University of Glasgow

The Changing Internet

Networked Systems (H) – Lecture 1

2019-2020

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow | This work is licensed under the Creative Commons Attribution-NoDerivatives 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nd/4.0/ or send a letter to Creative Commons, PO Box 1866,

Lecture Outline

- Administrivia
- Review of the traditional Internet architecture
- The changing Internet

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Administrivia

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Course Coordinator

- Lecturer and course coordinator
  - Dr Colin Perkins
  - colin.perkins@glasgow.ac.uk
  - https://csperkins.org/
- Office hours:
  - Thursday 13:00-14:00 – S101b Lilybank Gardens

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Course Materials

- Lecture notes and other material on online:
  - https://csperkins.org/teaching/2019-2020/networked-systems/ (or on the School's Moodle site)

- Paper handouts and lecture recordings will not be provided
  - The act of taking notes helps learning
  - The act of revising by reviewing your notes and other textbooks gives a useful additional perspective on the material, helps you understand the topic in depth
  - Lecture recordings discourage questions – it's already hard enough to ask a question in a large lecture theatre, but still harder when you know it's being recorded

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Aims and Objectives

- To introduce fundamental concepts and theory of communications
- To provide a solid understanding of the technology that supports modern networked computer systems
- To introduce low-level network programming, and give students practice with systems programming in C
- To give students the ability to evaluate and advise industry on the use and deployment of networked systems

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Intended Learning Outcomes

- By the end of the course, you should be able to:
  - Describe and compare capabilities of various communication technologies and techniques;
  - Know the differences between networks of different scale, and how these affect their design;
  - Describe the issues in connecting heterogeneous networks;
  - Describe importance of layering, and the OSI reference model;
  - Understand demands of different applications on quality of service requirements for the underlying communication network;
  - Demonstrate an understanding of the design and operation of an IP network, such as the Internet, and explain the purpose and function of its various components; and
  - Write simple low-level communication software, showing awareness of good practice for correct and secure programming

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Course Outline

Week	Lecture	Laboratory Session
1	#1: The Changing Internet	#1: Introduction to Network Programming in C
2	#2: Connection Establishment in a Fragmented Network	#2: Connection Establishment
3	#3: Secure Communications	#3: Secure Communications
4	#4: Improving Secure Connection Establishment	
5	#5: Reliability and Data Transfer	#4: TCP Behaviour and Congestion Control
6	#6: Lowering Latency	
7	#7: Real-time and Interactive Applications	
8	#8: Naming and the Tussle for Control	#5: Naming and Network Topology
9	#9: Networks and Inter-domain Routing	
10	#10: Future Directions	

The aims of the labs are to improve your understanding of the network, of network programming using the Sockets API, and to practice systems programming in C.

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Assessment

- Assessed exercises: 20%
  - Exercise 1: Secure Communications (10%)
    - Available: lab 3 – 30 January 2020
    - Deadline: lab 4 – 13 February 2020
  - Exercise 2: TCP Behaviour and Congestion Control (10%)
    - Available: lab 4 – 13 February 2020
    - Deadline: lab 5 – 5 March 2020
  - Don’t leave coursework to the last minute: it’s designed to be completed over several weeks, allowing time for thought and reflection on the material, and is too large to complete in a rush in a couple of days
- Examination: 80%
  - Exam format: answer all three questions

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Assessment of Coursework

- University code of assessment prescribes penalties for late submission, and for non-adherence to submission instructions
- These penalties will be strictly enforced

- Illness or special circumstances? Inform course coordinator *before* the deadline

University of Glasgow | School of Computing Science

Assessed Coursework

Course Name				
Coursework Number				
Deadline	Time:	4:30pm	Date:	
% Contribution to final course mark			This should take this many hours:	
Solo or Group ✓	Solo		Group	
Submission Instructions				
Who Will Mark This? ✓	Lecturer	Tutor	Other	
Feedback Type? ✓	Written	Oral	Both	
Individual or Generic? ✓	Generic	Individual	Both	
Other Feedback Notes				
Discussion in Class? ✓	Yes	No		
Please Note: This Coursework cannot be Re-Done				

Code of Assessment Rules for Coursework Submission

Deadlines for the submission of coursework which is to be formally assessed will be published in course documentation, and work which is submitted later than the deadline will be subject to penalty as set out below. The primary grade and secondary band awarded for coursework which is submitted after the published deadline will be calculated as follows:

(i) in respect of work submitted not more than five working days after the deadline
  a. the work will be assessed in the usual way;
  b. the primary grade and secondary band so determined will then be reduced by two secondary bands for each working day (or part of a working day) the work was submitted late.

(ii) work submitted more than five working days after the deadline will be awarded Grade H.

Penalties for late submission of coursework will not be imposed if good cause is established for the late submission. You should submit documents supporting good cause via MyCampus.

Penalty for non-adherence to Submission Instructions is 2 bands

You must complete an "Own Work" form via
http://www.dcs.gla.ac.uk/socs-online for all coursework
UNLESS submitted via Moodle

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Required Reading

- Any good text on computer networks, for example:
  - Peterson and Davie, Computer Networks: A Systems Approach, 5th Edition, Morgan Kaufman, 2011, ISBN 0123851386 (https://book.systemsapproach.org)
  - Kurose and Ross, Computer Networking: A Top-Down Approach, 6th Edition, Addison-Wesley, 2012, ISBN 0273768964
  - Tanenbaum and Wetherall, Computer Networks, 5th Edition, Prentice Hall, 2010, ISBN 0132553171
  - Bonaventure, Computer Networking: Principles, Protocols and Practice, online textbook (http://cnp3book.info.ucl.ac.be/index.html)
    - Blog relating to the book: https://obonaventure.github.io/cnp3blog/
- You are expected to read-along with the lectures – the lectures introduce the concepts, and the books provide detail

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

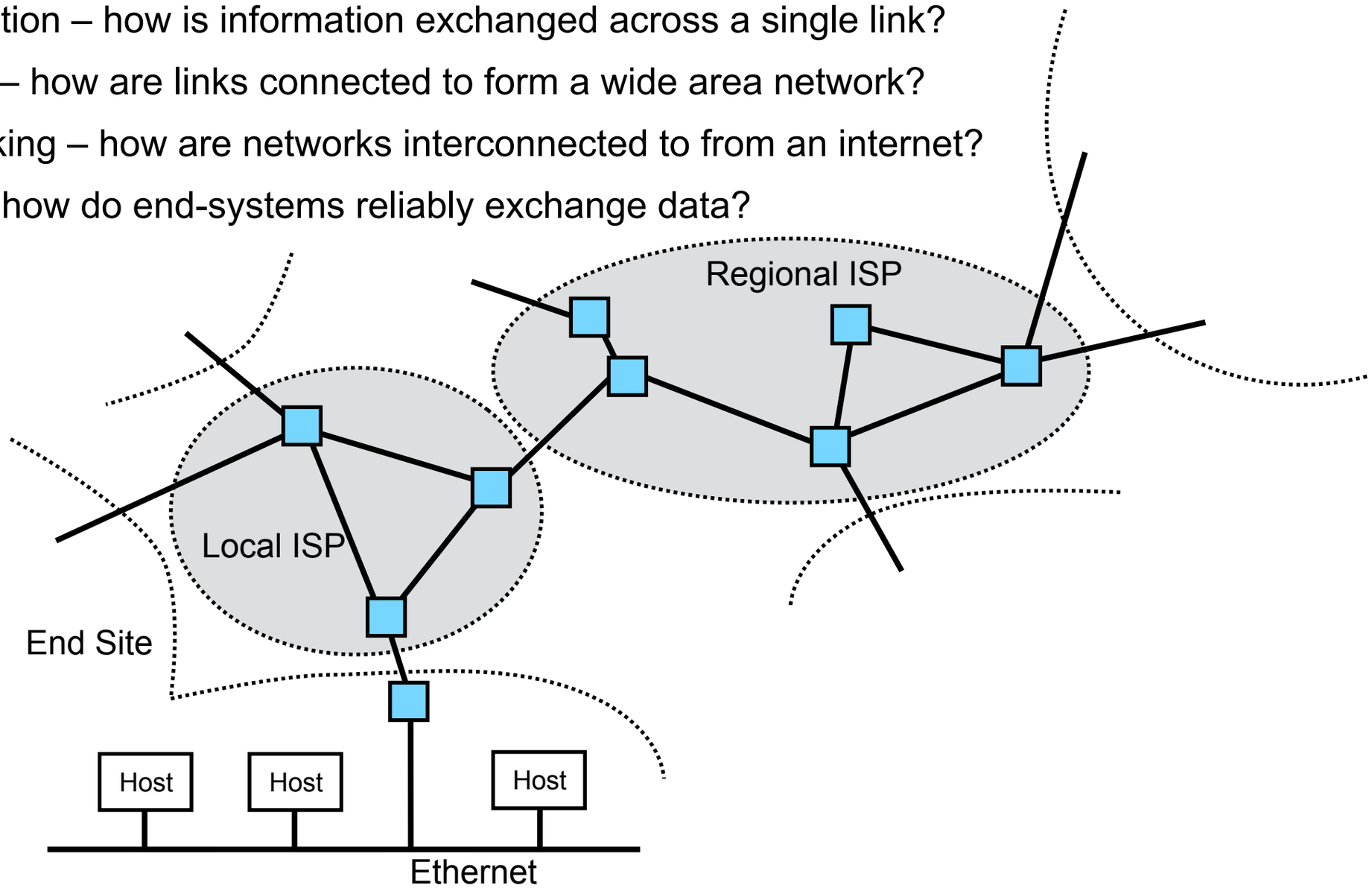
Review of the traditional Internet architecture

- Protocols and Layers
- Physical and data link layers
- IP and the network layer
- TCP and UDP transport
- Applications

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

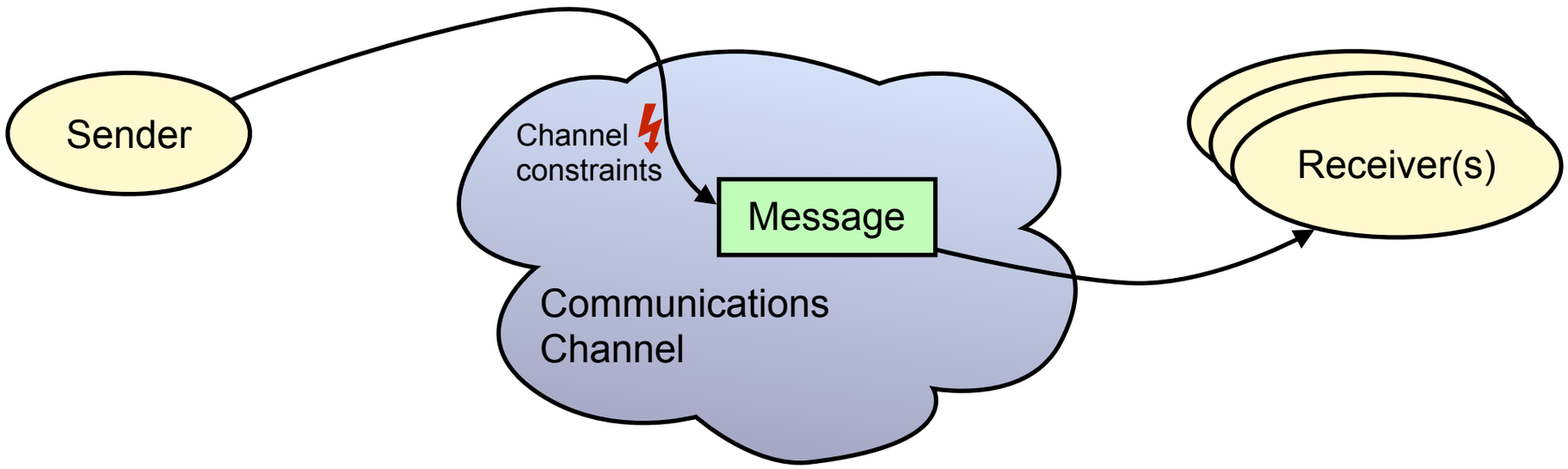
What is a Networked System?

- A cooperating set of autonomous computing devices that exchange data to perform some application goal
  - Communication – how is information exchanged across a single link?
  - Networking – how are links connected to form a wide area network?
  - Internetworking – how are networks interconnected to from an internet?
  - Transport – how do end-systems reliably exchange data?



Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

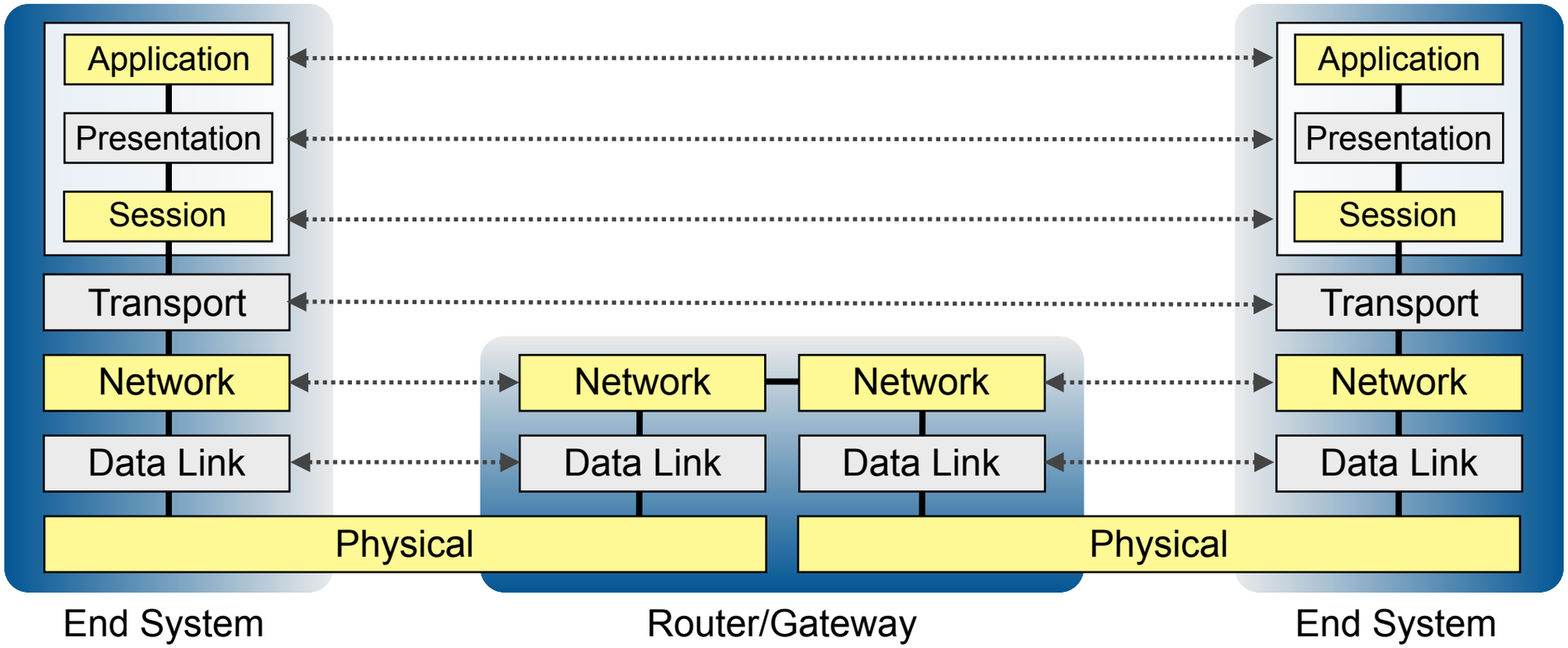
Communication Protocols



- Network protocols define message syntax, semantics, and communication patterns – HTTP, TCP, IP, etc.
- Channel constraints limit communications speed and reliability
- Protocols compose and layer to raise level of abstraction

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Protocol Layering: The OSI Reference Model



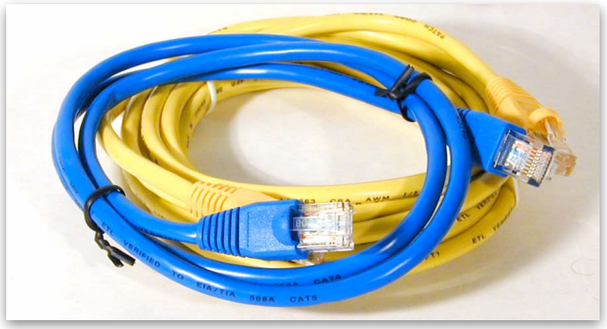
ISO/IEC 7498-1(1994) "Information technology — Open Systems Interconnection — Basic Reference Model: The Basic Model"

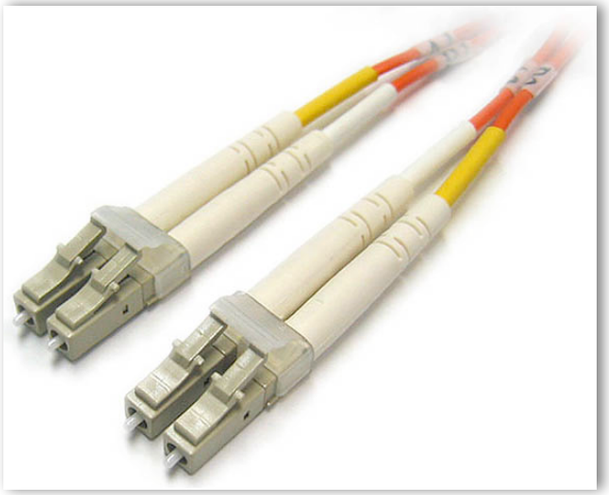
- A standard way of thinking about layered protocol design
  - The OSI protocols were not widely deployed
  - The Internet conflates the upper three layers
- A design tool – real networks are more complex, layer violations, sublayers, and tunnels are commonplace

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Physical and Data Link Layers

- Enable communication between directly connected devices
- Physical layer – transmission of a raw bitstream
  - What type of cable or wireless link do you use?
  - How to encode bits onto that channel?
    - Baseband encoding
    - Carrier modulation
  - What is the capacity of the channel?
- Link layer – framing, addressing, media access control
  - Structure the bitstream into meaningful frames of data
  - Detect and correct transmission errors
  - Identify devices
  - Arbitrate access to the channel







Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Physical Link Characteristics and Limitations

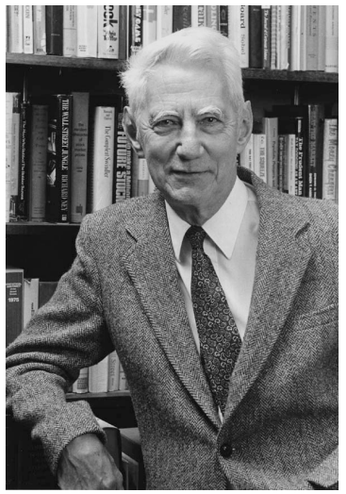
- Links have finite capacity → bandwidth, *H*
- Determines maximum transmission rate of signal, assuming a perfect channel: $R_{max} = 2H \log_2 V$
  - *V* = number of discrete values of each symbol



Source: IEEE

Harry Nyquist (1889-1976)

- Real-world channel capacity determined by channel noise
  - Simplest to model is Gaussian noise: uniform noise that impacts all frequencies equally
  - Maximum transmission rate of a channel subject to Gaussian noise: $R_{max} = H \log_2(1 + S/N)$
    - *S* = strength of signal, *N* = amount of noise



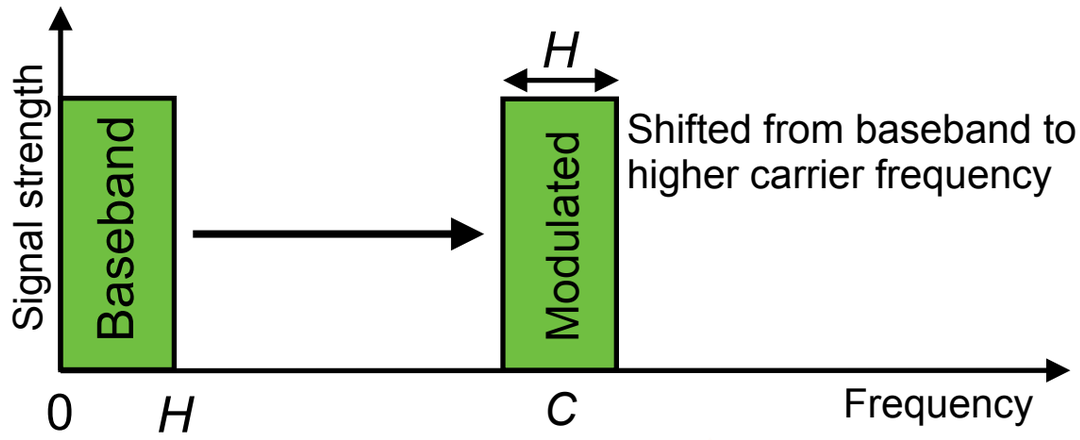
Source: AT&TT Bell Labs

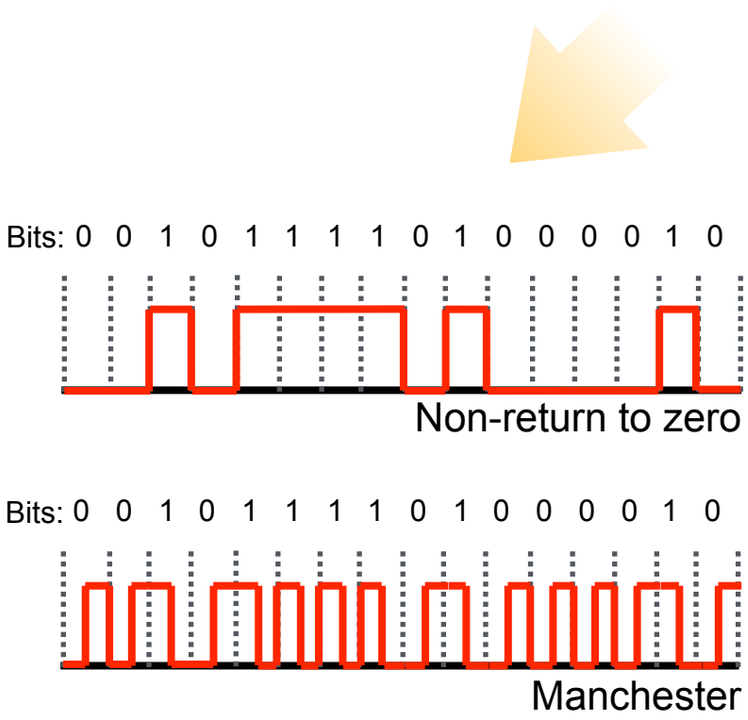
Claude Shannon (1916-2001)

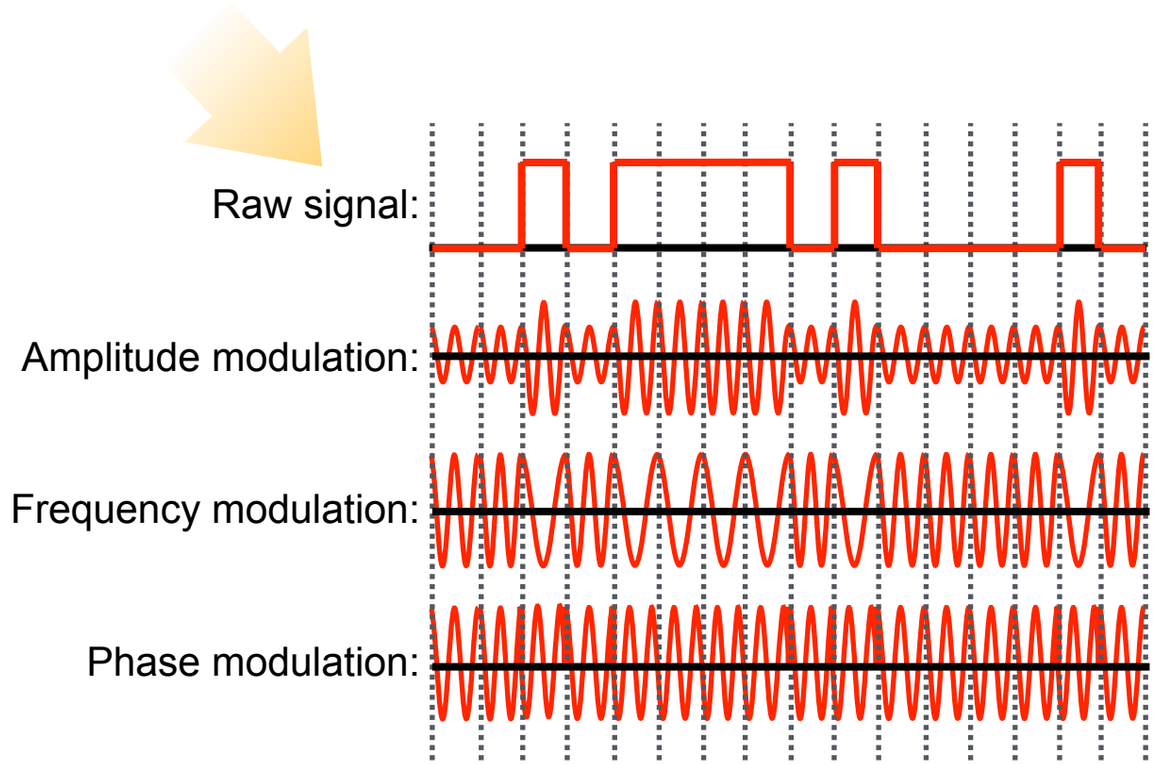
- Bandwidth and signal-to-noise ratio determine maximum rate of data transfer for a link

