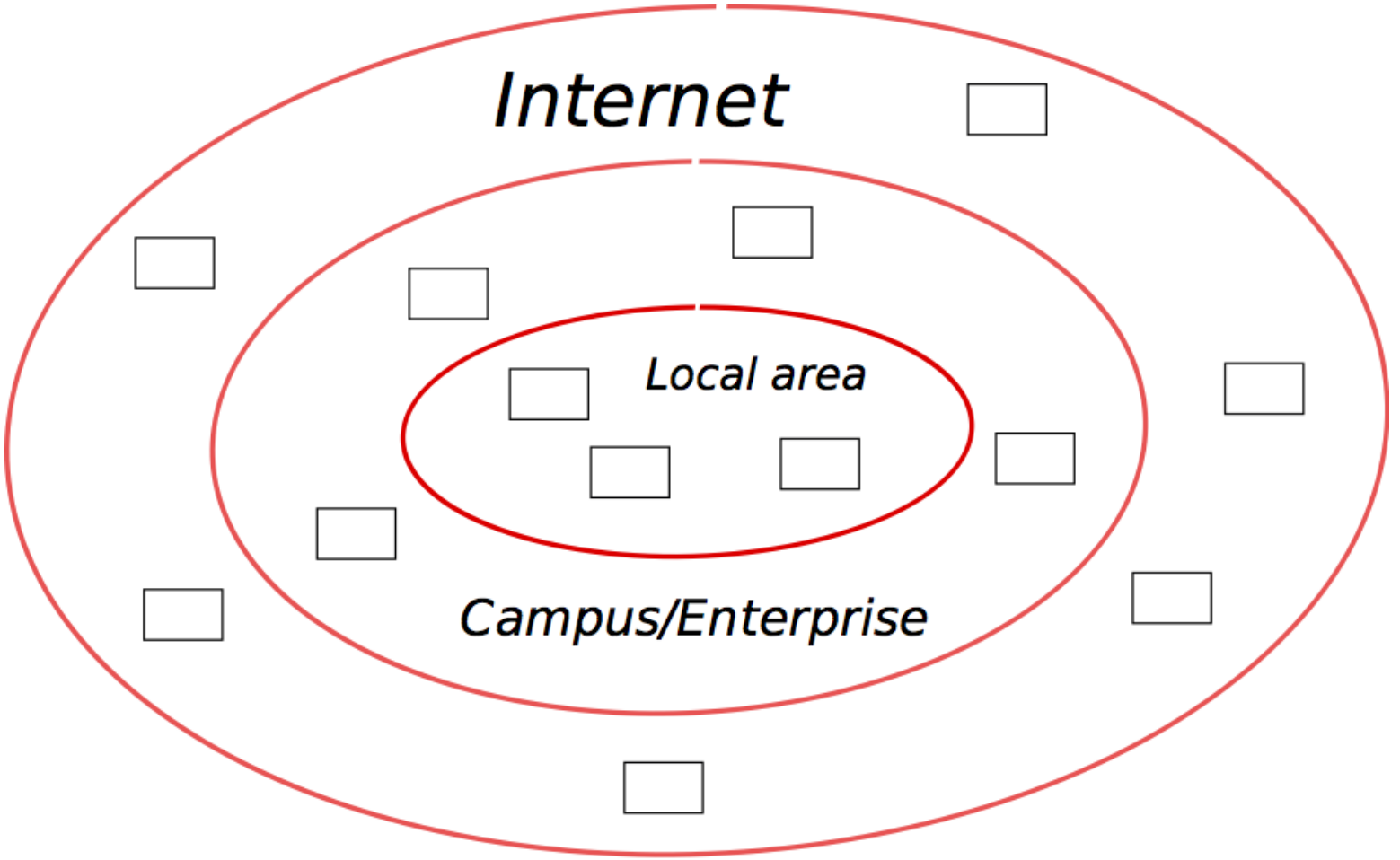
Computer Networking: from LANs to Internet

