

- This protocol is know as the ***Alternating Bit Protocol***
  - Sequence numbers alternate between 0 and 1



Selecting Timeout Value

- **Too short** and we resend unnecessarily
  - Duplicate packets or ACKs
  - Not good, but protocol can handle this
- **Too long** and any problem gives long delay
- Value needs to be at least *Round Trip Time* (RTT)
  - Or we'll resend everything
  - Remember: includes buffering and processing time

We have a reliable protocol, but…

1. Send packet
2. Wait at least RTT for ACK
3. Send next packet

- Round Trip Time (RTT)
  - Lancaster to London: ~9ms
  - Lancaster to Las Vegas: ~144ms

  } ***What value do we use?***
- Wrong value could seriously impact performance

Time to Transmit a Packet

- Transmission delay

  d_{trans} = packet length / transmission rate
- So, if we send a 1000 byte packet at 1Gbps
- d_{trans} = 1000 x 8 bits / 1Gbps
  = 8000 / 10^9
  = 0.8µs
- Takes 0.8µs to get whole packet onto network link

What's the Utilisation of the Network?

- If we assume zero length ACK sent instantly on receipt
  - ACK received after d_{trans} + RTT
- Lancaster to Las Vegas = 0.8µs + 144ms
  = 144.008ms
- $Utilisation_{time}$ = active time / total time
  = 0.008 / 144.008
  = 0.000,055,55
- For our 1Gbps link: 1Gbps * 0.000,055,55 = 56kbps!

That's almost back to…

On a 1Gbps link!

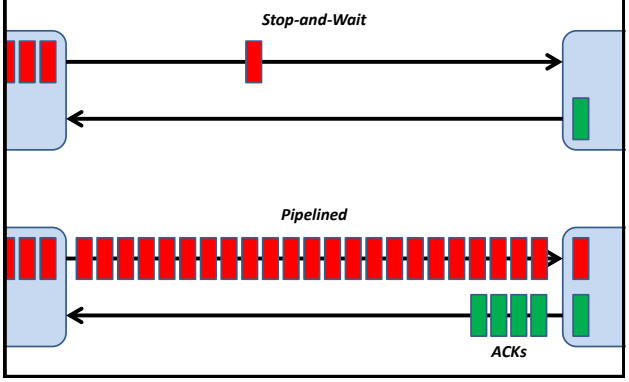
A different approach…

PIPELINING…

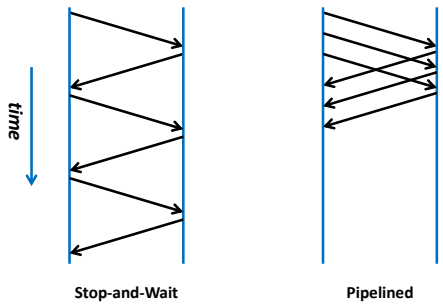
Pipelined Transmission

- Increase range of sequence numbers
- Keep sending packets until out of sequence numbers
  - ACKs free sequence numbers for re-use
- Range of sequence numbers required depends on
  - Protocol error handling
  - Transmission speed, faster → more seq. numbers
  - Round Trip Time, longer → more seq. numbers

Stop-and-Wait vs Pipelined

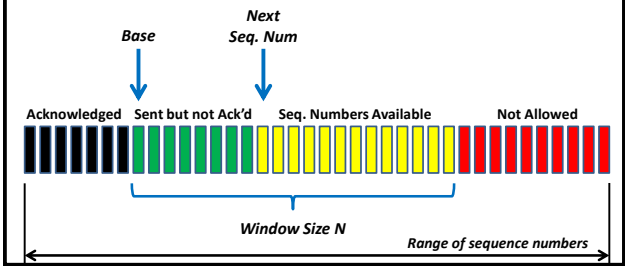


Comparing Performance

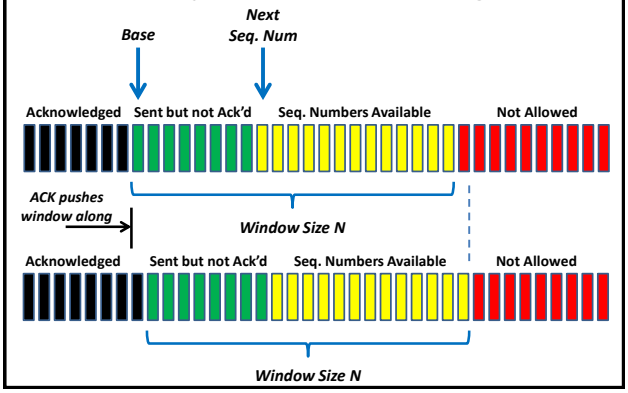


Sliding Window Based Systems

- Each ACK frees a sequence number for re-use
  - Window slides forward by one with each ACK
  - Known as a ***Sliding Window Protocol***