CC BY ND

Modulation





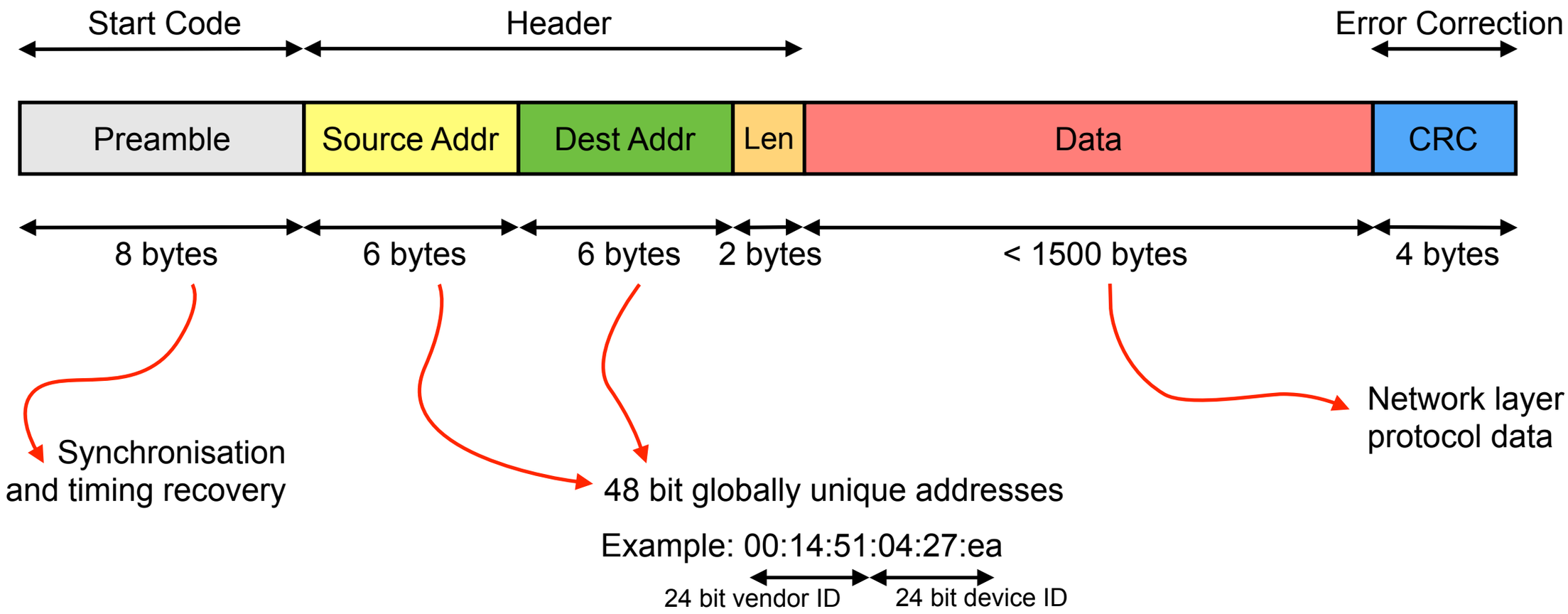


Modulation delivers a sequence of bits over the physical channel

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Framing and Addressing

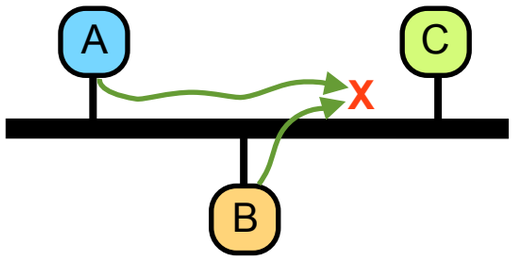
Example: Ethernet frame format

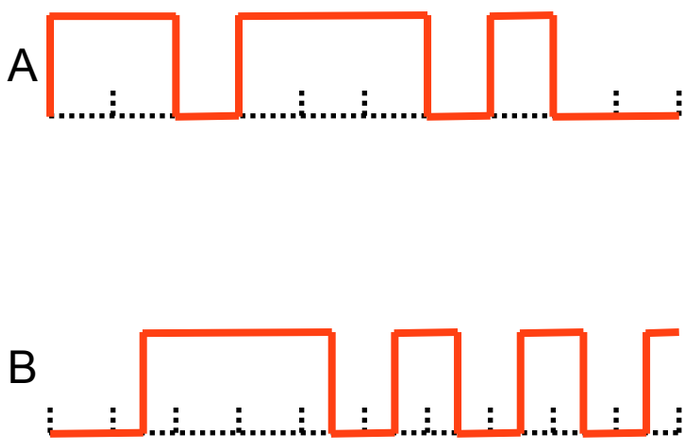


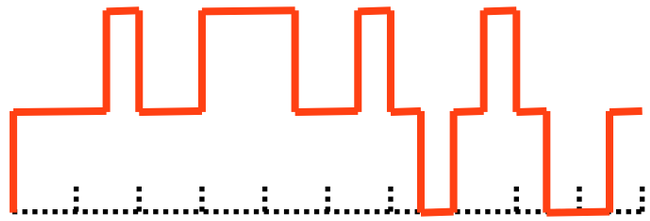
Separate the bitstream into meaningful frames of data

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

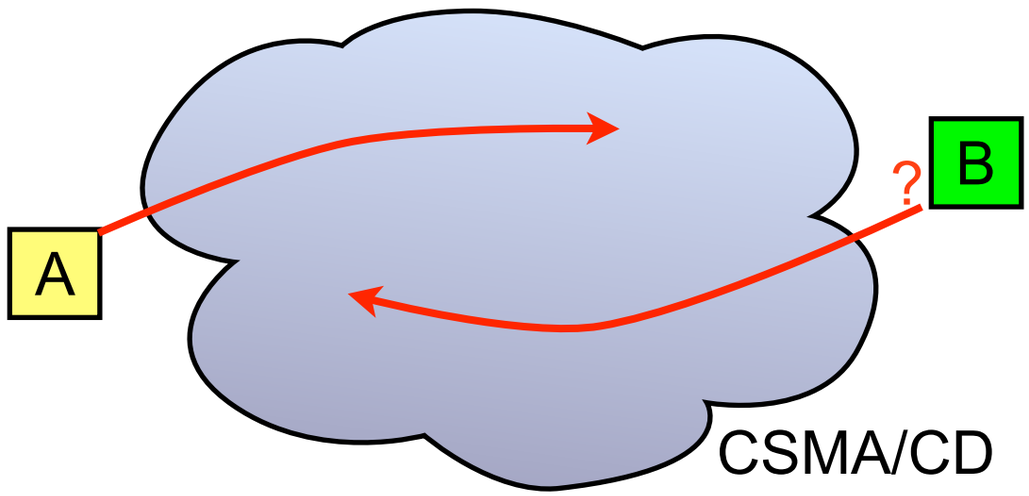
Media Access Control







Simultaneous transmission corrupts signal

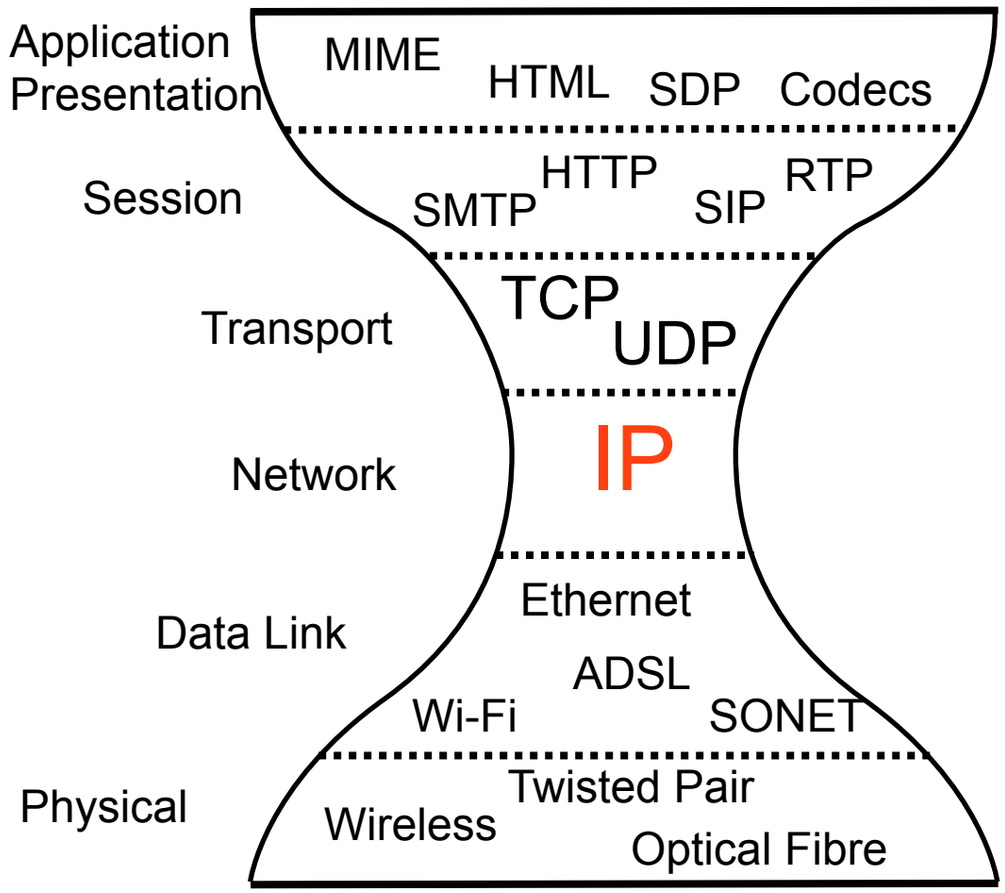


Control when frames are sent, to avoid collisions

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

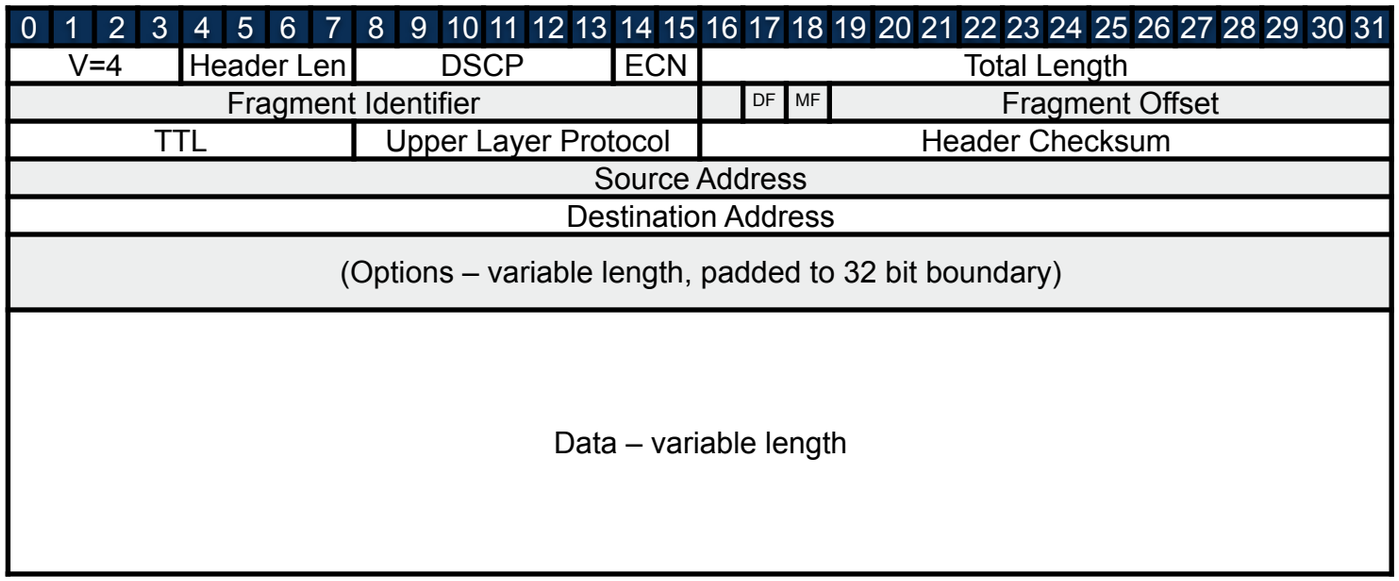
IP and the Network Layer

- Global inter-networking protocol
- Hour glass protocol stack
  - Single standard network layer protocol (IP)
    - Packet switched network, best effort service
    - Uniform network and host addressing
    - Uniform end-to-end connectivity – subject to firewall policy
  - Many transport & application layer protocols
  - Range of link-layer technologies supported
- Decouples end-to-end functionality from per-hop functionality



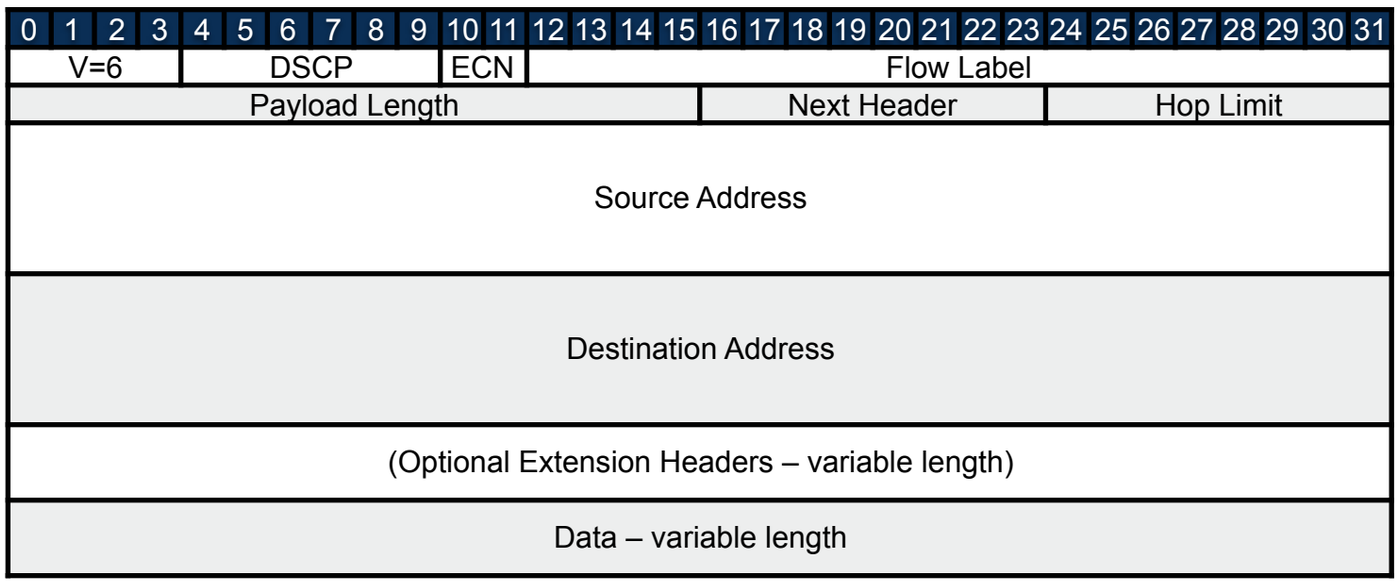
Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