Nevil Brownlee

n.brownlee@auckland.ac.nz

Friday, 5 Jan 2018

Getting computers (hosts) to communicate

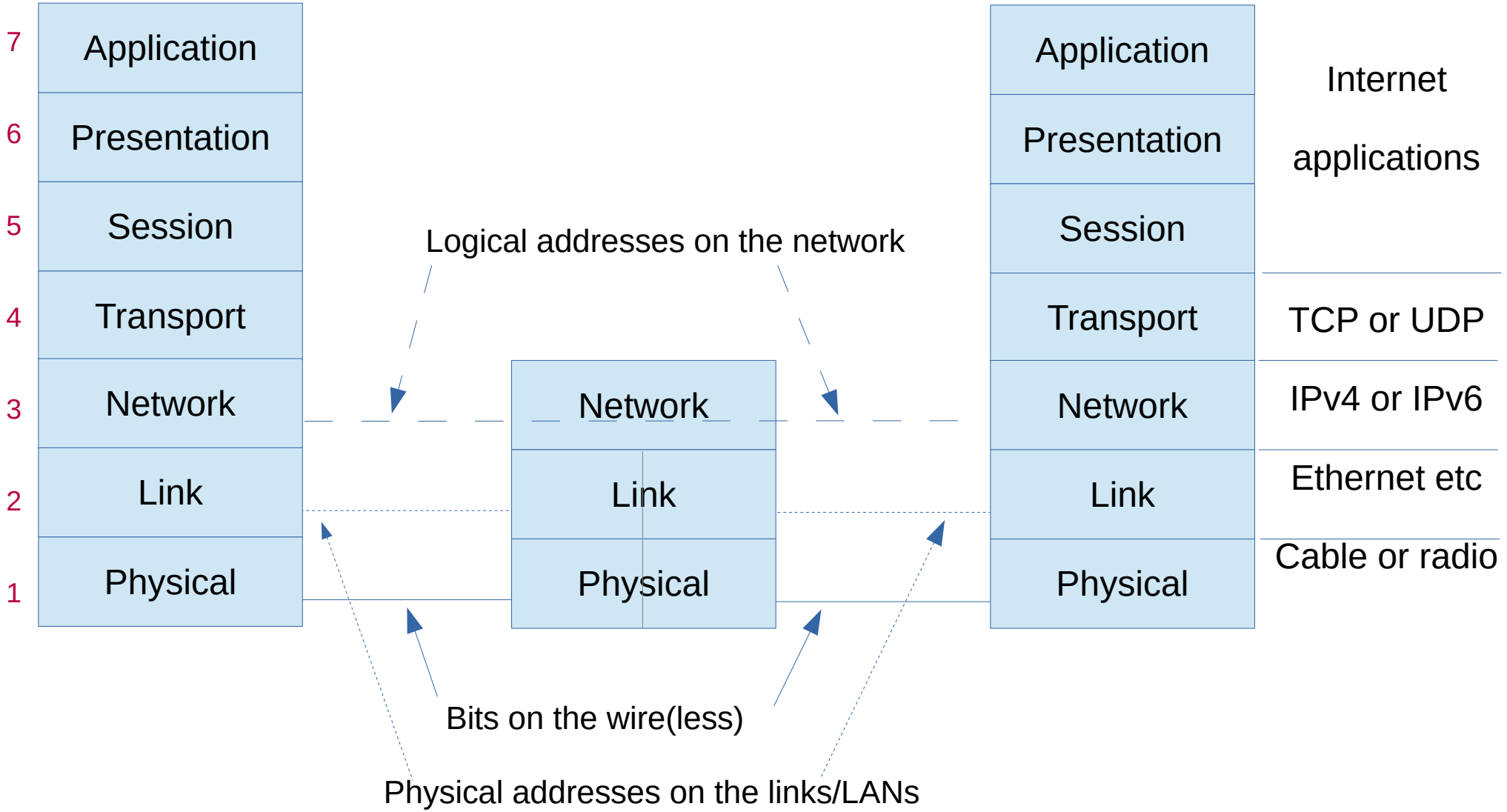


Rectangles = hosts, Ellipses show increasingly larger networks

Network layers, the seven-layer model

- *Any* computer should be able to communicate with *any other* computer
- Doing that in a single application program is too hard
- Instead, we split the problem up into seven sub-problems — the "protocol layers"
- Layers are an abstraction. They provide a simple view of what happens in a communication system
- Layer *n*
  - provides services to layer *n+1*
  - uses services from layer *n-1*