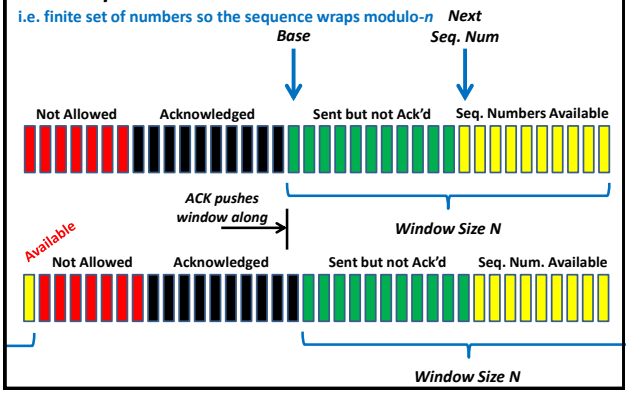


ACKs push window along…



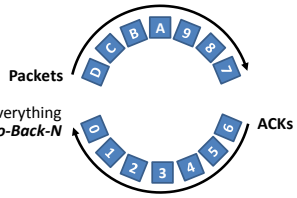
Sequence numbers use Modulo-N

i.e. finite set of numbers so the sequence wraps modulo-*n*



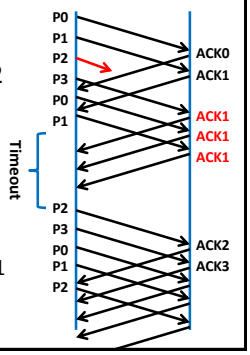
Window Size

- Limited by
  - Number of bits for sequence number
    - *n* bits, seq. numbers: 0 .. 2^n -1
  - Desire to limit number of un-ACK'd packets
    - Maintain feedback to sender
    - Congestion/ Flow control
    - Error constraints
  - **REMEMBER**
    Many protocols retransmit everything from corrupt/ lost packet : ***Go-Back-N***



Go-Back-N

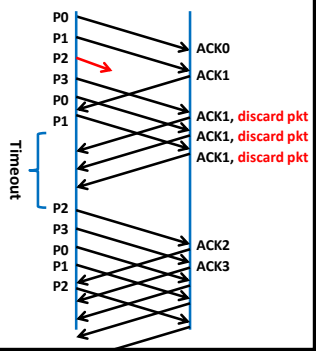
- Packet 2 lost
- Sender cannot reuse seq. #2
  - Receiver keeps ACKing #1
- Timeout waiting for ACK2
- Sender goes back to 2
  - Resends packets 2, 3, 0, and 1



Refinements: *simplifying receiver*

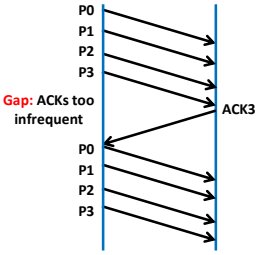
- Receiver discards out of order packets
  - They'll be resent
    - Though could then be lost
  - No need to handle re-ordering packets in receive buffer

    e.g. Lost 2, discard 3, …



Cumulative Acknowledgements

- No need to send every ACK
  - ACK*n*, ACKs all packets up to and including *n*
- To avoid gaps
  - Window size needs to be big enough, and…
  - ACKs often enough
    - And don't forget the timeout



Bandwidth-Delay Product

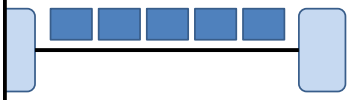
- Large bandwidth + large end-to-end delay

  → large number of 'in-flight' packets

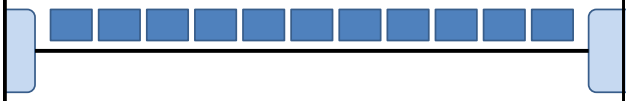
- With Go-Back-N a single error causes all subsequent packets to be resent