IPv4 and IPv6



IPv4

- 32-bit addresses are insufficient
- Fragmentation difficult to support at high data rates
- Limited extensibility



IPv6

- Larger address space
- Removes in-network fragmentation
- Removes unnecessary checksum
- Simpler header format
- Simpler extension mechanism

IPv5? The Internet Stream Protocol (ST-II); an experimental multimedia streaming protocol developed between 1979 and 1995, but no longer used – http://www.ietf.org/rfc/rfc1819.txt

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

IP Addressing

- IP addresses encode location of network interface
  - If a host has multiple network interfaces (e.g., Wi-Fi and Ethernet), it will have multiple IP addresses
  - A host can support both IPv4 and IPv6 on each interface
  - A host can have multiple IP addresses of each type assigned to each interface
- DNS names are an application concept – not used by the network

IPv4

IP address: 130.209.247.112 = 10000010 11010001 11110111 01110000

Netmask: 255.255.240.0 = 11111111 11111111 11110000 00000000

20 bits → network = 130.209.240.0/20

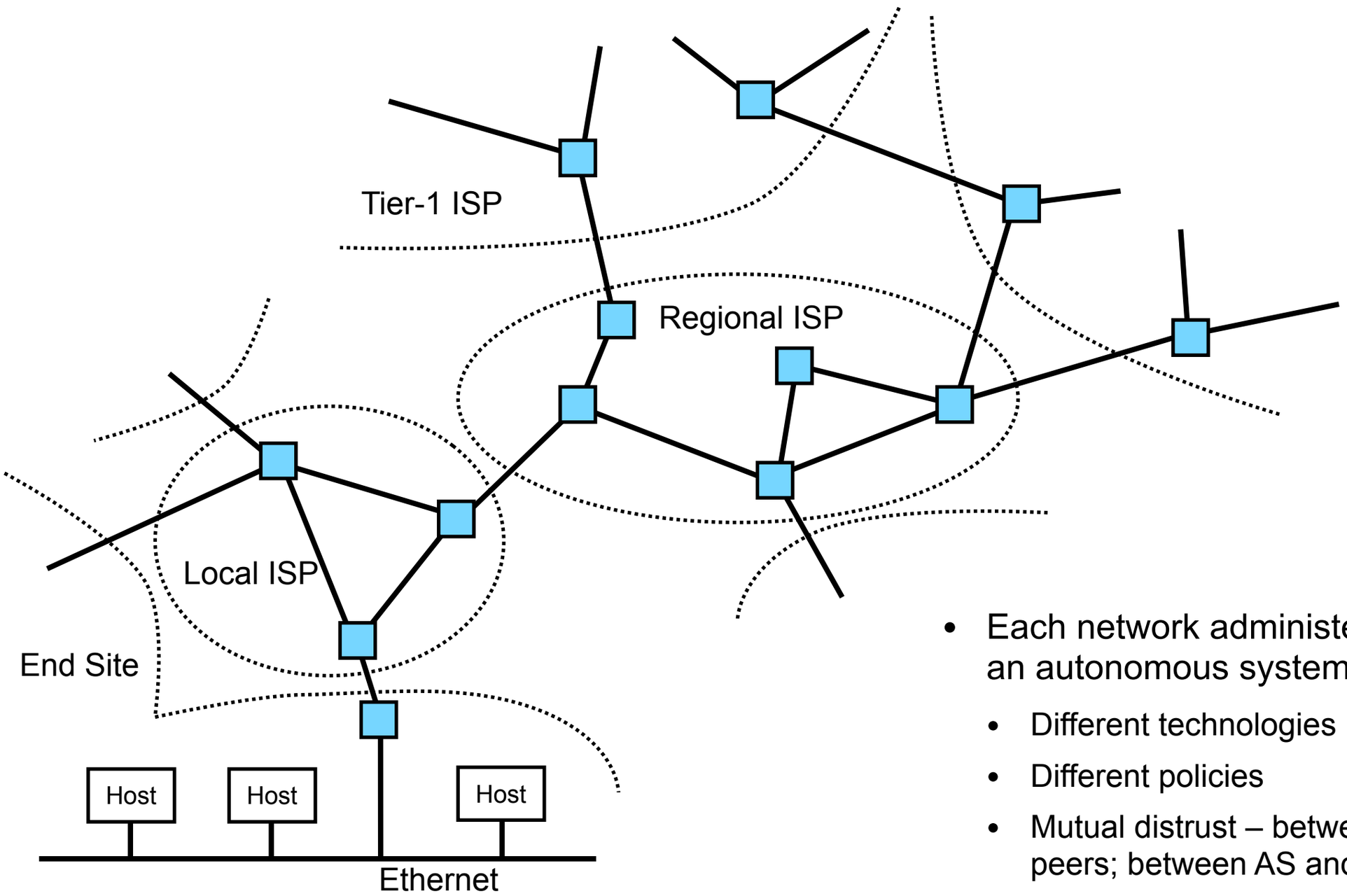
IPv6

2001:0db8:85a3:08d3:1319:8a2e:0370:7334

Network | Local identifier

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Routing

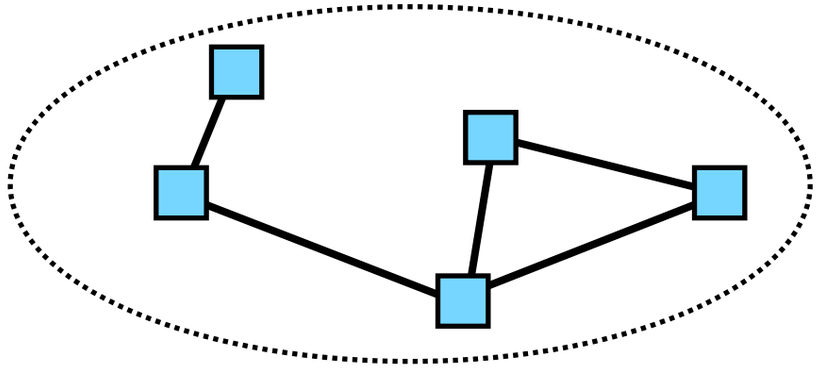


- Each network administered separately - an autonomous system (AS)
  - Different technologies
  - Different policies
  - Mutual distrust – between AS and its peers; between AS and its customers

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Autonomous Systems and Inter-domain Routing

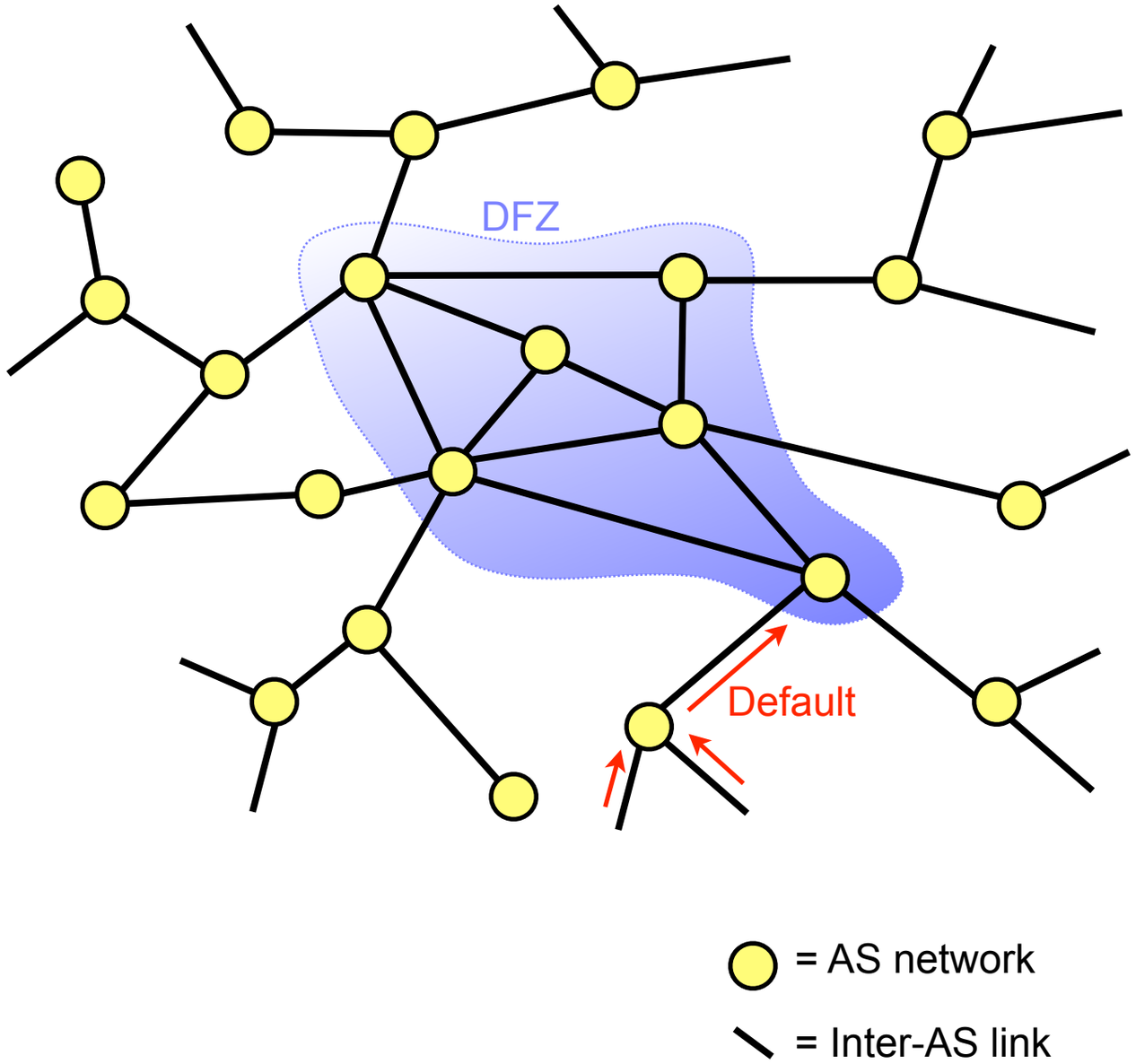
- Each network administered separately – an autonomous system (AS)
  - Separately administered; different technologies
  - Shortest path routing
  - Distance vector or link state algorithms



Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

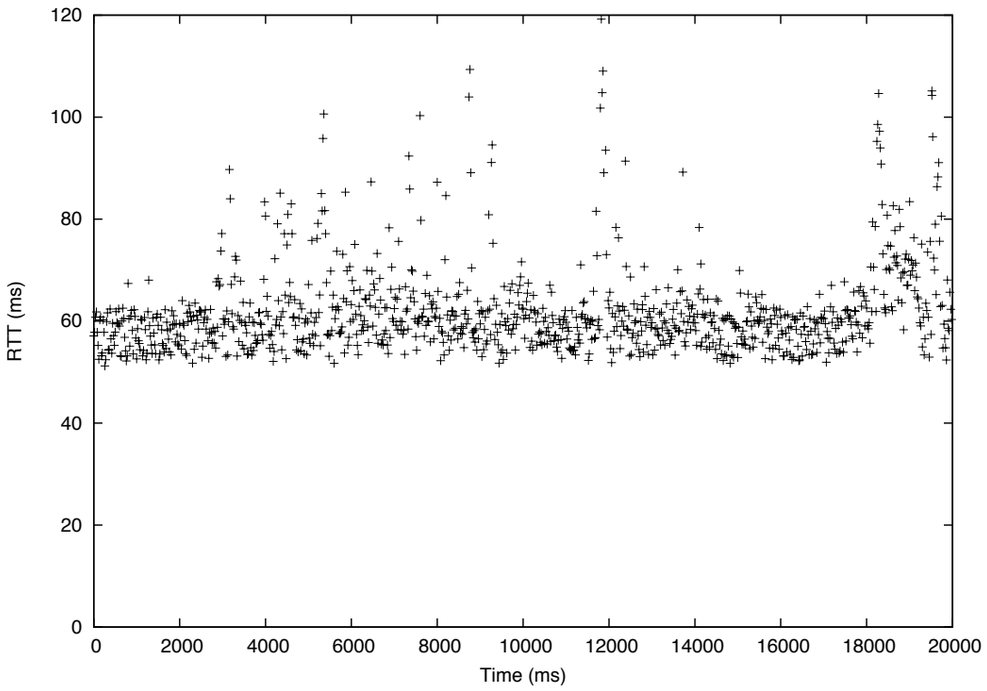
Inter-domain Routing and BGP

- Treats each network as a graph node and routes between ASes
- The AS-level topology:
  - Well connected core; sparse edges
  - Edge networks can use default route to the core
  - Core networks need full routing table: the default free zone (DFZ)
- Routing between competitors – no real trust between organisations
  - Policy, politics, and economics are key constraints; not shortest path
  - BGP routing protocol



Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

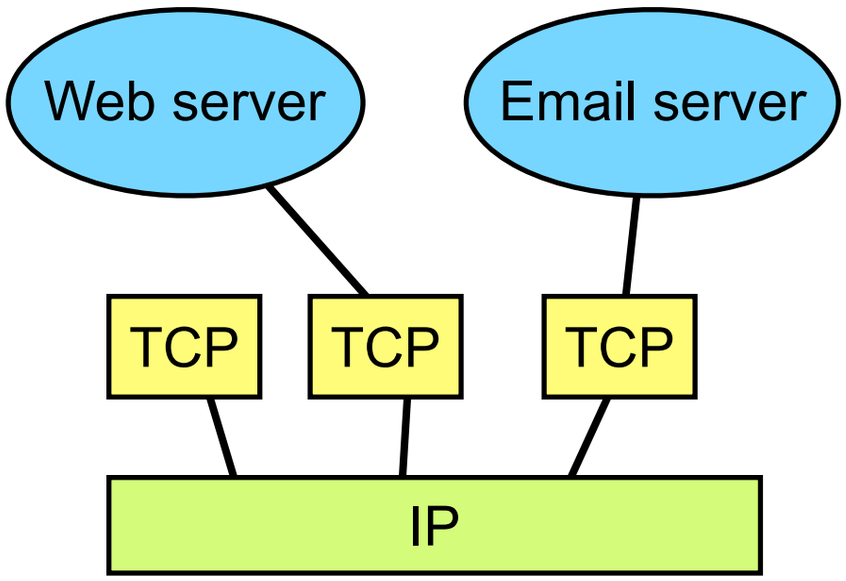
IP Service Model



- Best effort, connectionless, packet delivery
  - Just send – no need to setup a connection first
  - Network makes its best effort to deliver packets, but provides no guarantees
  - Time taken to transit the network may vary
  - Packets may be lost, delayed, reordered, duplicated or corrupted
  - The network discards packets it can't deliver
- Easily run over any type of link layer

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

TCP and UDP Transport



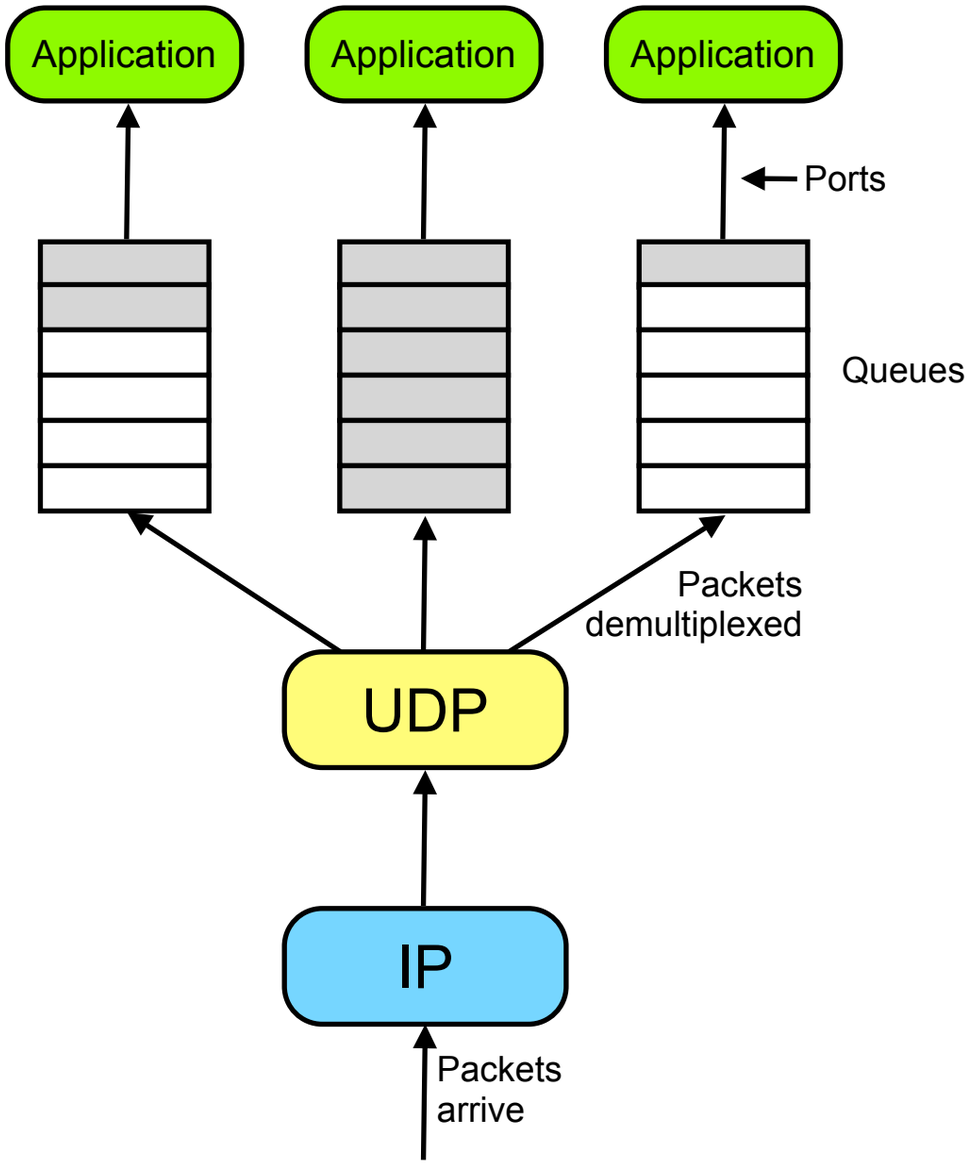
- IP network provides best effort service
  - Packets can be lost, duplicated, delayed, or re-ordered
- Transport isolates applications from the network
  - Demultiplexes traffic for different applications
  - Enhances network quality of service to offer appropriate reliability
  - Performs congestion control, adapts to network capacity

- Two common transport protocols in the Internet
  - UDP
  - TCP

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

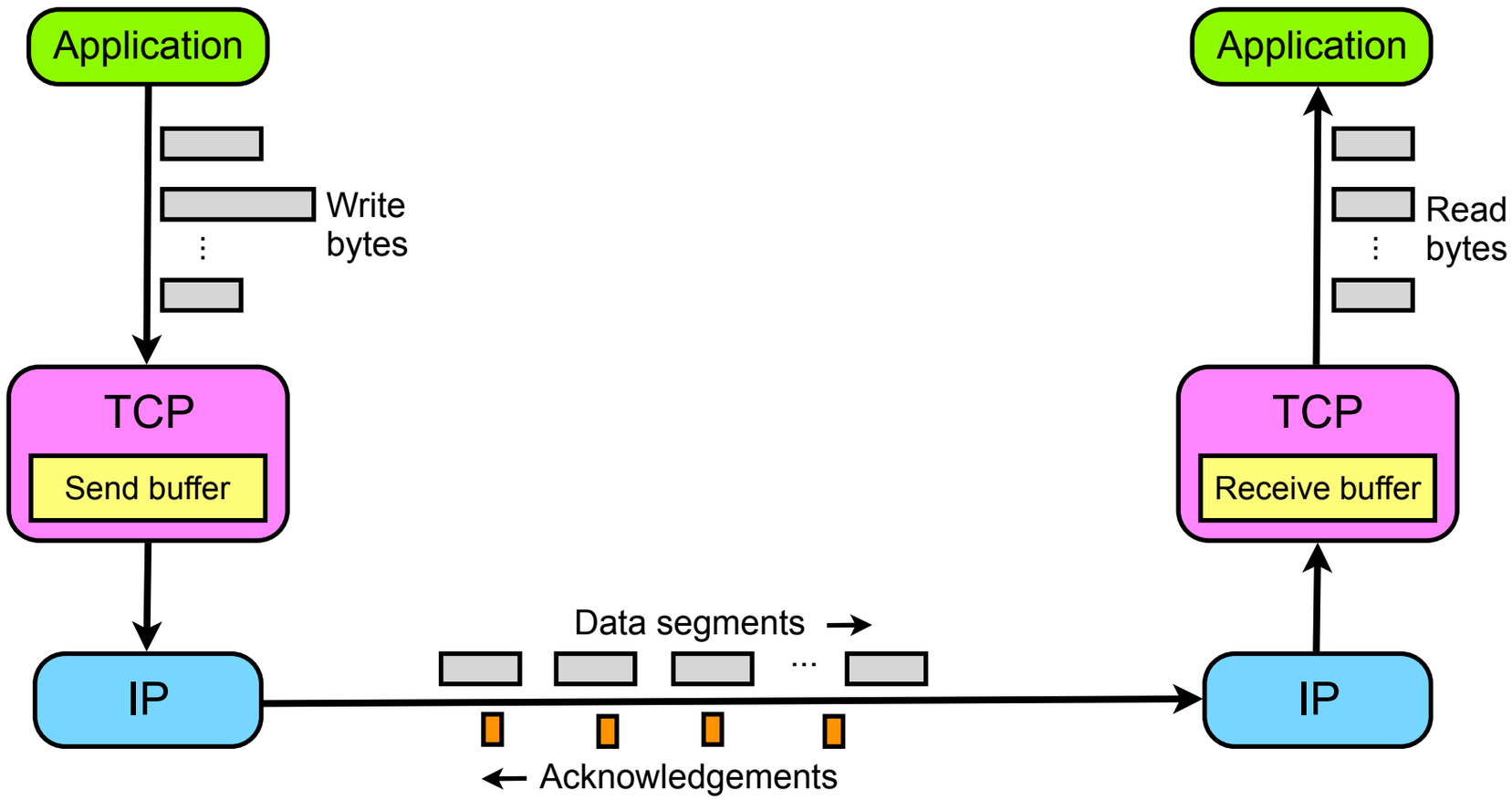
UDP

- Simplest transport protocol
- Exposes raw IP service to applications
  - Connectionless, best effort packet delivery: framed, but unreliable
  - No congestion control
  - Adds 16 bit port number to identify services
- Used by applications preferring timeliness over reliability
  - Voice-over-IP
  - Streaming video
  - Gaming
- Must be able to tolerate some loss of data
- Must be able to adapt to congestion in the application layer



Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

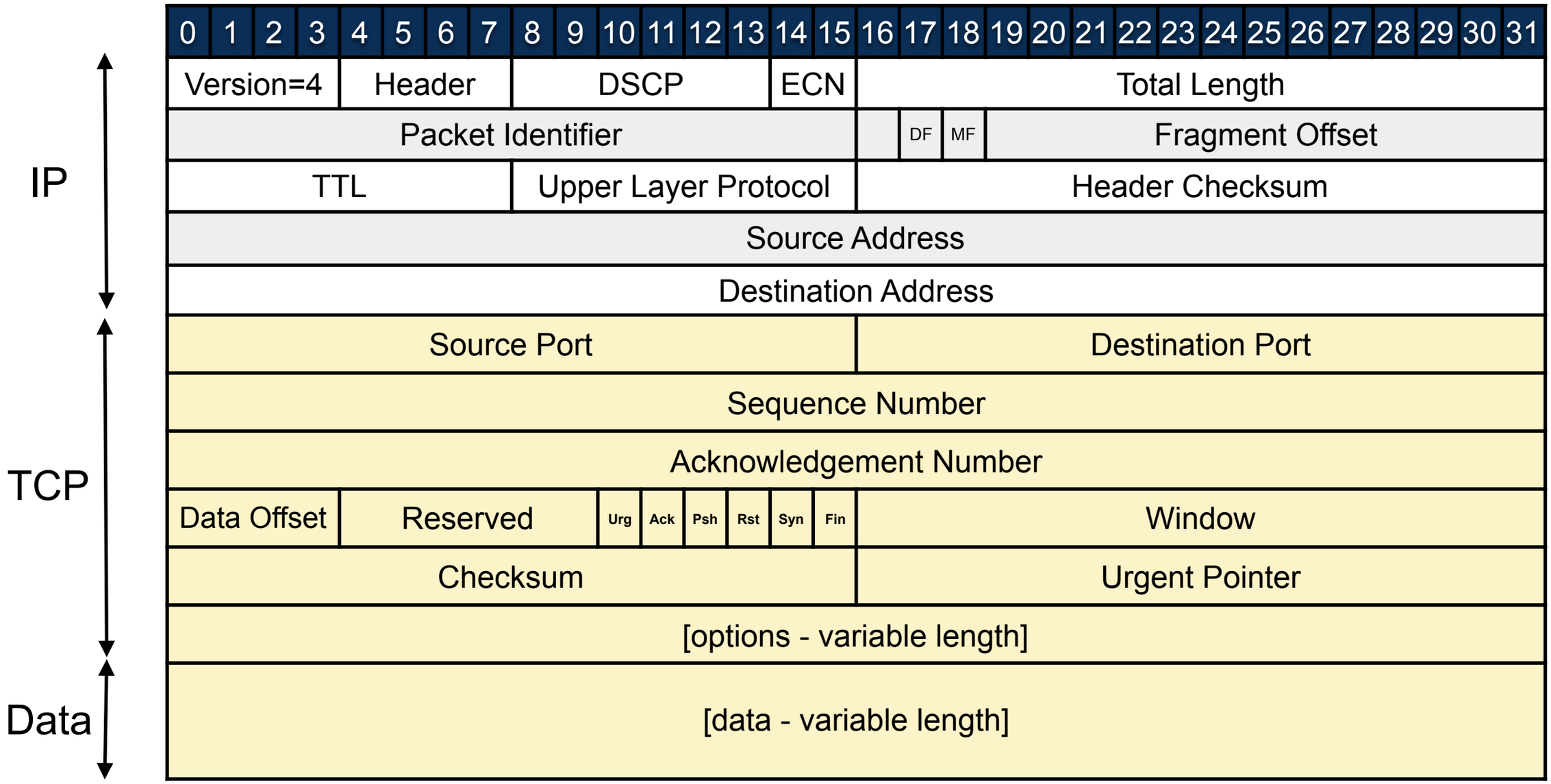
TCP



- Reliable, ordered, byte stream delivery service running over IP
  - Lost packets are retransmitted; ordering is restored; message boundaries not preserved
  - Adapts transmission rate to match network capacity – congestion control
  - Adds port number to identify services
- Used by applications that require reliability – default choice for most applications