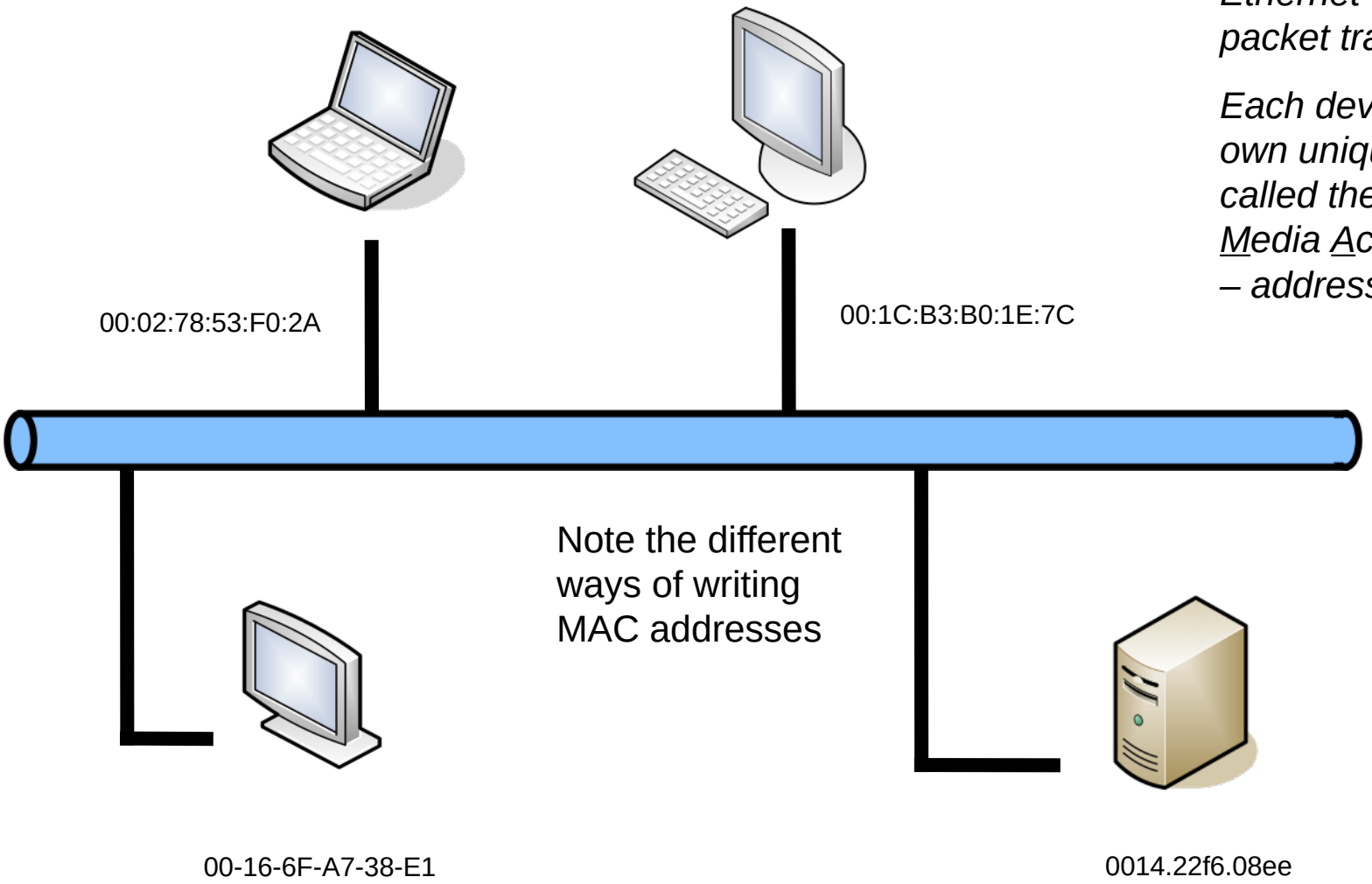
ISO seven-layer formal model



Protocols

- A network *protocol* is a set of rules that defines how devices talk to each other
  - it's an algorithm with two (or maybe more) separate participants (*hosts*)
  - they'd better obey the rules; otherwise communication will fail
- Many any examples of protocols around us, e.g:
  - traffic rules (Road Code)
  - telephone conversations

An Ethernet Bus Network



All devices on the Ethernet will see all packet transmissions

Each device has its own unique address called the MAC – Media Access Control – address

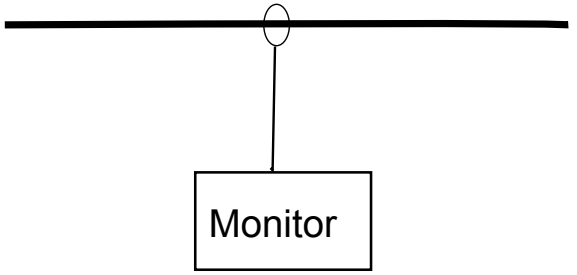
Example protocol: Ethernet on a shared bus

- These days most user computers use a wireless physical layer (smartphones, tablets, notebooks, etc)
- Ethernet was developed using a shared bus (i.e. a common cable, the *ether*)
- Devices (*hosts*) sharing the *ether* must cope with *collisions* on it
  - wait until medium is quiet
  - send packet, but watch for collision(s)
  - if collision, wait random time, then try again

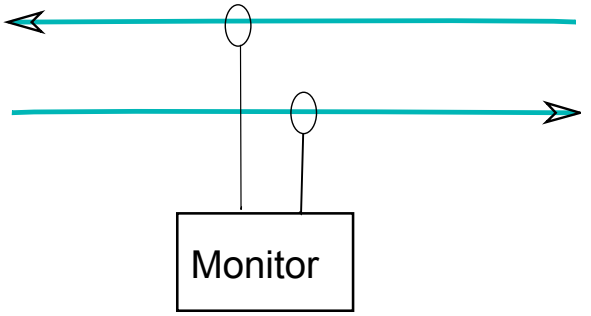
Internet Standards

- Internet protocols are documented as *Standards* by SDOs (Standards Development Organisations)
  - IETF (Internet Engineering Task Force)
    - published as RFCs (Requests for Comment)
    - e.g. TCP is RFC 793, September 1981
  - IEEE (Institute of Electrical and Electronic Engineers)
    - e.g. Ethernet is IEEE 802.3, WiFi is IEEE 802.11
  - W3C (World Wide Web Consortium)
  - and lots more … !

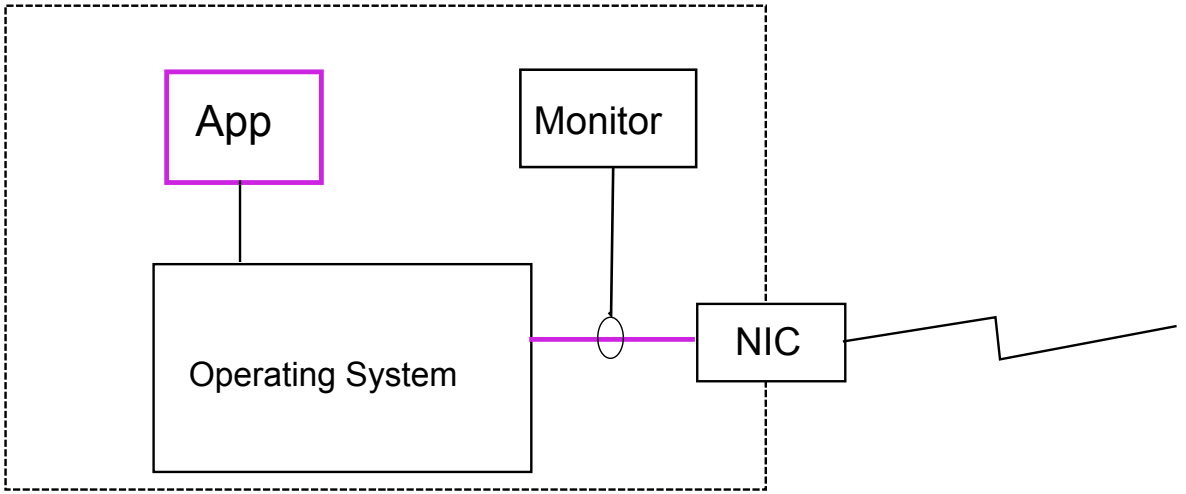
Looking at packets 'in flight'