Bandwidth-Delay Product

- Low (end-to-end) delay



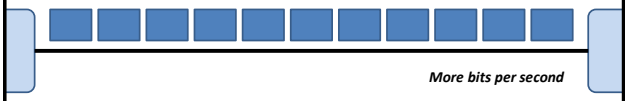
- High (end-to-end) delay



Bandwidth-Delay Product

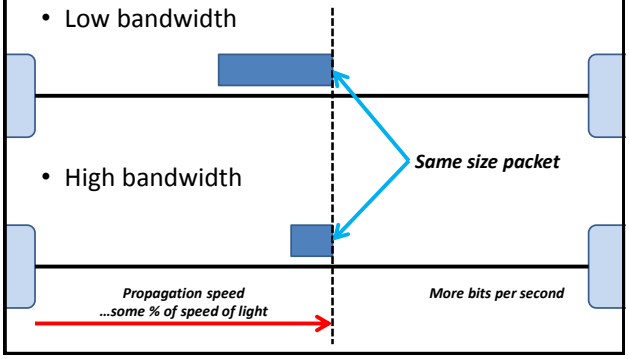
- Low bandwidth

- High bandwidth



Bandwidth-Delay Product

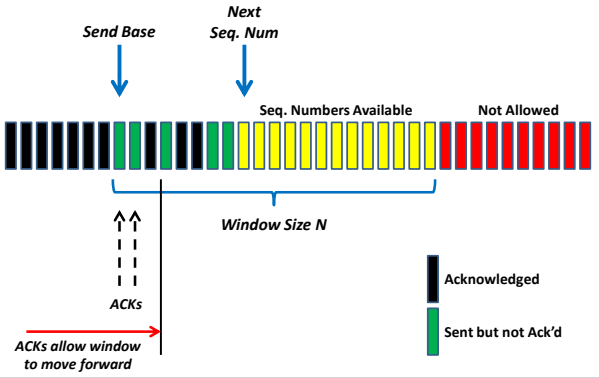
- Low bandwidth
- High bandwidth



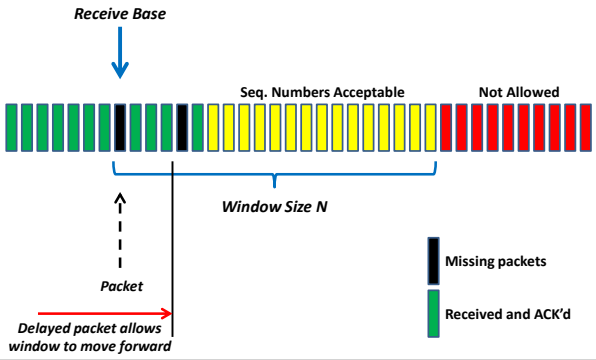
An Alternative: *Selective-Repeat*

- Receiver
  - Individually ACKs all packets (in window) *including resends*
  - Out of order packets accepted
  - Slides window forward only when no missing messages
- Sender
  - Slides window forward once all ACKs to that point received
  - Need timeout for each sent packet as resent individually

Selective-Repeat: *sender*

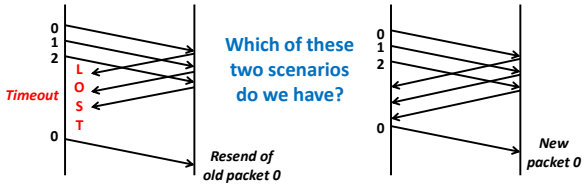


Selective-Repeat: *receiver*



Large Windows in Selective-Repeat

- Sender may not distinguish retransmissions
  - Remember end-point knows nothing of other
  - Window size **MUST be <= half seq. number space**



An aside…

- It should now be clear that there are many situations where sender and receiver can have very different views of the world
  - Protocols must reconcile these differences

OUT OF ORDER PACKETS

Out of Order Packets

- Sequence number space MUST be large enough to ensure packets cannot 'live' in network long enough to reappear
  - Packet TTL limits lifetime (at least in hops)
  - Can be serious problem with slow networks
    - e.g. GPRS network can introduce ***very*** long delays
  - Internet protocols assume ~2-3min lifetime possible

Reliability Mechanisms

- Checksums
- Timers
- Sequence Numbers
- Acknowledgements (incl. NAKs)
- Sliding windows and pipelining