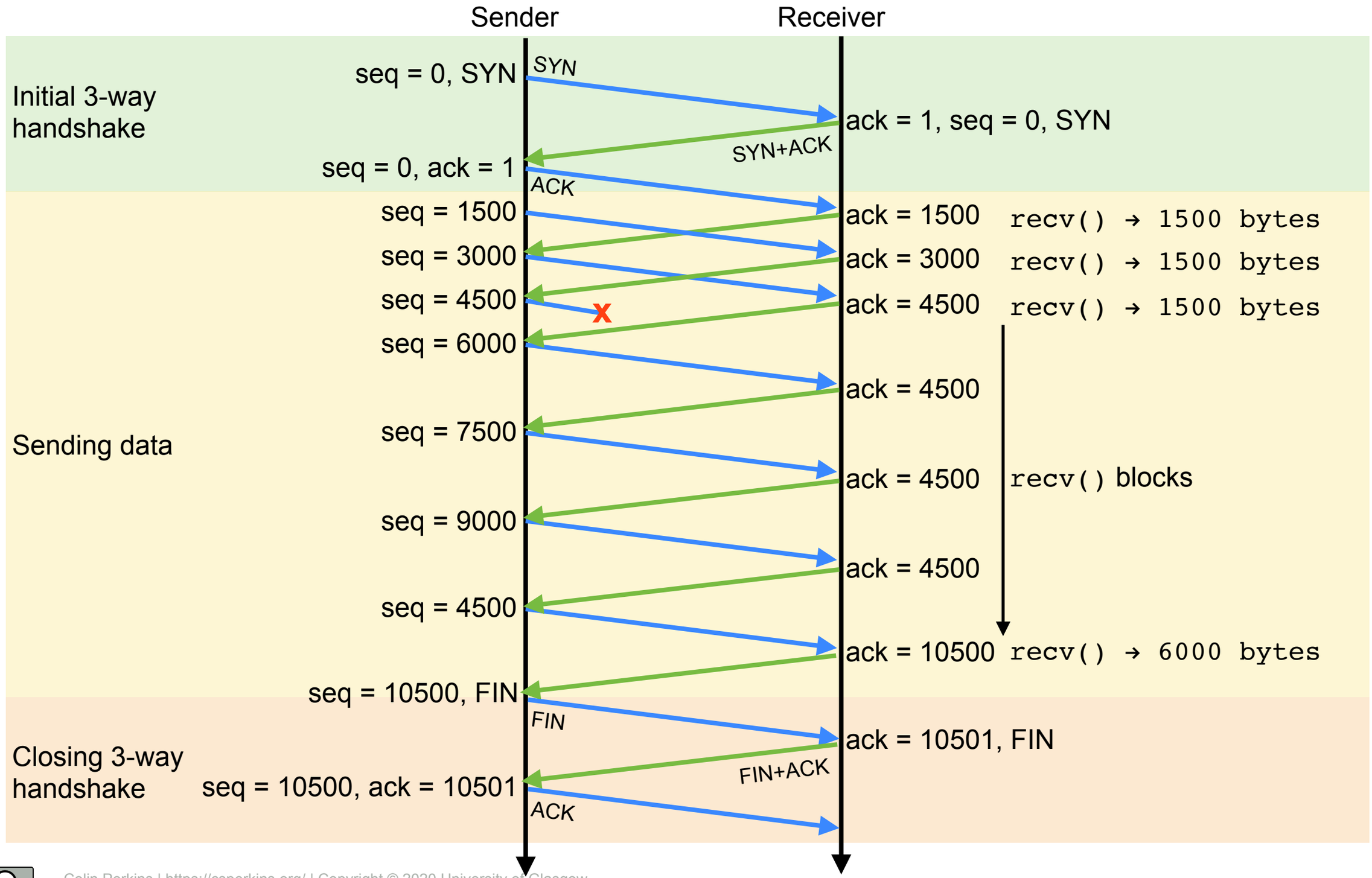
Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

TCP Packet Format



Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

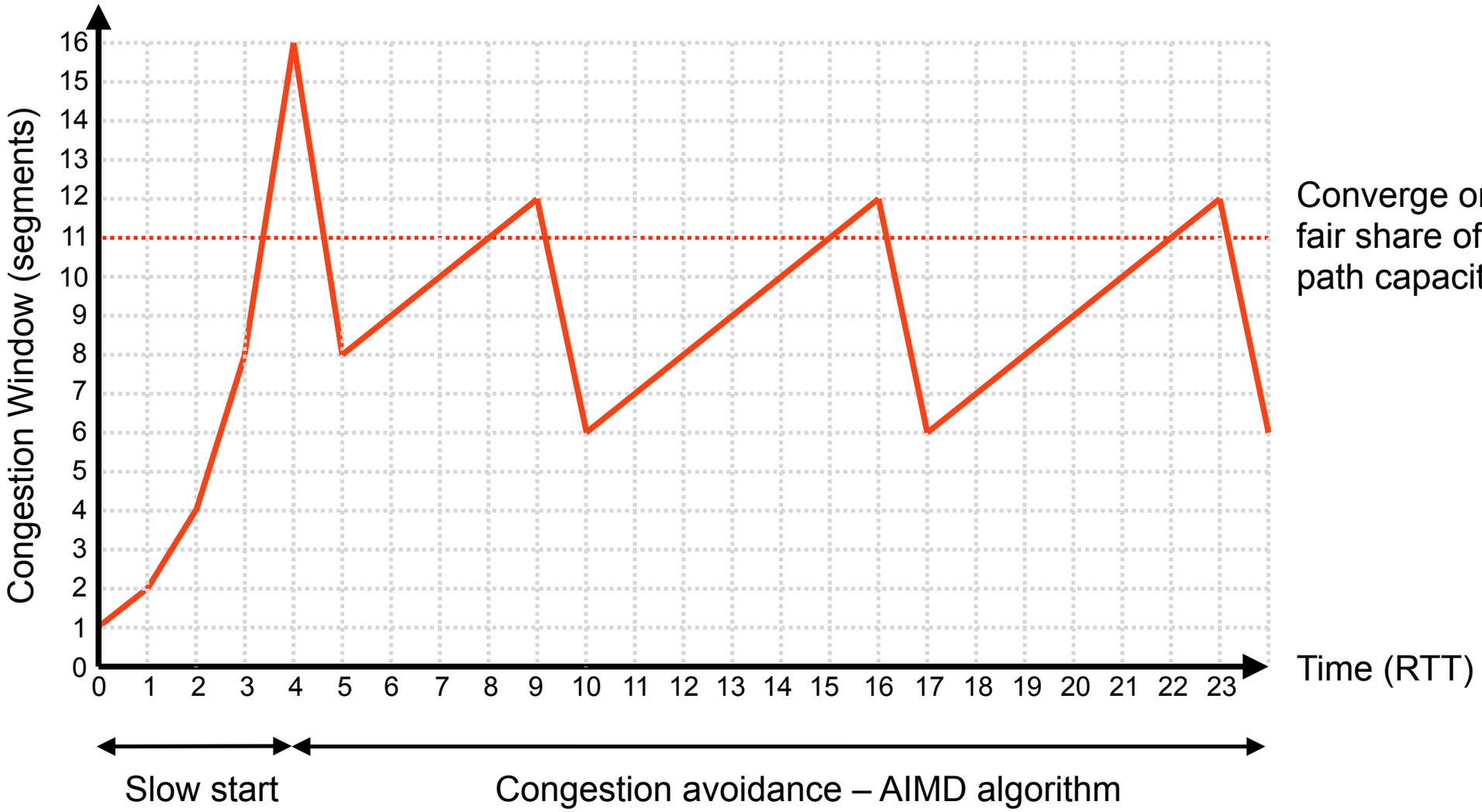
TCP Connection Timeline (Example)



Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

TCP Congestion Control

Typical evolution of TCP window, assuming W_{init} = 1



Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Higher Layer Protocols

- The OSI reference model defines three layers above the transport:
  - Session layer
  - Presentation layer
  - Application layer
- Internet architecture makes no clear distinction between these layers
  - Application protocols such as HTTP
- Goal – support application needs:
  - Manage transport layer connections
  - Name and locate application-level resources
  - Negotiate data formats, and perform format conversion if needed
  - Present data in appropriate manner
  - Implement application-level semantics

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Example: HTTP/1.1

```
C: GET /index.html HTTP/1.1
C: Accept-Language: en-gb
C: Accept-Encoding: gzip, deflate
C: Accept: text/html, text/plain
C: User-Agent: Safari/523.12.2
C: Cache-Control: max-age=0
C: Connection: keep-alive
C: Host: www.dcs.gla.ac.uk
C:

S: HTTP/1.1 200 OK
S: Date: Wed, 27 Feb 2008 22:44:25 GMT
S: Server: Apache/2.0.46 (Red Hat)
S: Last-Modified: Mon, 17 Nov 2003 08:06:50 GMT
S: Accept-Ranges: bytes
S: Content-Length: 3646
S: Connection: close
S: Content-Type: text/html; charset=UTF-8
S:
S: <HTML>
S: <HEAD>
S: <TITLE>Computing Science, University of Glasgow</TITLE>
S: ...remainder of page elided...
```

- HTTP/1.1 is a client-server protocol
  - Client opens TCP connection to server, sends **GET**
  - Server responds with requested page
  - Textual headers; binary response bodies
- Builds on reliable, ordered, TCP connections for data transfer, congestion control
- IP layer provides uniform connectivity
- Will explore in more detail later in the course

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

The Changing Internet

- Exhaustion of IPv4 address space
- Wireless and mobility
- Centralisation, CDNs, & Hypergiants
- Real-time and low-latency
- Innovation and ossification

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

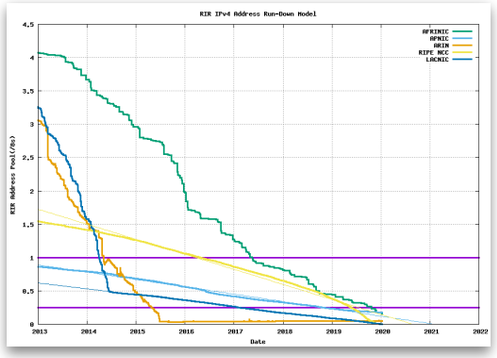
The Changing Internet

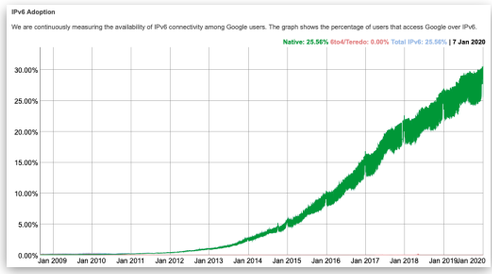
- Internet architecture makes certain assumptions:
  - Devices are uniquely addressable
  - Devices are not generally mobile
  - The network and services are decentralised
  - Best effort service provides sufficient quality
  - The network is trusted
  - Innovation can happen at the edges; the network is dumb
- Reasonable for the 1980s, when the IP architecture was defined – but the network has changed

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Changes in Addressing and Reachability

- Assumption: every network interface has a globally unique IP address
  - Devices are uniquely addressable, have uniform connectivity
  - In principle, any device can connect to any other
  - Policy restrictions enforced by firewalls
  - Privacy protected by changing address assignments
- IPv4 has insufficient addresses to support this model
- IPv6 deployment is slow; NAT is widespread



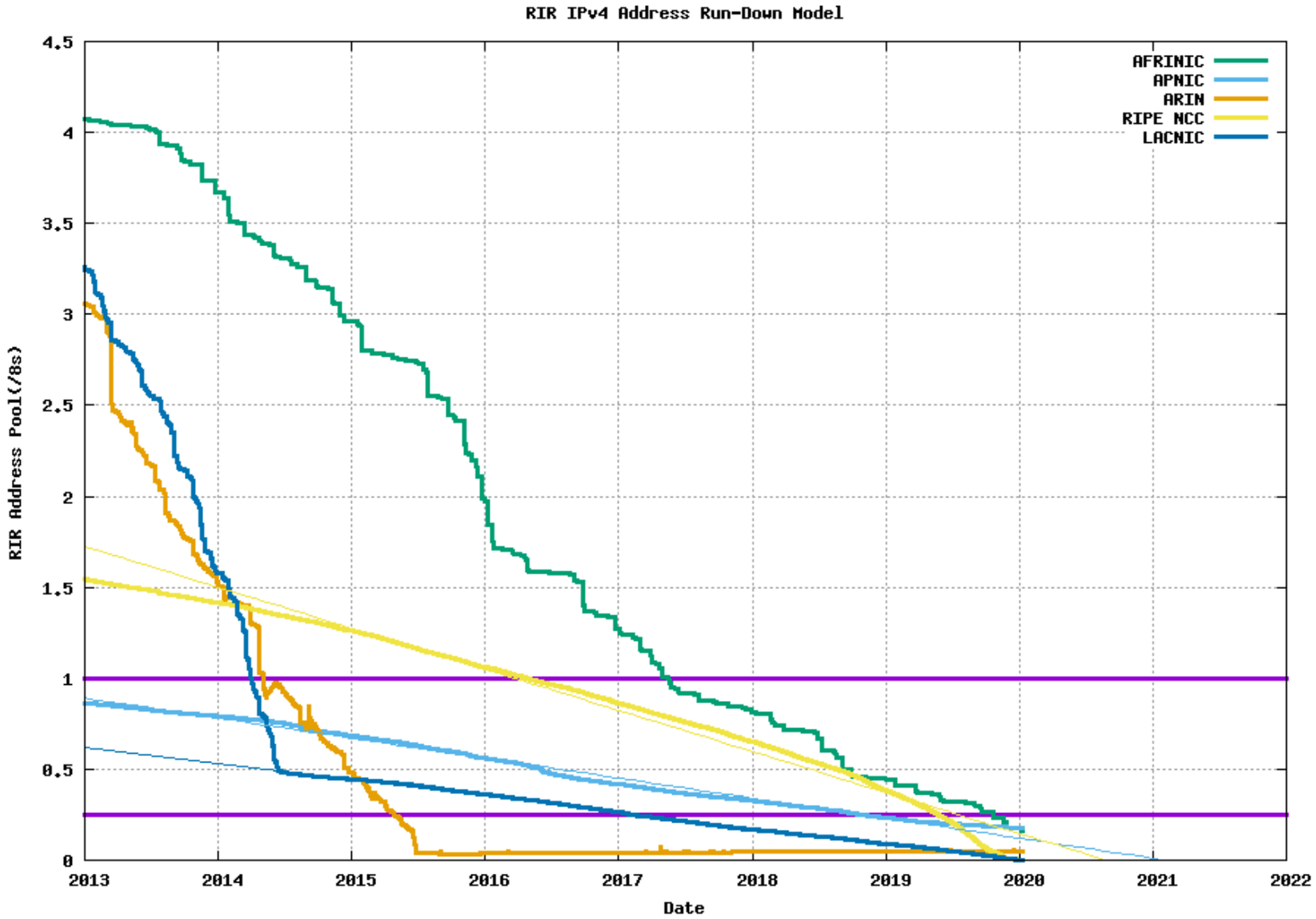


- Implication: connectivity becomes difficult
  - Dual-stack IPv4/IPv6 connection racing (“happy eyeballs”)
    - Complicates software design and implementation
  - NAT traversal for peer-to-peer applications
    - Complicates software design and implementation
    - Forces reliance on cloud services
  - Encourages centralisation