Copper link



Fibre-optic link

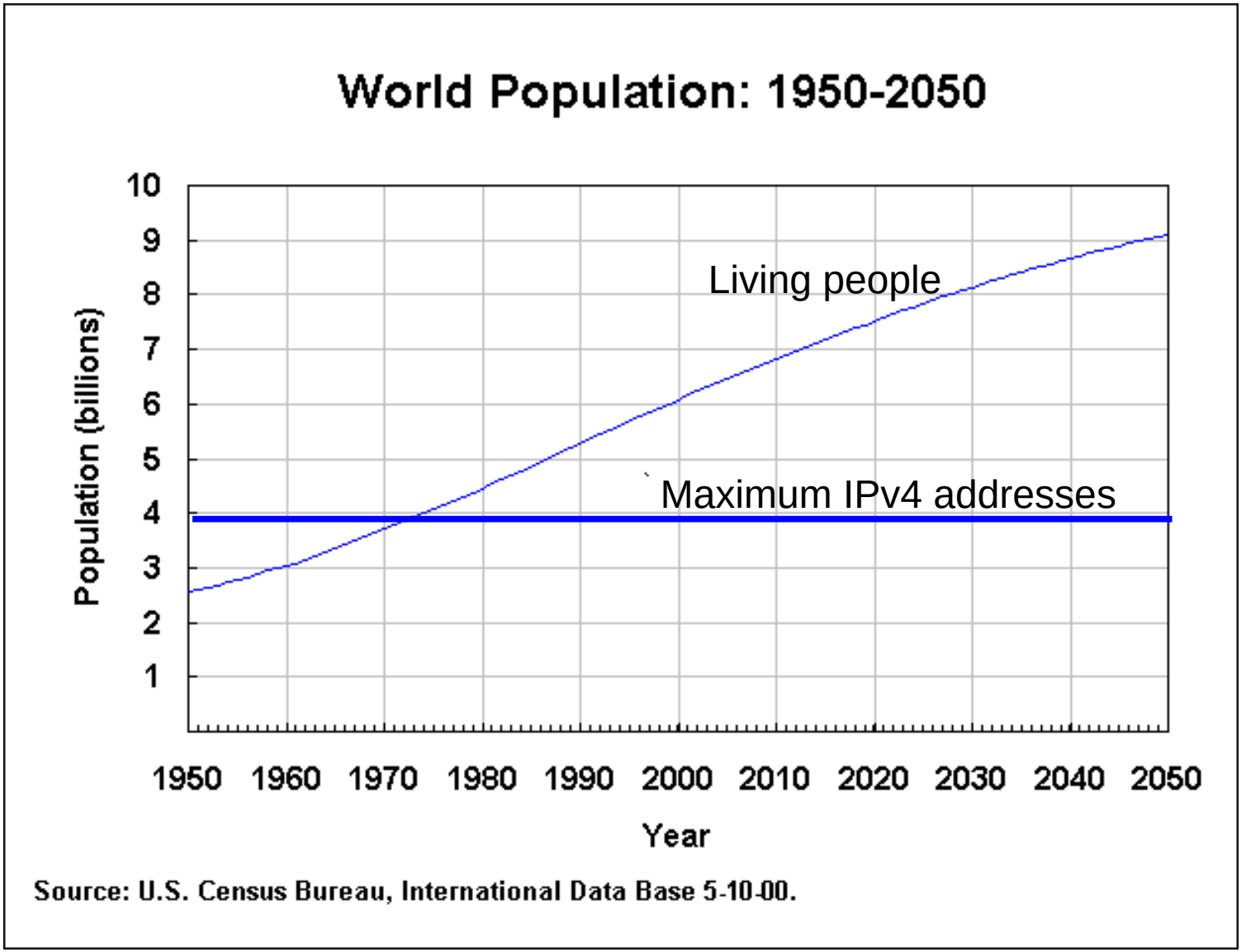


Computer

Layer 3, network layer

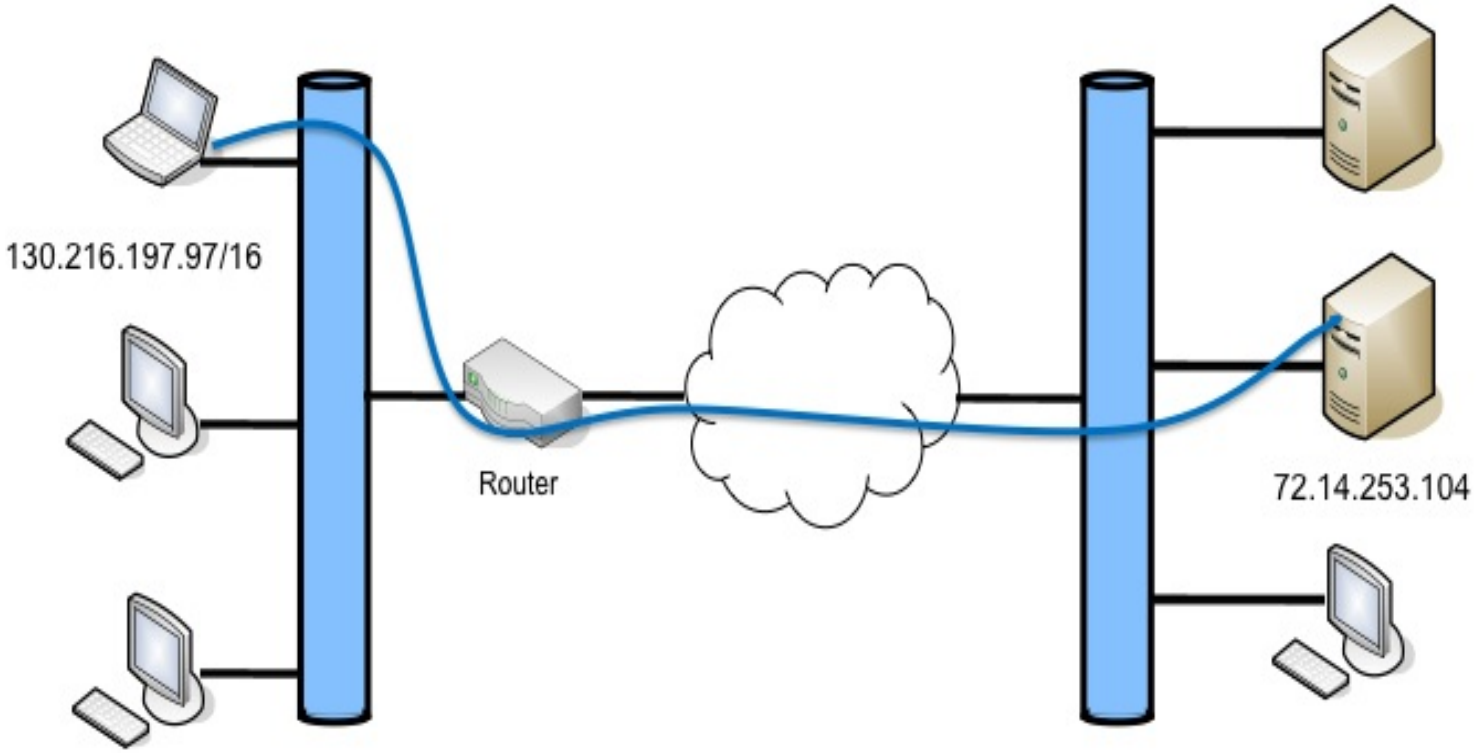
- Layers 1 and 2 (Physical and Link) allow two hosts on a LAN to send messages using physical (MAC) addresses
- At layer 3 (network, i.e. interconnecting LANs) the hosts use logical (IP) addresses
- IP exists in two flavours:
  - v4 is currently the common one, using 32-bit (4 byte) addresses, e.g. 130.216.35.123/16
  - v6 is the new one that is catching up, it uses 128-bit (16 byte) addresses. IPv6 is used throughout our Computer Science Department, e.g. 2001:df0::2006:1184:4d85:73f6:7033/48

Why we need IPv6



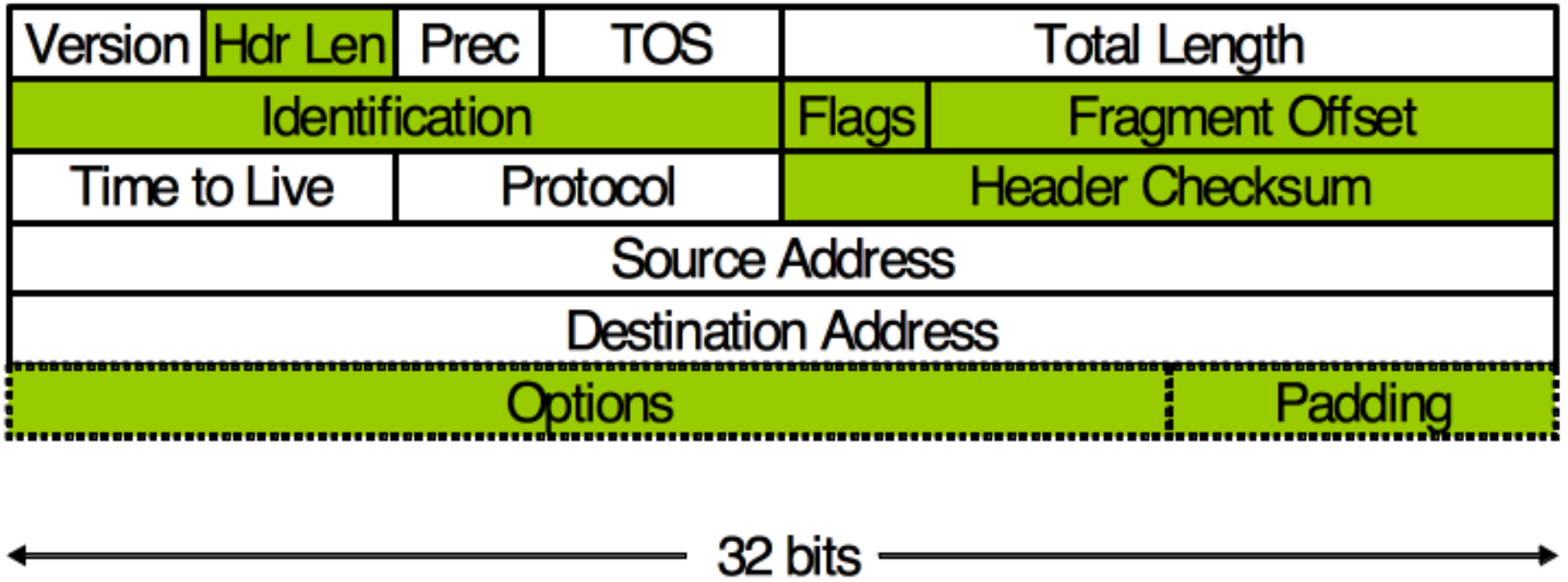
Obviously, having fewer addresses than people is silly

Delivering IP Packets



- Networks (e.g. LANs) are interconnected by *routers*
- Routers know the *IP address blocks* of all the hosts in the networks they connect to

An IPv4 datagram's header



- Source and Dest IP addresses allow two hosts to send IP *datagrams* from one to the other. *Payload* follows the datagram header
- Protocol indicates what's in the packet payload, e.g. 6 is TCP
- TTL (time to live) stops packets looping forever in the network!

[diagram credit: Steve Deering]

Layer 4, Transport Layer

- An IP datagram goes from host to host. The actual data transfer, however, is between applications running on those hosts (e.g. a web browser such as Firefox and a web server such as Apache)
  - the data is carried by a *transport protocol*, e.g. TCP or UDP
  - if the host is running two web browser instances, how can we know which of the two instances the data from the Apache server gets delivered to?
- We need to distinguish applications running on the same host
  - an application that wants to communicate with another application (usually running on a different host) will grab a *port* through which it channels the communication.
  - that port distinguishes the application

User Datagram Protocol

- Protocol 17
- UDP is a transport level protocol that adds ports to IP
- It is as unreliable as IP and the delivery is “best-effort”
- Unlike IPv4, which only checks the header for corruption in transit, UDP checks for corruption in data as well
  - the UDP’s checksum field includes both header and data while IP’s only include header
  - delivery is not guaranteed, but if the data is delivered, then it is error-free (when the checksum is correct – but this is done under the hood for an application)

UDP Header

Source Port (16 bits)	Destination Port (16 bits)
Length (16 bits)	Checksum (16 bits)
Data	

- Checksum is a 16-bit ones-complement sum of the data
  - It is used to check that UDP data is delivered correctly
  - It may also be set to zero, i.e. not used!

Transmission Control Protocol

- Most of the time, what we want is a protocol that reliably delivers data in the order it was sent
  - file transfers (web browsing), email, etc. require reliability

- TCP provides reliable data delivery and guarantees the *byte* ordering at reception is the same as the transmission
  - this comes at an extra cost. TCP has a higher overhead than UDP
  - UDP makes no guarantee on data delivery, but if data is delivered, UDP can (if its checksum is used) guarantee the received data to be correct (i.e. error-free)
  - TCP guarantees delivery as well as correctness

Transmission Control Protocol (2)

- TCP uses IP for transport, but provides transport-layer functionality that IP itself doesn't provide (hence, TCP/IP)
  - *ports* to distinguish applications running within the same host
    - This is the same as UDP
  - re-transmission of dropped packets
    - Note that IP can discard packets when there is network congestion
    - A slow receiver can discard packets when its receive buffers are full
    - A packet can be discarded at the physical layer because of bit errors (due to noise or interference)
  - re-arranging out-of-order packets
  - *handshake* to establish a TCP connection between two Applications (IP address + port)

How TCP achieves reliability

- Two mechanisms: A checksum in each packet, and sequence numbering of bytes with acknowledgement
- Each packet contains a sequence number for its first payload byte. Missing packets are detected by a sequence number that is too far ahead compared to the number of bytes actually received
- Received bytes are acknowledged (in an acknowledgment packet) by returning the sequence number of the *next expected byte* to the sender
- Sender resends unacknowledged bytes after a certain timeout
- Checksum detects errors in TCP packets – packet is dropped at receiver and correction is done by re-transmission

Layer 5, Applications – DNS over UDP

- DNS is the Internet's Domain Name System
- DNS maps domain names to IP addresses
- It's a distributed database that can look up information for a domain, e.g. www.auckland.ac.nz
- DNS mostly uses UDP for its transport
  - UDP doesn't use handshakes (it's connectionless)
  - if a DNS request isn't answered, a nameserver will resend the request after a timeout (it doesn't need reliability)

TCP Applications

- WWW World Wide Web
  - Sends requests for URLs to a Web Server
    - web pages can be big, they need reliable transmission (including retransmission)
- SMTP Simple Mail Transfer Protocol
  - moves email messages between Mail Servers
    - again, reliability is essential
    - mail servers may store and forward email messages
  - user email client Apps use other protocols, e.g. IMAP

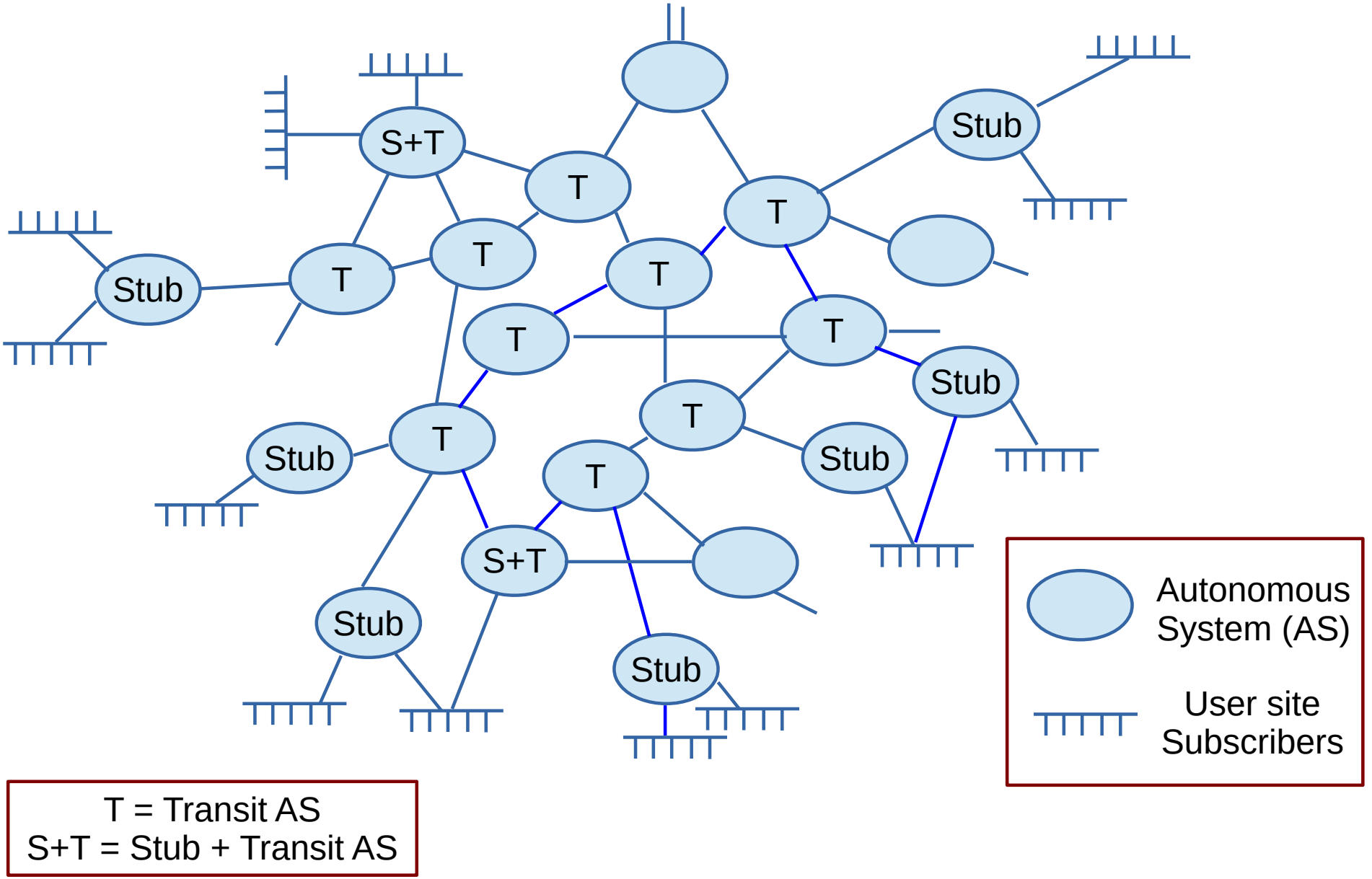
Global Routing: the problem

- The real network is massive
  - thousands of nodes on campus, millions per country, hundreds of millions worldwide
- The real network is constantly changing
  - nodes, routers and links added or removed constantly
  - View of network will be different in different places
- Therefore, algorithms must be scaleable, dynamic and distributed
  - central static route computation only works for small, very stable networks

Global Routing: how it's done

- Each router maintains a list of networks (address blocks) it knows about, and
- Each router maintains a database of where to send packets for all other address blocks
- Routers swap information about how to reach address blocks using a *routing distribution protocol*
- The Internet's only routing protocol is BGP4

Overview of the BGP4 system



New: QUIC transport protocol

- Introduced by Google in 2014
  - on content servers, and on Android
- Transports web content (http) over UDP
  - no opening handshake (as done by TCP)
  - all content encrypted (using TLS 3)
  - recovery from lost packets said to be better than TCP
- Now being standardised in the IETF
  - work under way for QUIC to transport more protocols that just http

Newish: Content Distribution Networks

- Large content providers run many big Data Centres world-wide
  - e.g. Google, Akamai, Cloudflare, etc
- They all have copies of the same content
- If a server doesn’t have an item when a user clicks on content, it fetches it from the 'best other' server, and caches it
  - 'best' usually means 'shortest time to fetch'
- The effect of CDN's is to move all content closer to its users, i.e. it reduces the time to fetch and display it

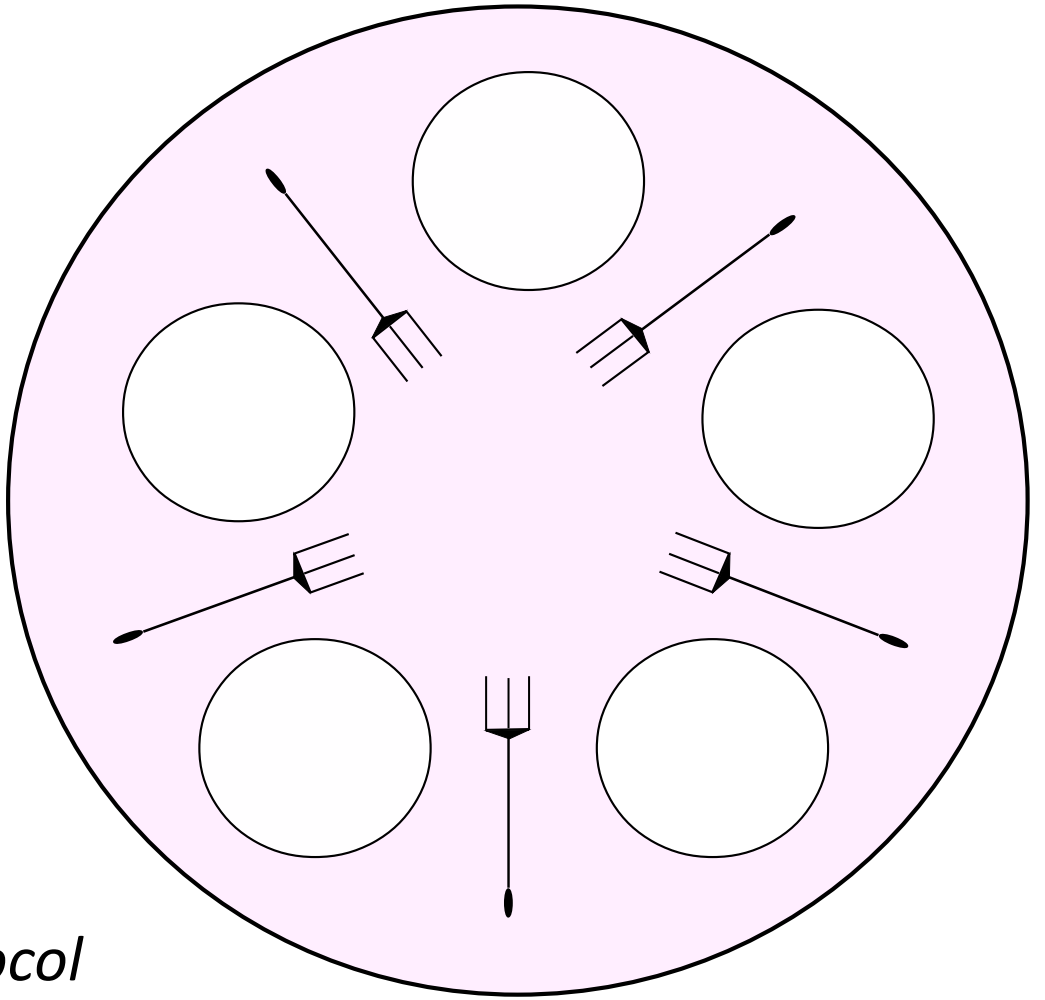
Older: Software Defined Networking

- Switches determine the path for each packet travelling through a network, using proprietary software
- SDN uses simple commodity hardware to do the actual switching for every *flow*
  - the switch selects output ports based on its own *flow tables*
  - when the first packet of a new flow appears, the switch sends it to an SDN Controller
  - the controller determines a complete path for the flow's packets, and downloads it to each switch's flow table

Question time !

A problem for you: the Dining Philosophers

- E. W. Dijkstra, 1965
- Five silent philosophers sit at a table, where they alternately think and eat
- The table has five forks and five plates of spaghetti
- To eat, a philosopher picks up the two forks by his plate
- A philosopher can only eat when neither of his neighbours are eating
- *What's the philosophers protocol if they're not to starve?*



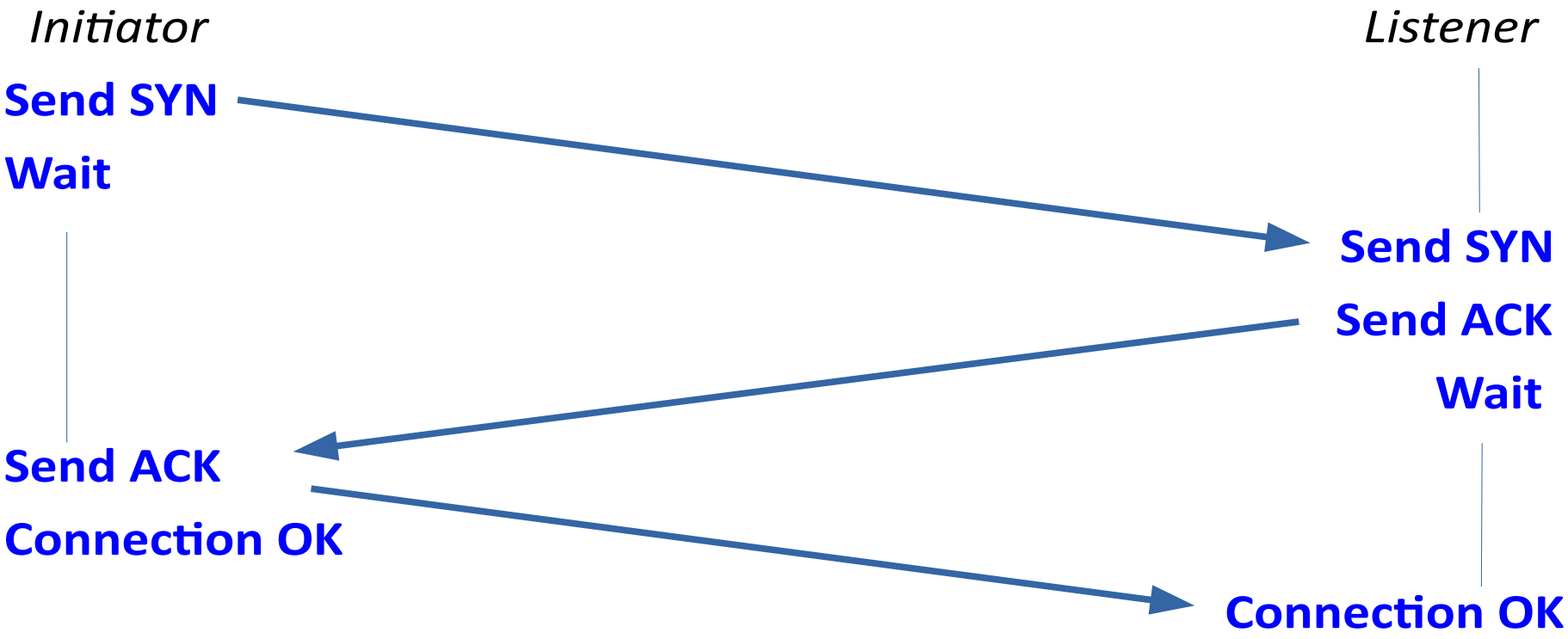
Network *trace* files, Wireshark

- Open-source network trace analyser tool
- Can also view a live network interface
- Can interpret and display many protocols
- Has lots of filtering features
- Plenty of tutorials / how-to-use material on the web, e.g.
  - http://www.howtogeek.com/104278/how-to-use-wireshark-to-capture-filter-and-inspect-packets

Who allocates IP addresses?

- For individual users: their Internet Service Provider (ISP)
- For an ISP: their Regional Internet Registry (RIR)
  - there are five RIRs: AfriNIC, APNIC, ARIN, LACNIC and RIPE NCC
- For an RIR: IANA (Internet Assigned Numbers Authority)

Connection establishment



- Note that whoever goes first, one SYN and one ACK datagram is sent in each direction
  - this will work even if both initiate simultaneously
  - listener's SYN and ACK are usually sent in a single datagram

TCP's *Sliding Windows* in action