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

IPv4 Addresses Exhausted

- IANA assigned last IPv4 block to RIRs in February 2011
- RIRs have either exhausted their available IPv4 addresses, or soon will:



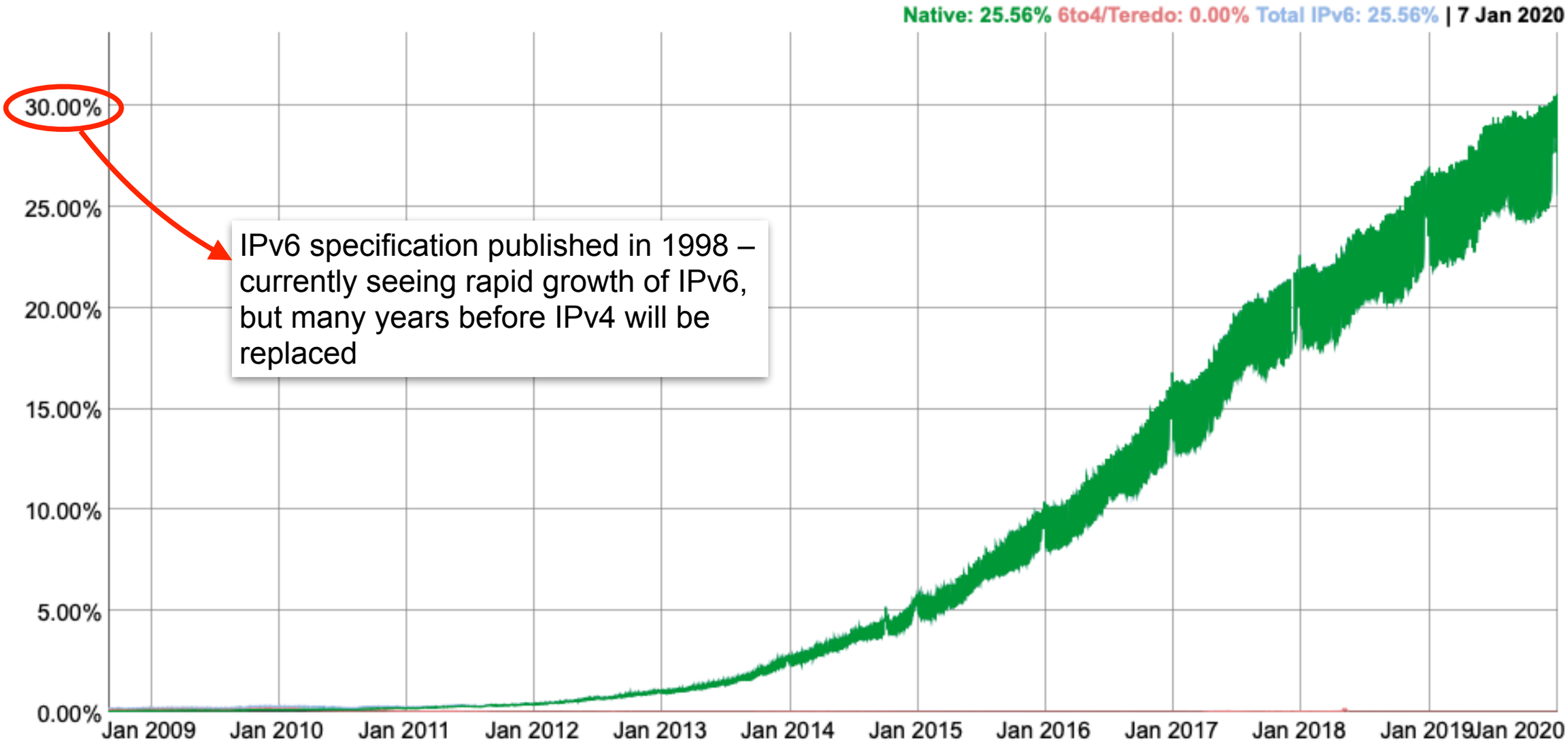
Source: Geoff Huston, APNIC
9 January 2020 http://ipv4.potaroo.net/

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

IPv6 Deployment is Slow

IPv6 Adoption

We are continuously measuring the availability of IPv6 connectivity among Google users. The graph shows the percentage of users that access Google over IPv6.

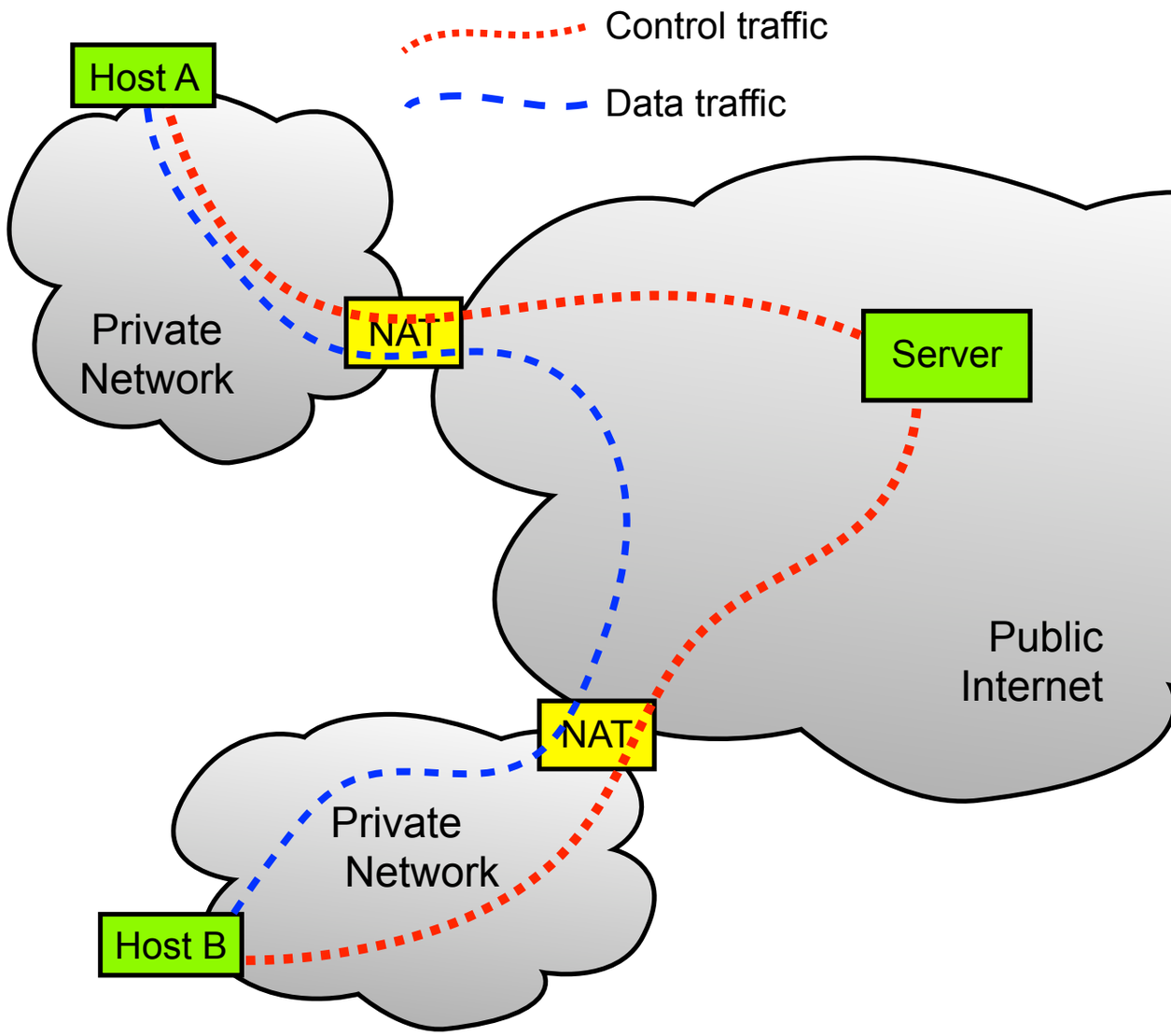


Source: Google, 9 January 2020
https://www.google.com/intl/en/ipv6/statistics.html

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Establishing Connectivity in the Modern Internet

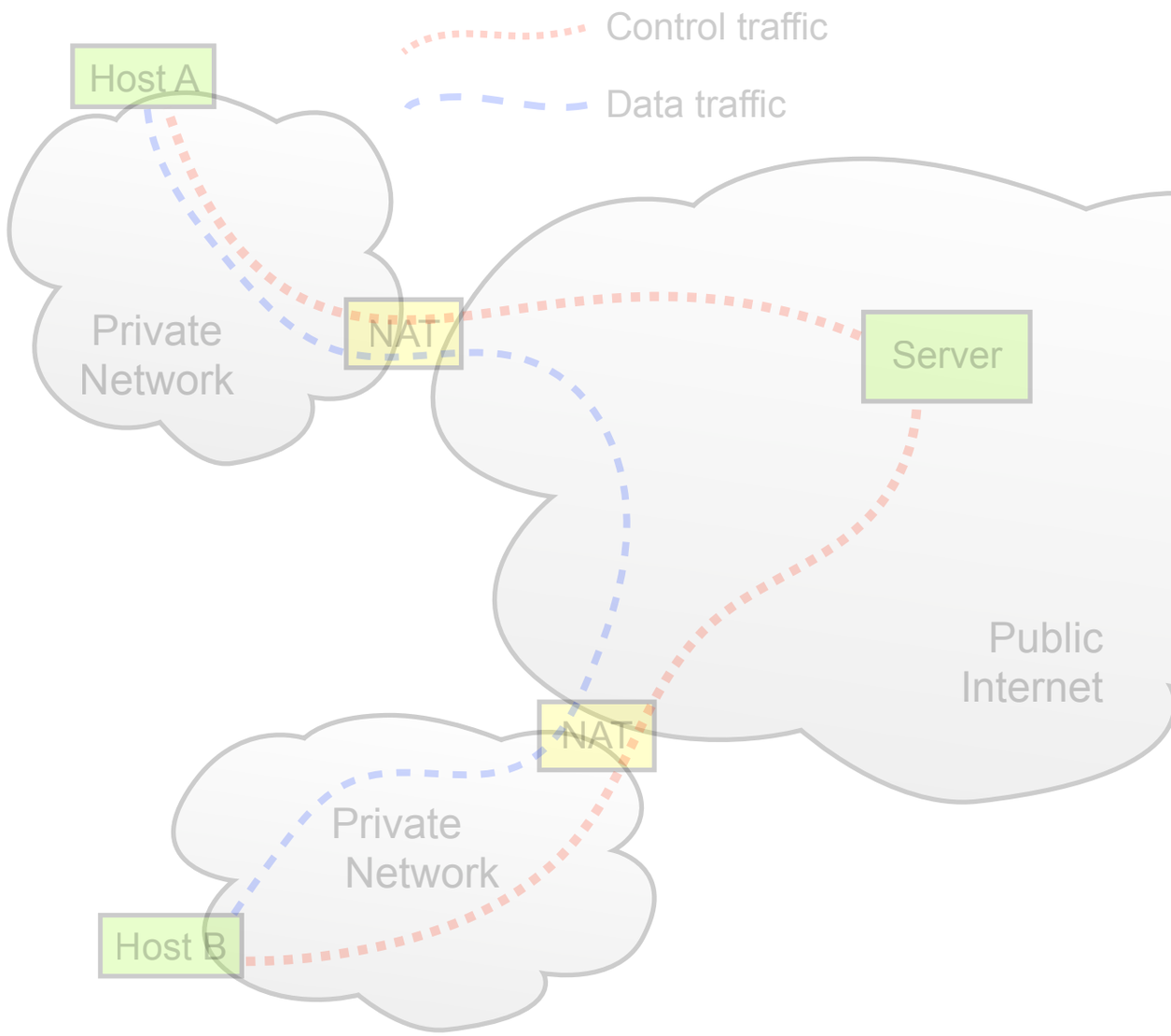
- NAT traversal → binding discovery, exchange, probing
  - STUN, TURN, and the ICE algorithm
  - Requires a server with public IP address for binding discovery; centralised point of control
  - Complex, slow, unreliable, wastes power, generates unnecessary traffic



Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Establishing Connectivity in the Modern Internet

- NAT traversal → binding discovery, exchange, probing
  - STUN, TURN, and the ICE algorithm
  - Requires a server with public IP address for binding discovery; centralised point of control
  - Complex, slow, unreliable, wastes power, generates unnecessary traffic
- Dual stack → connection racing
  - Some paths support IPv4, some IPv6; probing and connection timeouts slow for client-server applications
  - Race connections by probing in parallel – complex to implement, wastes resources



- NAT and dual stack operation complicates connection establishment; encourages centralisation onto cloud services

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

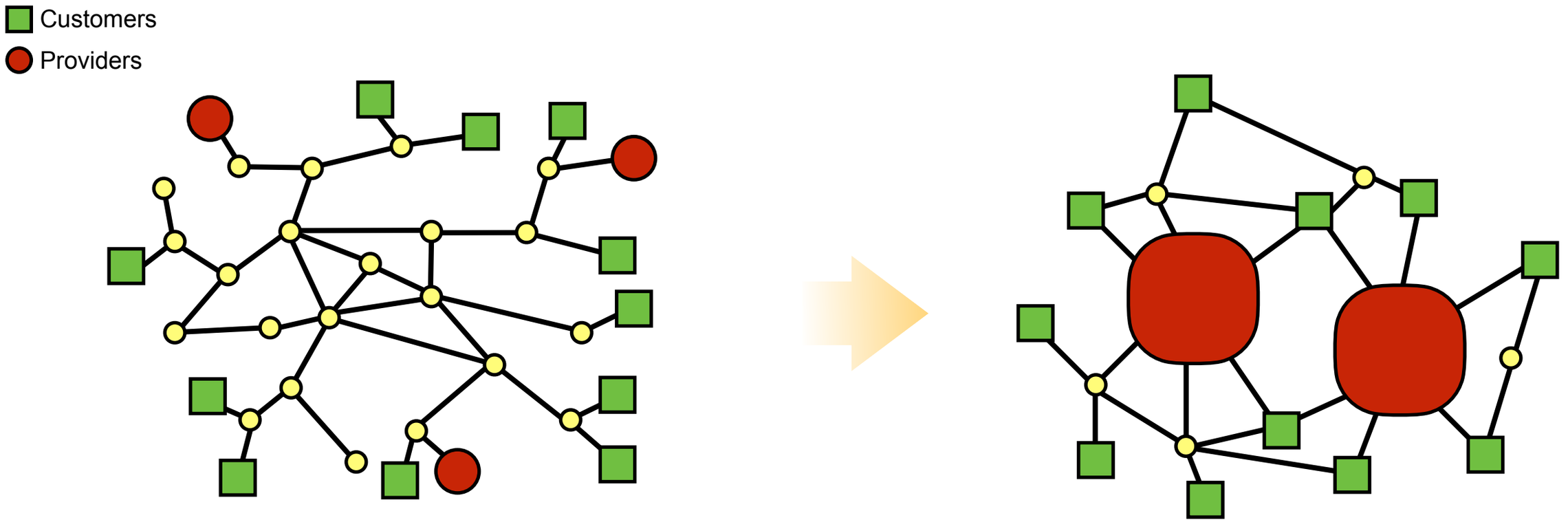
Increasing Mobility

- IP addresses intended to encode location in the network
- Complicates addressing and reachability of mobile devices:
  - Devices change their IP address each time they move:
    - TCP connections must be re-established after each move
    - UDP packets must be redirected after each move
    - Complex signalling and connection management if devices move frequently
  - Devices retain their IP address when moving
    - Routing scales because addresses encode location and can be aggregated into prefixes
    - Routers would need to store a route for each device if mobile devices keep their IP address

- Ubiquitous mobile devices – *i.e., smartphones* – disrupt routing and connection management

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

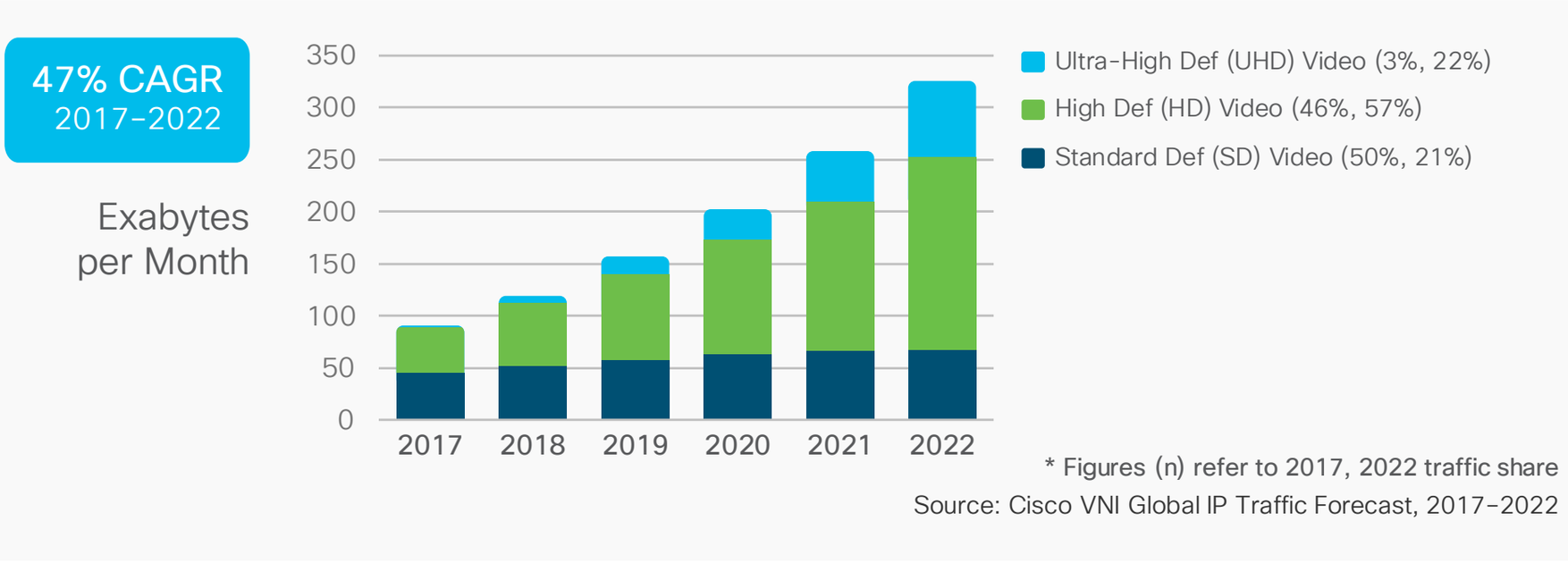
Hypergiants and Centralisation



- Internet topology is flattening, becoming increasingly centralised
- Transit network ecosystem being replaced by direct connections from “eyeball” networks to content providers – Google, Facebook, Amazon, Akamai, …
- Implications for network neutrality, competition, innovation

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Supporting Real-time Traffic



* Figures (n) refer to 2017, 2022 traffic share

Source: Cisco VNI Global IP Traffic Forecast, 2017–2022

- Video streaming dominates Internet traffic, and is growing >40% per year
  - Best effort IP service not well suited to real-time traffic
  - Packet loss → quality impairments or increased delay
  - Driving centralisation and direct CDN connections → easier to manage quality
  - Driving TCP congestion changes and TCP replacements → BBR algorithm, QUIC; low-latency
- Interactive real-time media has even stricter constraints
  - WebRTC, Skype, VoIP, gaming, … → non-TCP transport to lower latency

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Pervasive Monitoring and Trust in the Network



Edward Snowden

Internet Engineering Task Force (IETF)
Request for Comments: 7258
BCP: 188
Category: Best Current Practice
ISSN: 2070-1721

S. Farrell
Trinity College Dublin
H. Tschofenig
ARM Ltd.

May 2014

Pervasive Monitoring Is an Attack

Abstract

Pervasive monitoring is a technical attack that should be mitigated in the design of IETF protocols, where possible.

Status of This Memo

This memo documents an Internet Best Current Practice.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on BCPs is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7258.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

https://www.rfc-editor.org/rfc/rfc7258.html

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Innovation in the Network

- Can we still change the network protocols?
  - NATs, firewalls, middleboxes → TCP is *hard* to change now
  - Transport encryption, tunnel over UDP → QUIC? Pervasive encryption to defeat pervasive monitoring and surveillance?
  - Protocol ossification is a real concern
  - Is the model smart edges and a dumb network appropriate?
- Can new startups compete with the hypergiants?
  - Several forces pushing towards centralisation – can we evolve the network to counter these?
  - Can we enable a decentralised network?



Is it Still Possible to Extend TCP?

Michio Honda*, Yoshifumi Nishida*, Costin Raiciu†, Adam Greenhalgh‡, Mark Handley‡, Hideyuki Tokuda*
Keio University*, Universitatea Politehnica Bucuresti†, University College London‡
{micchie,nishida}@sfc.wide.ad.jp, costin.raiciu@cs.pub.ro
{a.greenhalgh,m.handley}@cs.ucl.ac.uk, hxt@ht.sfc.keio.ac.jp

ABSTRACT

We've known for a while that the Internet has ossified as a result of the race to optimize existing applications or enhance security. NATs, performance-enhancing-proxies, firewalls and traffic normalizers are only a few of the middleboxes that are deployed in the network and look beyond the IP header to do their job. IP itself can't be extended because "IP options are not an option" [10]. Is the same true for TCP?

In this paper we develop a measurement methodology for evaluating middlebox behavior relating to TCP extensions and present the results of measurements conducted from multiple vantage points. The short answer is that we can still extend TCP, but extensions' design is very constrained as it needs to take into account prevalent middlebox behaviors. For instance, absolute sequence numbers cannot be embedded in options, as middleboxes can rewrite ISN and preserve undefined options. Sequence numbering also must be consistent for a TCP connection, because many middleboxes only allow through contiguous flows.

We used these findings to analyze three proposed extensions to TCP. We find that MPTCP is likely to work correctly in the Internet or fallback to regular TCP. TcpCrypt seems ready to be deployed, however it is fragile if resegmentation does happen—for instance with hardware offload. Finally, TCP extended options in its current form is not safe to deploy.

Categories and Subject Descriptors

C.2.2 [**Computer-communication Networks**]: Network Protocols; C.2.6 [**Computer-communication Networks**]: Internetworking

General Terms

Measurement, Design, Experimentation, Standardization

Keywords

Middleboxes, Measurements, TCP, Protocol design

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
IMC'11, November 2–4, 2011, Berlin, Germany.
Copyright 2011 ACM 978-1-4503-1013-0/11/11 ...$10.00.

1. INTRODUCTION

The Internet was designed to be extensible; routers only care about IP headers, not what the packets contain, and protocols such as IP and TCP were designed with options fields that could be used to add additional functionality. The great virtue of the Internet was always that it was *stupid*; it did no task especially well, but it was extremely flexible and general, allowing a proliferation of protocols and applications that the original designers could never have foreseen.

Unfortunately the Internet, as it is deployed, is no longer the Internet as it was designed. IP options have been unusable for twenty years[10] as they cause routers to process packets on their slow path. Above IP, the Internet has benefited (or suffered, depending on your viewpoint) from decades of optimizations and security enhancements. To improve performance [2, 7, 18, 3], reduce security exposure [15, 29], enhance control, and work around address space shortages [22], the Internet has experienced an invasion of middleboxes that *do* care about what the packets contain, and perform processing at layer 4 or higher *within* the network.

The problem now faced by designers of new protocols is that there is no longer a well defined or understood way to extend network functionality, short of implementing everything over HTTP[25]. Recently we have been working on adding both multipath support[11] and native encryption[5] to TCP. The obvious way to do this, in both cases, is to use TCP options. In the case of multipath, we would also like to stripe data across more than one path. At the end systems, the protocol design issues were mostly conventional. However, it became increasingly clear that no one, not the IETF, not the network operators, and not the OS vendors, knew what will and what will not pass through all the middleboxes as they are currently deployed and configured. Will TCP options pass unchanged? If the sequence space has holes, what happens? If a retransmission has different data than the original, which arrives? Are TCP segments coalesced or split? These and many more questions are crucial to answer if protocol designers are to extend TCP in a deployable way. Or have we already lost the ability to extend TCP, just like we did two decades ago for IP?

In this paper we present the results from a measurement study conducted from 142 networks in 24 countries, including cellular, WiFi and wired networks, public and private networks, residential, commercial and academic networks. We actively probe the network to elicit middlebox responses that violate the end-to-end transparency of the original Internet architecture. We focus on TCP, not only because it is by far the most widely used transport protocol, but also because while it is known that many middleboxes modify TCP behavior [6], it is not known how prevalent such middleboxes are, nor precisely what the *emergent behavior* is with TCP extensions that were unforeseen by the middlebox designers.

Honda, M., Nishida, Y., Raiciu, C., Greenhalgh, A., Handley, M., and Tokuda, H. Is it still possible to extend TCP? Proc. ACM Internet Measurement Conference, Berlin, Nov. 2011. https://dx.doi.org/10.1145/2068816.2068834

CC BY ND

Can the network evolve to address these challenges?

- How to establish connections in a fragmented network?
- How can encryption protect against pervasive monitoring and prevent transport ossification?
- How can we reduce latency and support real-time and interactive content?
- How can we adapt to the vagaries of wireless networks?
- How can we identify and distribute content? How can we manage the tussle for control of the DNS and naming?
- How can we manage interdomain routing to efficiently delivery content?
- How can we re-decentralise the network?

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow