University of Glasgow

Networks and Internet Routing

Networked Systems (H) – Lecture 9

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

- Networks and Internet Routing:
  - Content distribution networks
  - Inter-domain routing
  - Routing security
  - Intra-domain routing
- Wrap-up

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

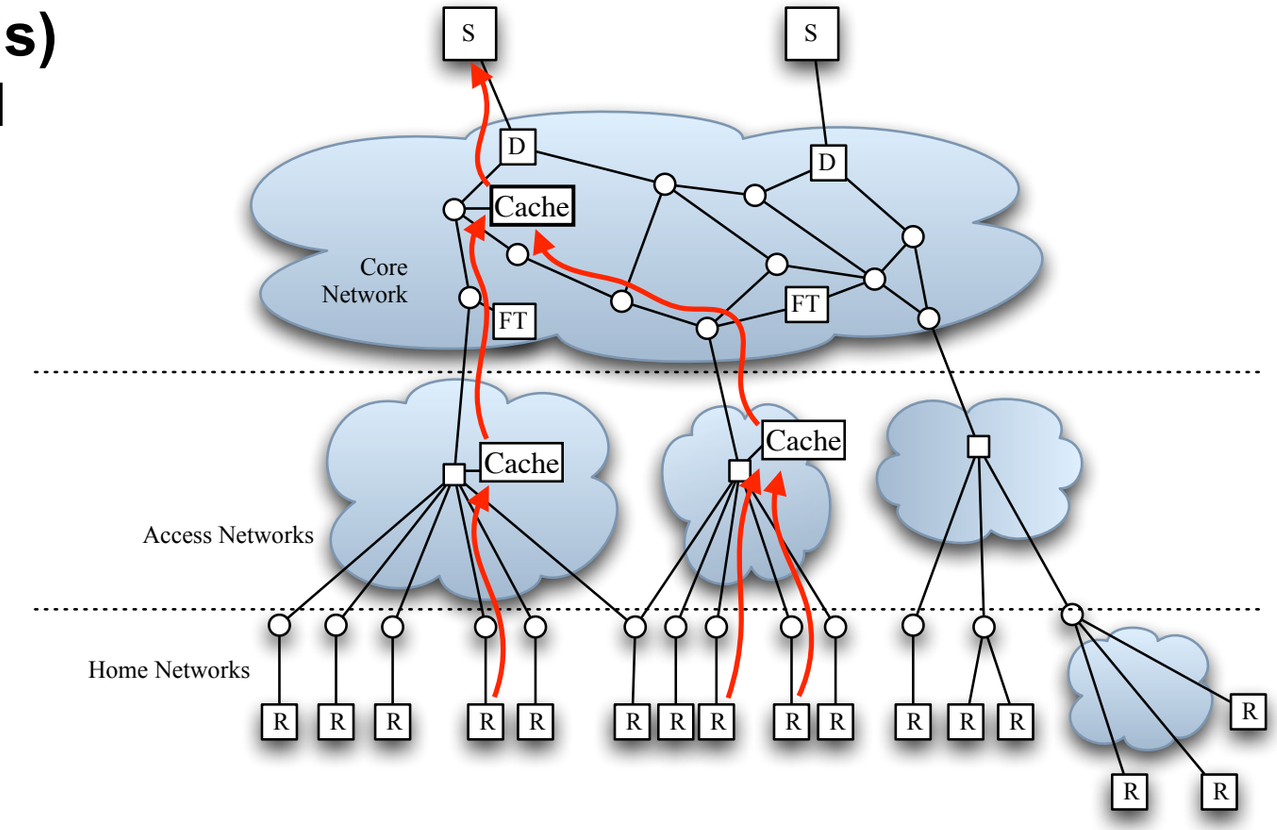
Content Distribution Networks (CDNs)

- Load balancing and latency reduction
- Implementation using DNS
- Implementation using anycast routing

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

Content Distribution Networks (CDNs)

- **Content distribution networks (CDNs)** provide scalability, load balancing, and low latency
  - Distribute web caches around the world, in ISP edge networks and data centres
  - Where possible, direct HTTP requests to local caches rather than the original site
    - Reduces load on the main servers
    - Reduces latency to respond to requests
    - Reduces chances of successful denial-of-service attack
- Many commercial CDNs available
- Many large organisations run their own
  - **Hypergiants** such as Google, Facebook, Netflix, …

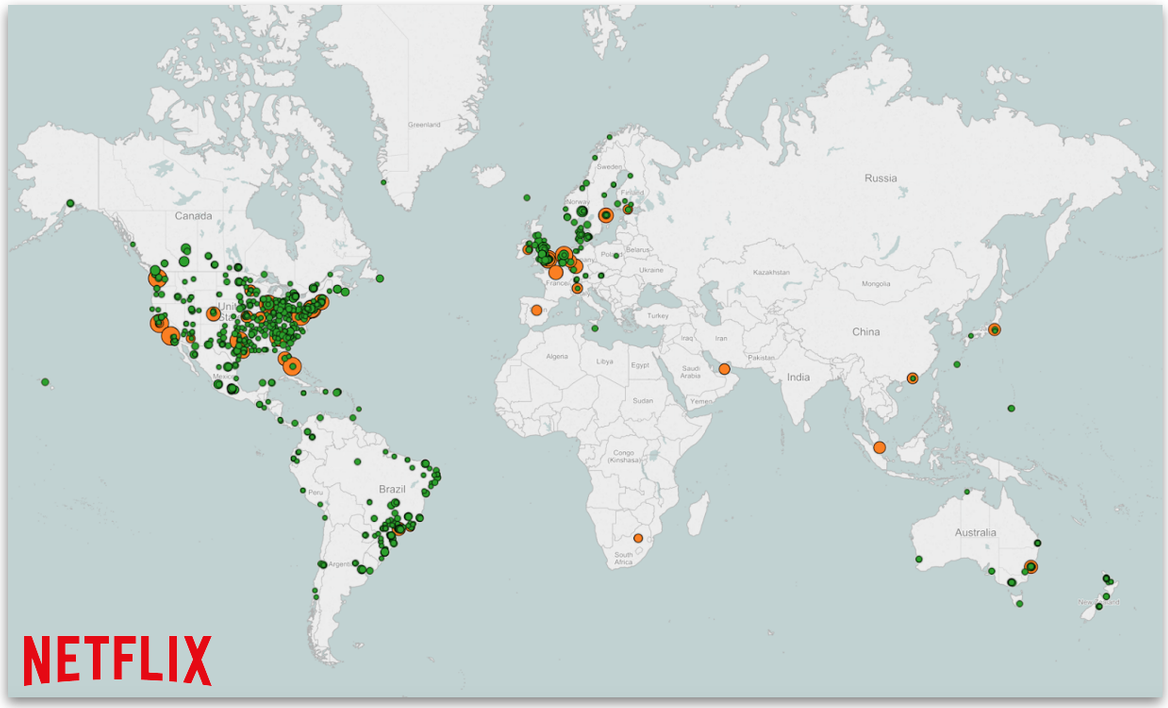


Akamai CLOUDFLARE fastly

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

Using CDNs to Distribute Load

- CDNs distribute load by locating web caches around the world and answering most requests from a local cache
- Requires extensive investment and large-scale cooperation with ISPs, Internet Exchange Points, etc.
- Mutual benefit for an ISP to host the servers for a CDN
  - Reduces the load on the ISP's network
    - One copy of the content sent to cache; the cache then distributes many times
    - Avoids overload of link from ISP to outside world
    - e.g., Netflix distribute "tens of terabits per second" of video – not possible from a single data centre; needs local fanout
  - Increases the reach and robustness of the CDN



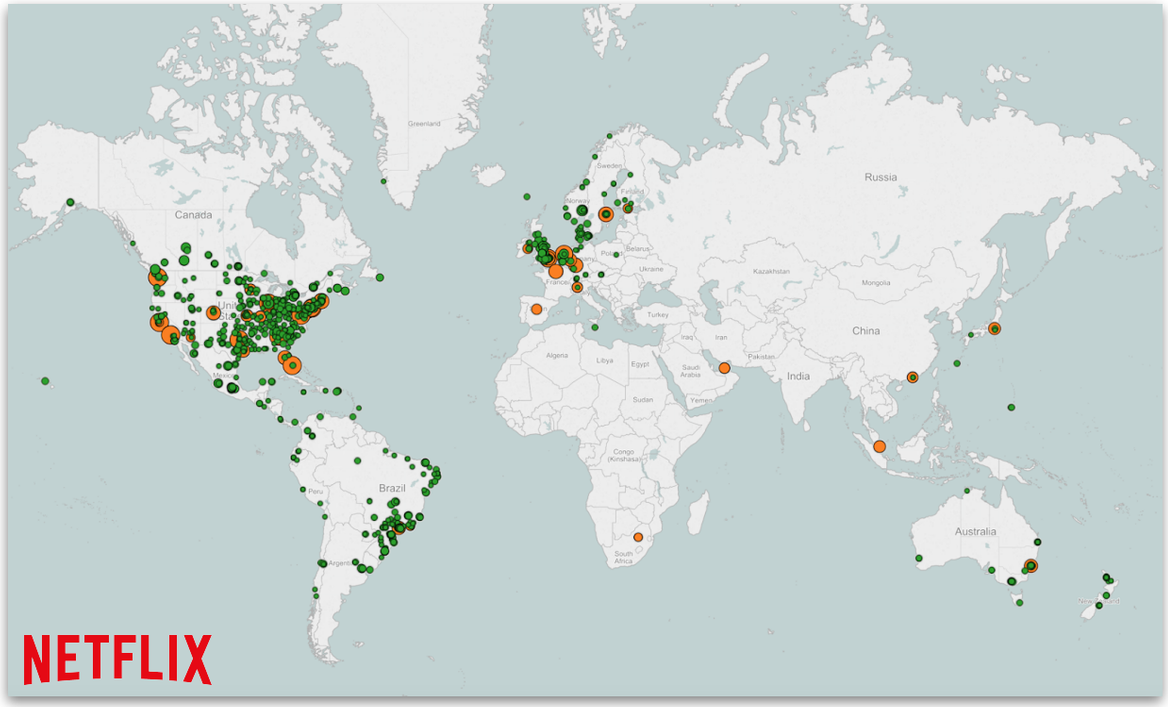
Netflix Open Connect CDN locations (source: https://media.netflix.com/en/company-blog/how-netflix-works-with-isps-around-the-globe-to-deliver-a-great-viewing-experience)

Akamai "more than 240,000 servers in over 130 countries"

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

Using CDNs to reduce latency

- Key CDN benefit is reducing latency → content is delivered from nearby caches
  - e.g., requests to a US-based service are answered from a CDN cache in Europe, rather than having to cross long-distance path to the US
- Requires global distribution of CDN proxy caches
  - Are CDNs effectively serving the entire world?
  - Providing effective Internet access to developing regions requires more than just connectivity; also needs data centres and infrastructure to host the CDN nodes
- Effective for cacheable static content – video, software updates, images – may need edge compute infrastructure to support other applications

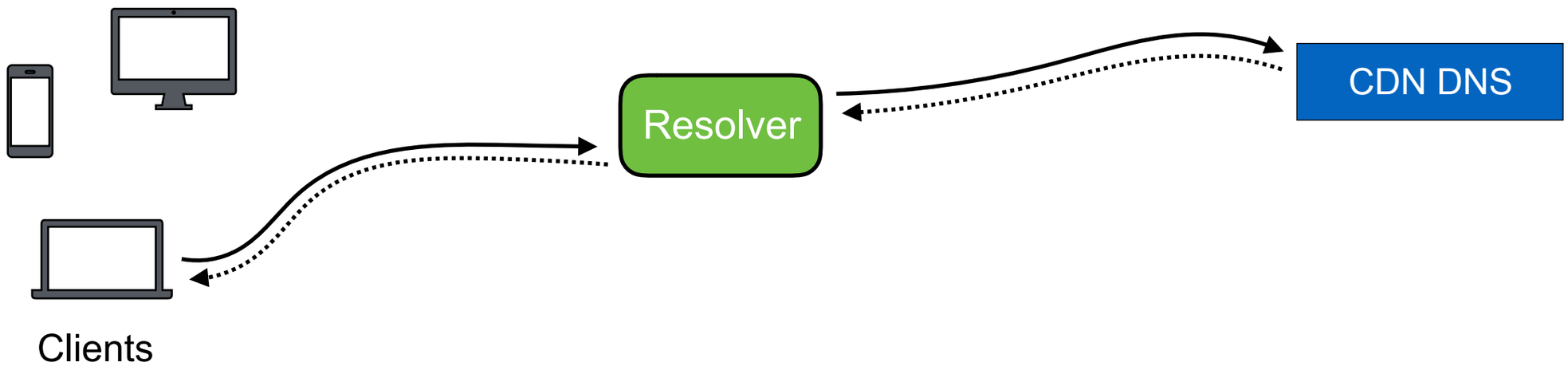


Netflix Open Connect CDN locations (source: https://media.netflix.com/en/company-blog/how-netflix-works-with-isps-around-the-globe-to-deliver-a-great-viewing-experience)

CC BY ND

Locating the Nearest CDN Node using DNS

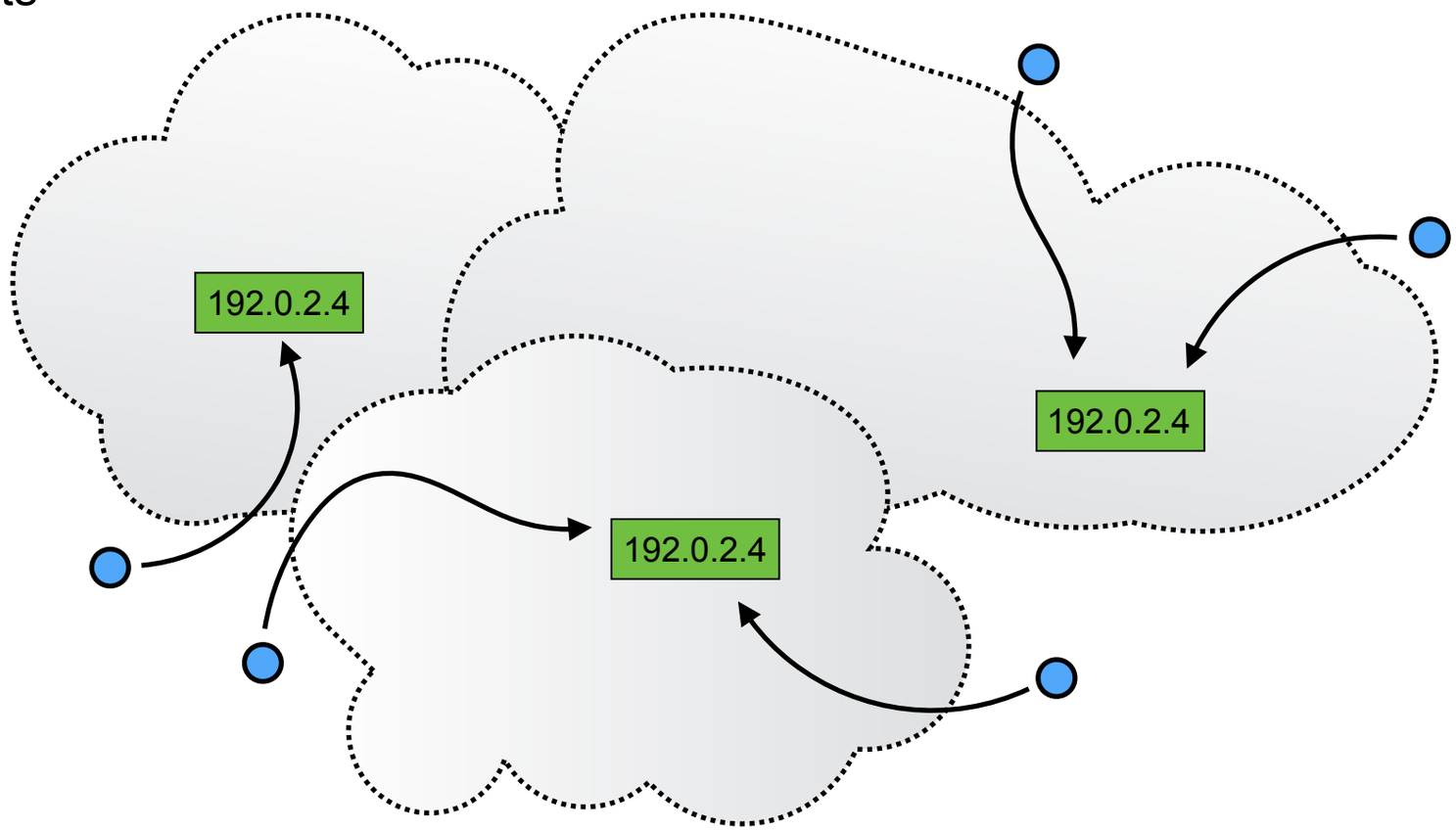
- Locate nearest CDN node using the DNS:
  - Each resource hosted by the CDN has a unique DNS name
    - e.g., rather than **`https://example.com/images/kitten.jpg`** the image might be hosted by the CDN as **`https://9BC1C10B7947A890B9.cdn-example.com/kitten.jpg`**
    - Each resource is given a different domain name
  - DNS server for the CDN returns different A or AAAA records for a name, depending on where it is requested from and what CDN caches have that data – directs to most appropriate cache
    - Based on IP address of the resolver making the query
      - DNS client subnet extension [RFC7871] for remote resolvers – e.g., Google public DNS on 8.8.8.8
    - Doesn't have to be particularly accurate: it's trying to figure out that you're in the UK rather than the US, and direct you to the data centre in London rather than New York
    - Allows very fine-grained control, but puts high load on DNS



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

Locating the Nearest CDN Node using Anycast Routing

- Locate nearest CDN node using **anycast routing**
  - Each resource hosted by the CDN has a unique filename
    - e.g., rather than **`https://example.com/images/kitten.jpg`** the image might be hosted by the CDN as **`https://cache.cdn-example.com/9BC1C10B7947A890B9.jpg`**
    - The DNS name always maps to the same IP address – that IP address is a load balancer at the entrance to a data centre
  - **The CDN uses multiple data centres – all using the same IP addresses**
    - Each data centre advertises its addresses via BGP
    - Inter-domain Internet routing will ensure traffic goes to the closest data centre to source



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

Inter-domain Routing

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

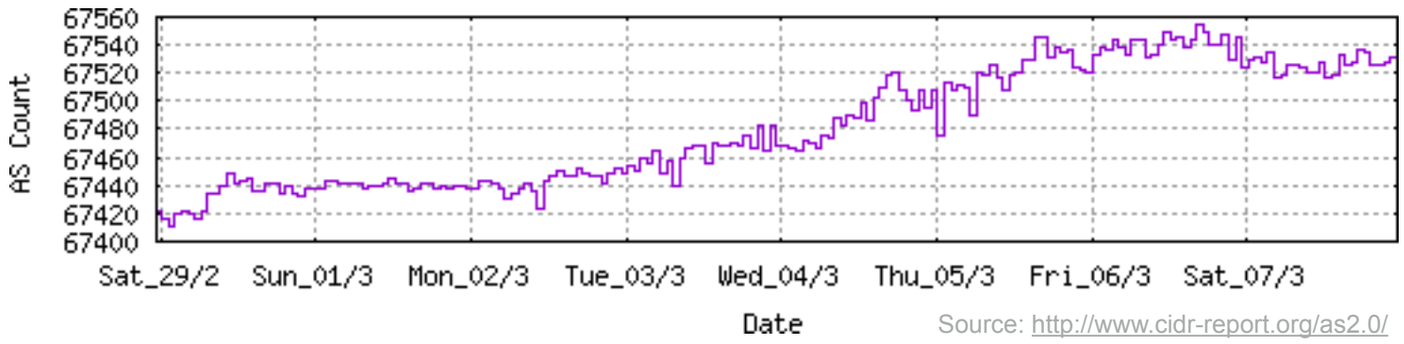
Inter-domain Routing

- The Internet is a network of networks
  - Each network is an **Autonomous System (AS)**
  - Each network is a separate routing domain

- **Inter-domain routing** is the problem of finding the best path from the source network to the destination network
  - Treat each network as a node on the routing graph (the “AS topology graph”)
  - Treat connections between networks as edges in the graph
  - What is the best set of networks to choose to get from source to destination

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

Autonomous Systems



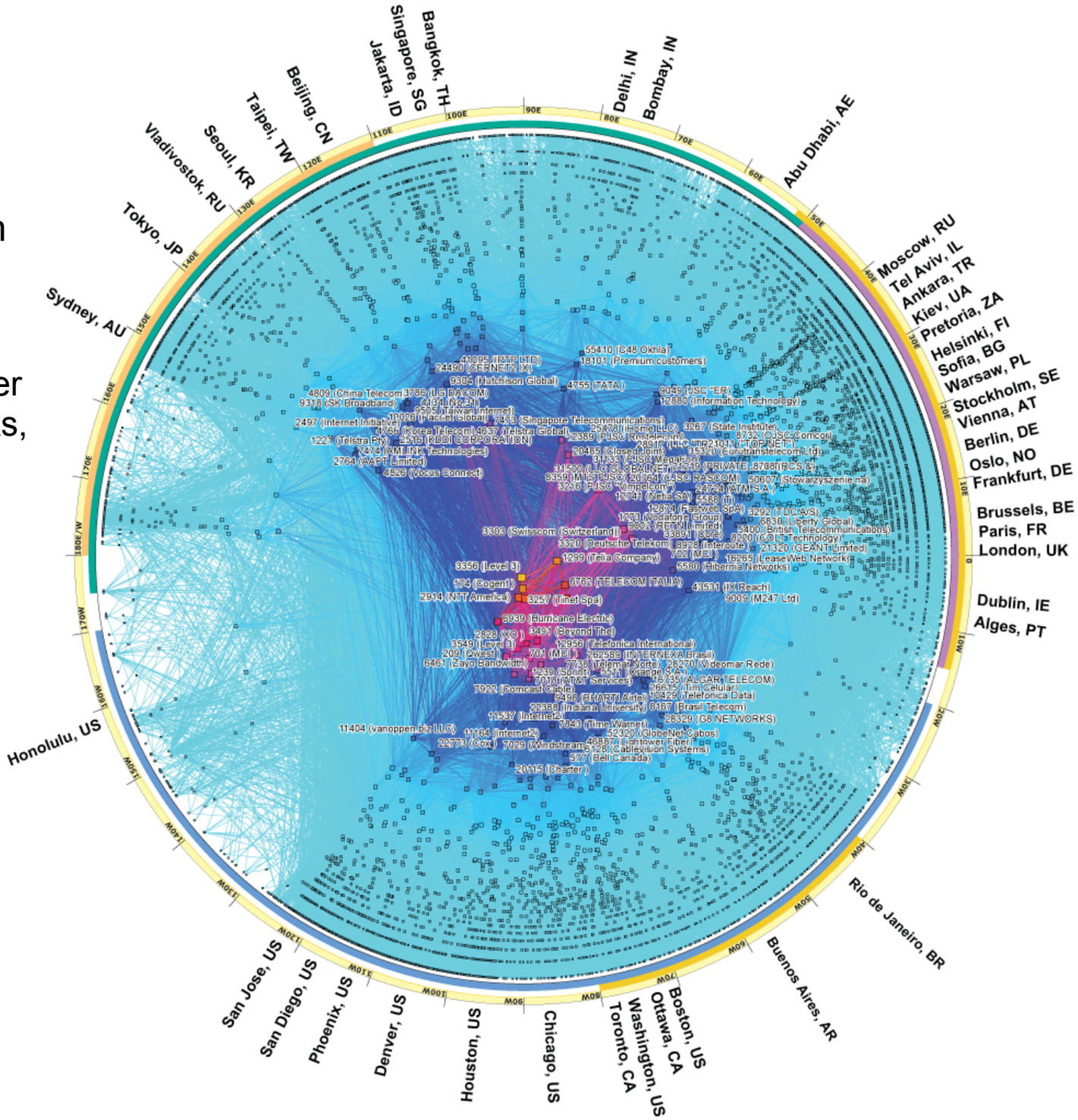
Source: http://www.cidr-report.org/as2.0/

- An **Autonomous System (AS)** is an independently operated network
  - An Internet service provider, or other organisation, that operates a network and wants to participate in interdomain routing
  - Some organisations operate more than one AS
    - For ease of administration; due to company mergers; etc.
  - Each AS is identified by a unique number, allocated by the RIR
    - 105,470 AS numbers allocated as of 7 March 2020; of these 67,531 are advertised in BGP
    - http://www.potaroo.net/tools/asn32/ – current state of allocation of AS numbers
    - http://bgp.potaroo.net/cidr/autnums.html – complete list AS numbers and names, with links to details
    - http://www.cidr-report.org/as2.0/ – classless interdomain routing (CIDR) report

CC BY ND

AS-level Internet Topology Graph (IPv4)

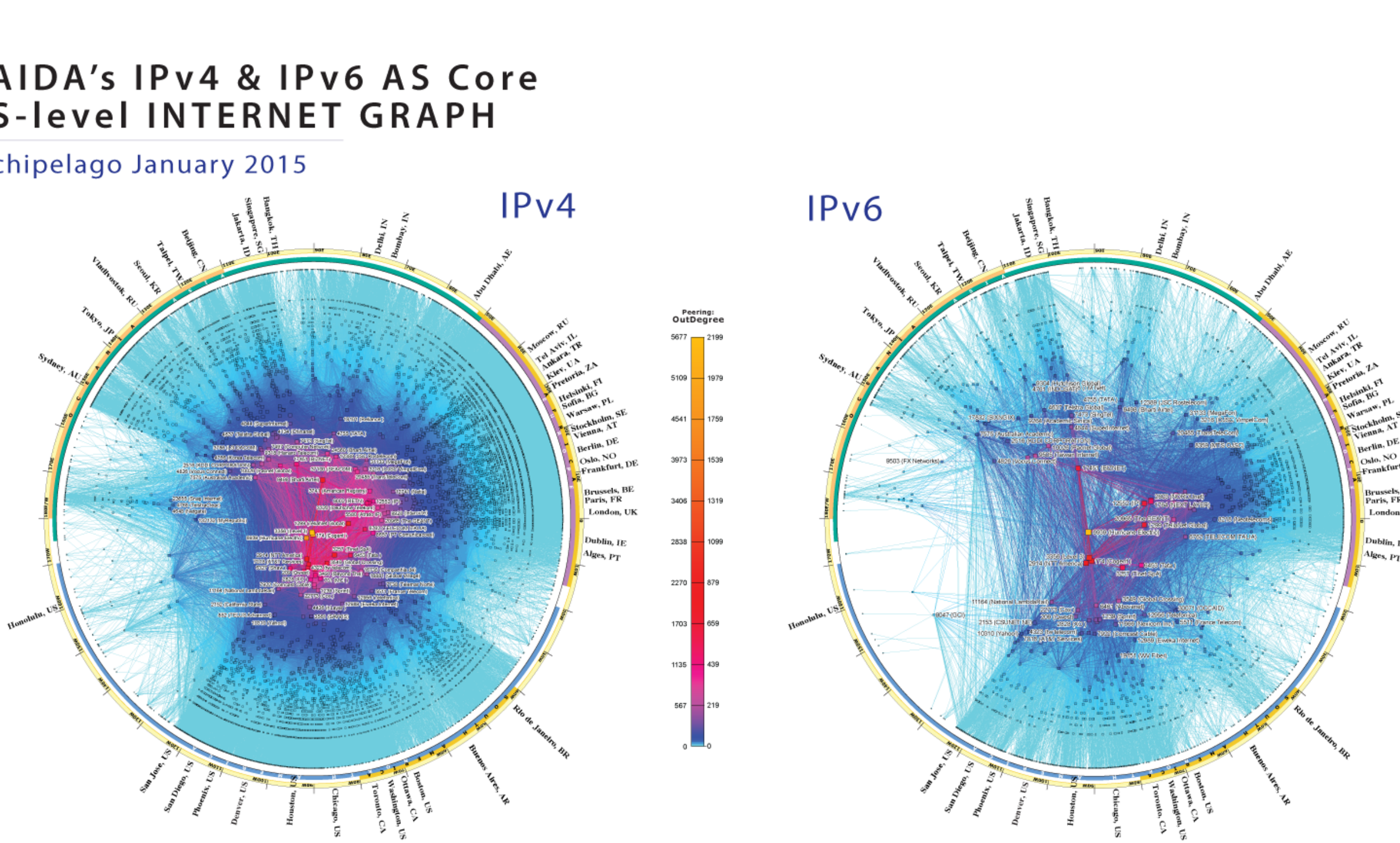
- AS-level topology – what are the inter-AS connections?
  - Each node is an autonomous system
    - Position around circle based on geographic location
    - Distance from centre based on number of links to other networks → more links, closer to the centre
- Each edge shows a link between two autonomous systems
  - A **potential route** traffic can flow
  - Not all possible routes are in use



Source: CAIDA (Feb. 2017)
http://www.caida.org/research/topology/as_core_network/2017/

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

AS-level Internet Topology Graph (IPv4 and IPv6)

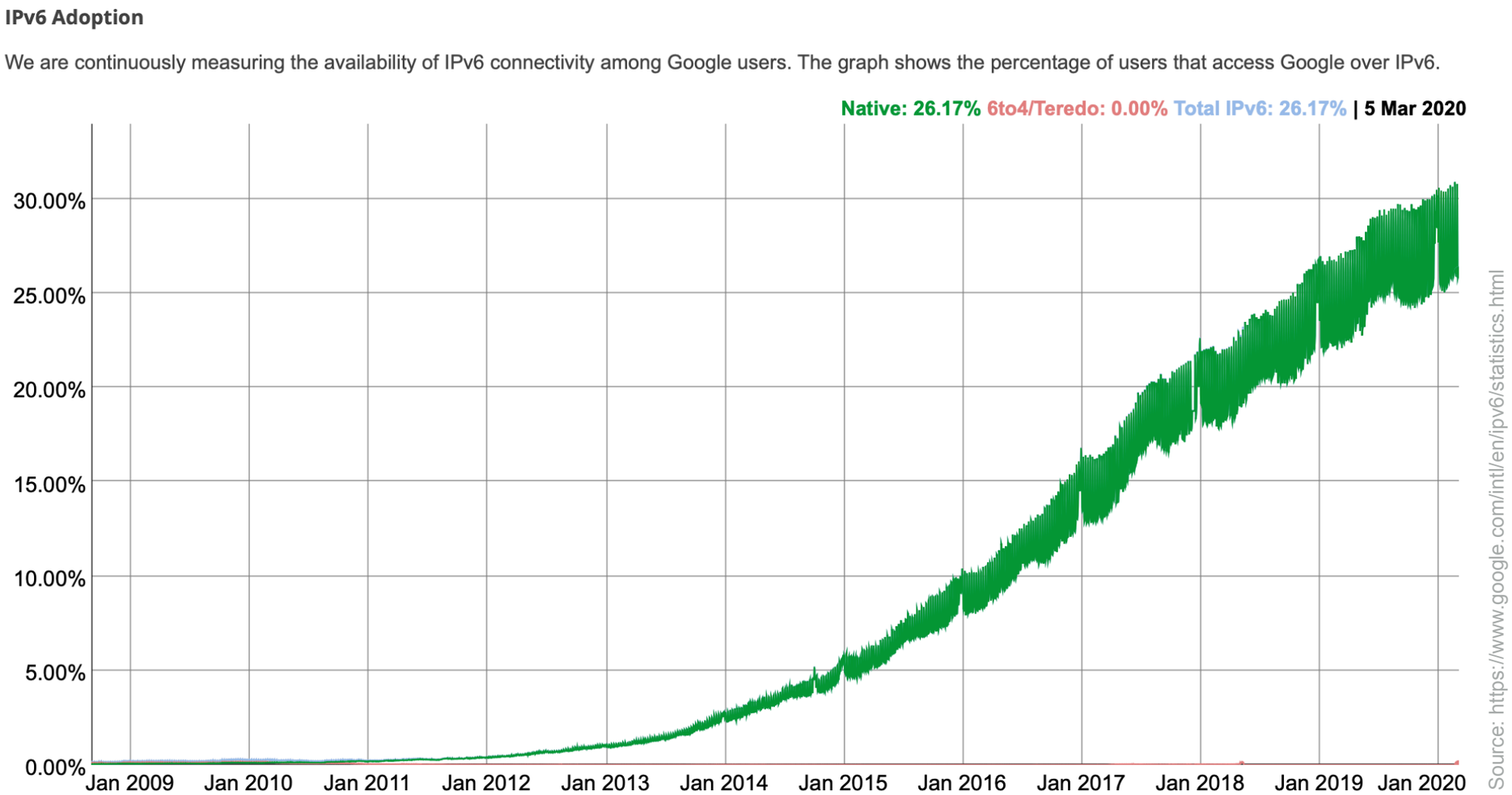


Copyright © 2015 UC Regents. All rights reserved.

Source: http://www.caida.org/research/topology/as_core_network/2015/

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

Differences Between IPv4 and IPv6 AS-level Topologies



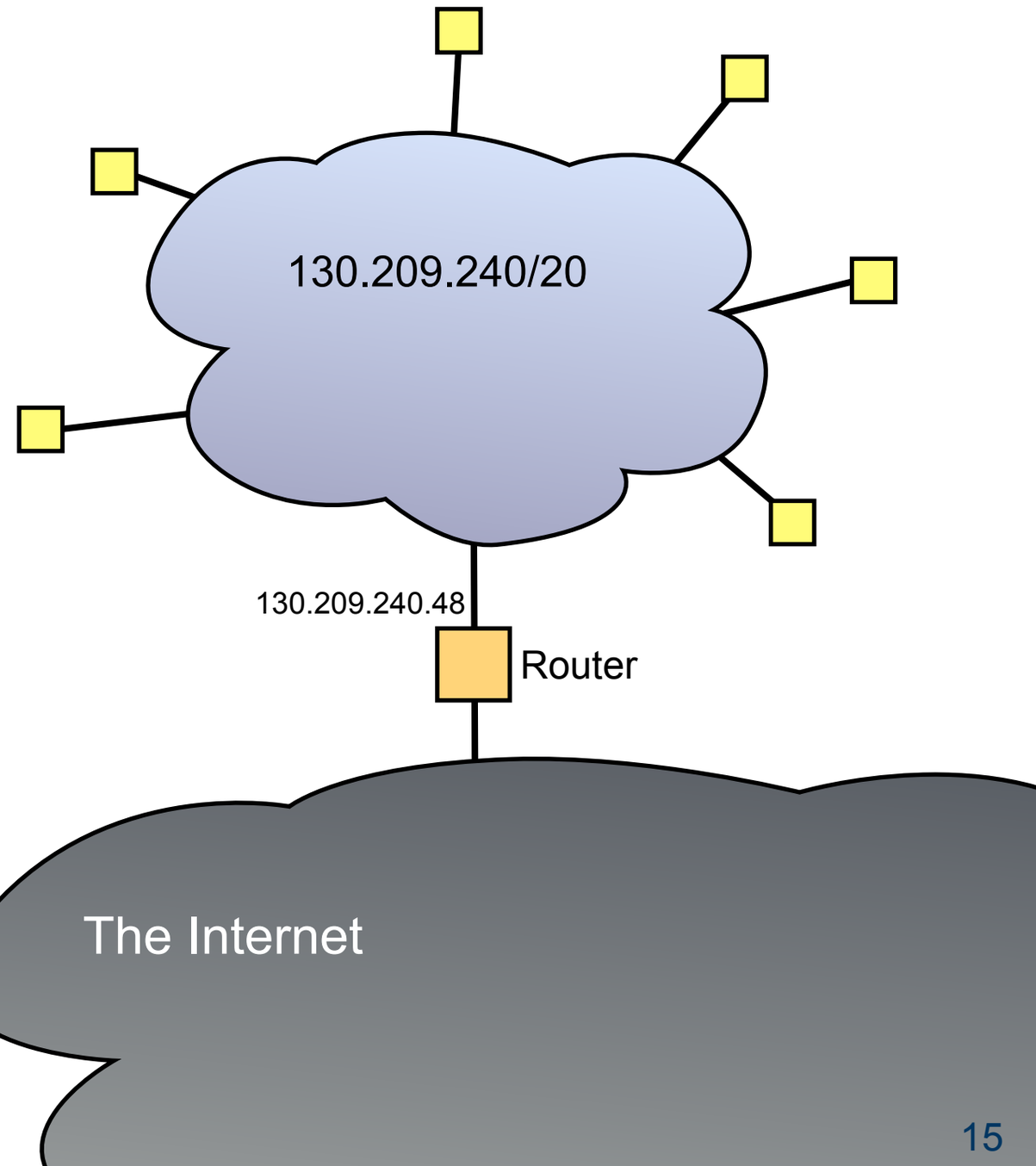
- IPv6 AS-level topology graph is sparse compared to IPv4, but there is significant IPv6 usage → IPv4 has 30 years head start on deployment; the topology can be expected to be more densely interconnected

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

Routing at the Edge

- Devices at the edge tend to have simple **routing tables**
  - The devices on the local network
  - A **default route** for everything else
  - e.g., routing table for hosts in Glasgow SoCS

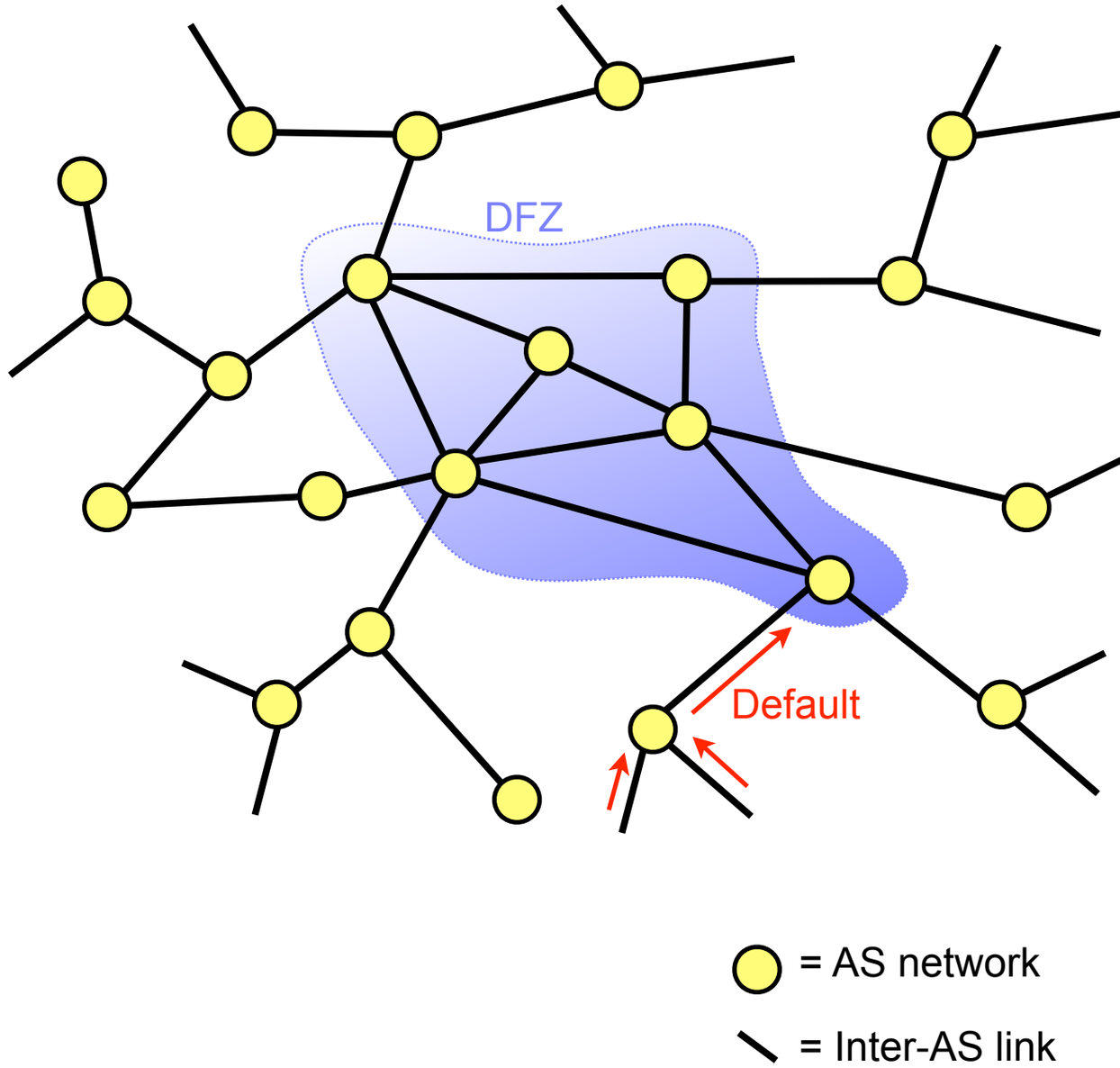
```
Network:        Netmask:        Gateway:
130.209.240.0   255.255.240.0   eth0
default         0.0.0.0         130.209.240.48
```



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

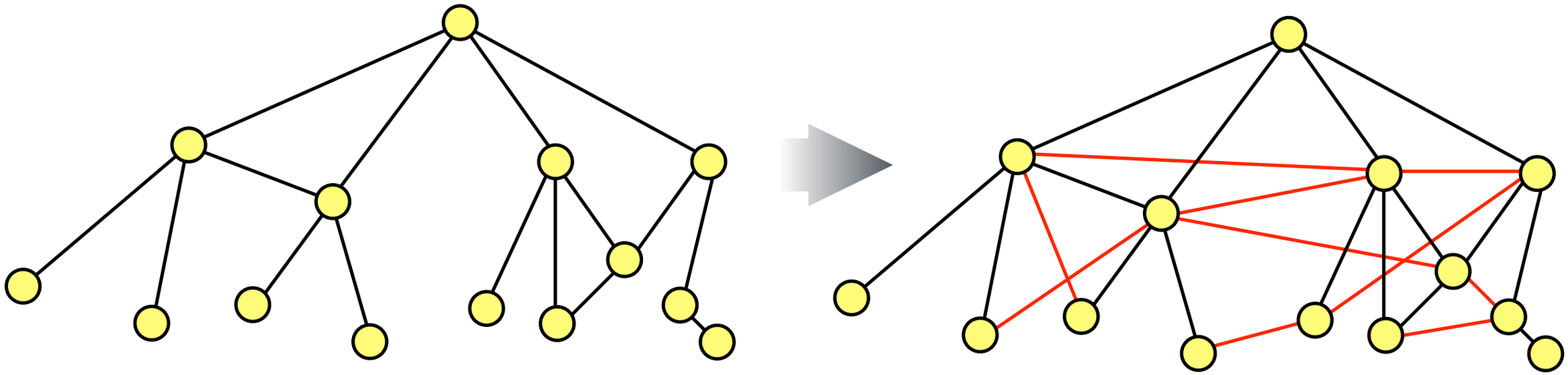
Routing in the Core of the Internet (1/2)

- Networks near the edge can often use a **default route** for much of their traffic
- Core networks need full routing table
  - The **default free zone** (DFZ)
  - With enough interconnections, they must store routing information about the broad network topology
- Many possible paths – need information to choose the most appropriate



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

Routing in the Core of the Internet (2/2)



- Internet AS-level topology is flattening → increasingly rich interconnections
- **Internet Exchange Points (IXPs)** are becoming commonplace
  - e.g., London Internet Exchange (LINX)
    - >850 different networks interconnected; >4.3 Tbps of traffic
  - Hundreds of such exchange points exist
- **Content Distribution Networks (CDNs)** and **hypergiants** interconnect broadly to deliver content to **eyeball** networks → consumers



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

Routing Policy

- Densely interconnected topology → routers must store detailed network map to make routing decisions
- Interdomain routing is between competitors
  - Each AS is independently operated and may compete for customers
  - Business, political, and economic relationships influence routing
- Routing **must** consider policy
  - Policy restrictions on who can determine the network topology
  - Policy restrictions on which route traffic should follow between a particular source and destination
  - Policy restrictions might prioritise non-shortest path routes
    - Do you prioritise cost, bandwidth, or latency when choosing a route?
    - Traffic between certain networks may be prohibiting from passing through other networks
    - Traffic between certain regions should avoid other regions

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

Border Gateway Protocol (BGP)

- Internet routing uses the **Border Gateway Protocol (BGP)**
  - External BGP (eBGP) used to exchange routing information between ASes
    - An eBGP session is used by an AS to exchange information about the AS-level topology with a neighbouring AS
    - Runs over a TCP connection between two routers, one in each AS
    - Exchanges knowledge of the AS-level topology of the network, filtered according to policy
    - Used to compute appropriate **interdomain routes** to each destination AS
  - Internal BGP (iBGP) propagates this information to routers within an AS
    - An **intra-domain routing protocol** determines routes within the AS
    - iBGP distributes information to internal routers on how to reach external destinations

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

Routing Information Exchanged in eBGP

- eBGP routers advertise lists of IP address ranges (“prefixes”) and their associated AS-level paths
- Combined to form a routing table

```
    Prefix              Next Hop          AS Path
…
*   12.10.231.0/24      194.68.130.254    5459 5413 5696 7369 i
*                       158.43.133.48     1849 702 701 6113 5696 7369 i
*                       193.0.0.242       3333 286 5696 7369 i
*                       204.212.44.128    234 266 237 3561 5696 7369 i
*>                      202.232.1.8       2497 5696 7369 i
*                       204.70.4.89       3561 5696 7369 i
*                       131.103.20.49     1225 3561 5696 7369 i
*   62.224.0.0/19       134.24.127.3      1740 3561 2578 8371 i
*                       194.68.130.254    5459 2529 3561 2578 8371 i
*                       158.43.133.48     1849 702 701 3216 3216 3216 8371 8371 i
*                       193.0.0.242       3333 286 3561 2578 8371 i
*                       144.228.240.93    1239 3561 2578 8371 i
*                       204.212.44.128    234 266 237 3561 2578 8371 i
*                       202.232.1.8       2497 3561 2578 8371 i
*                       205.238.48.3      2914 3561 2578 8371 i
*>                      204.70.4.89       3561 2578 8371 i
*                       131.103.20.49     1225 3561 2578 8371 i
…
```

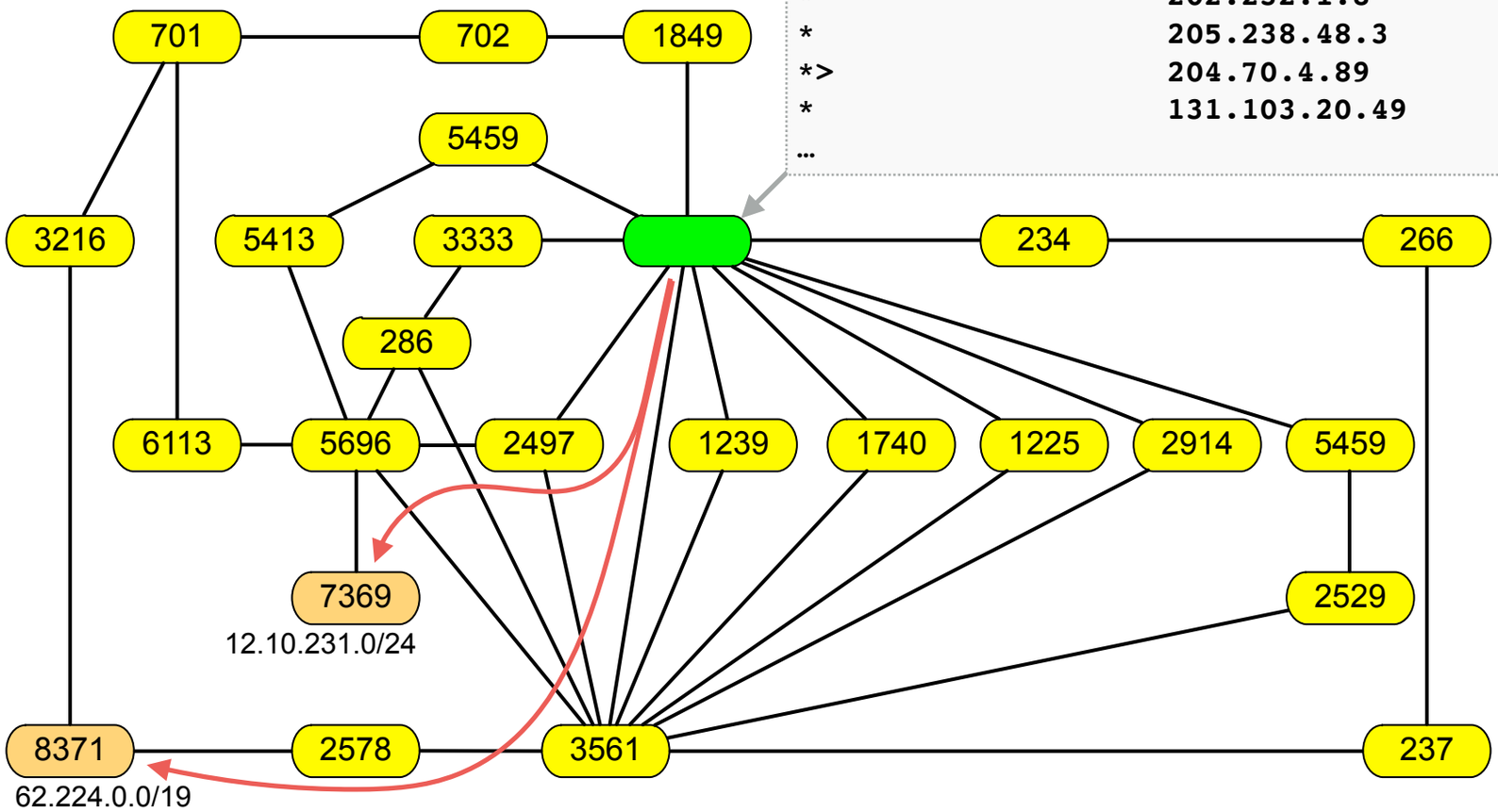
Hosts with IP addresses in the range 12.10.231.0 - 12.10.231.255 are in AS 7369. That AS is best reached via AS 2497 and then AS 5696. Packets destined for those addresses should be sent to address 202.232.1.8 next, from where they will be forwarded.

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

AS Topology Graph

An example fragment of the AS topology graph:

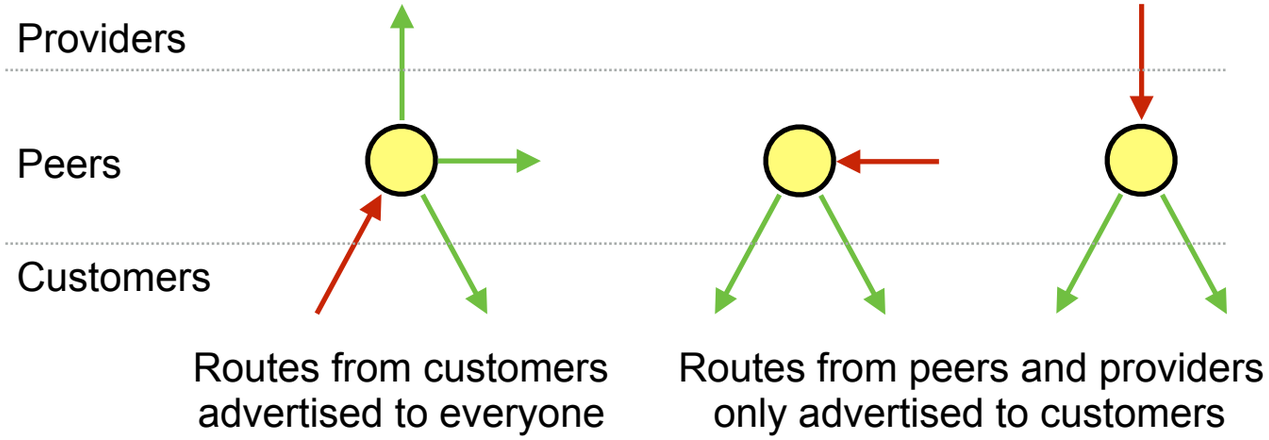
```
   Prefix            Next Hop          AS Path
…
*  12.10.231.0/24    194.68.130.254    5459 5413 5696 7369 i
*                    158.43.133.48     1849 702 701 6113 5696 7369 i
*                    193.0.0.242       3333 286 5696 7369 i
*                    204.212.44.128    234 266 237 3561 5696 7369 i
*>                   202.232.1.8       2497 5696 7369 i
*                    204.70.4.89       3561 5696 7369 i
*                    131.103.20.49     1225 3561 5696 7369 i
*  62.224.0.0/19     134.24.127.3      1740 3561 2578 8371 i
*                    194.68.130.254    5459 2529 3561 2578 8371 i
*                    158.43.133.48     1849 702 701 3216 3216 3216 8371 8371 i
*                    193.0.0.242       3333 286 3561 2578 8371 i
*                    144.228.240.93    1239 3561 2578 8371 i
*                    204.212.44.128    234 266 237 3561 2578 8371 i
*                    202.232.1.8       2497 3561 2578 8371 i
*                    205.238.48.3      2914 3561 2578 8371 i
*>                   204.70.4.89       3561 2578 8371 i
*                    131.103.20.49     1225 3561 2578 8371 i
…
```

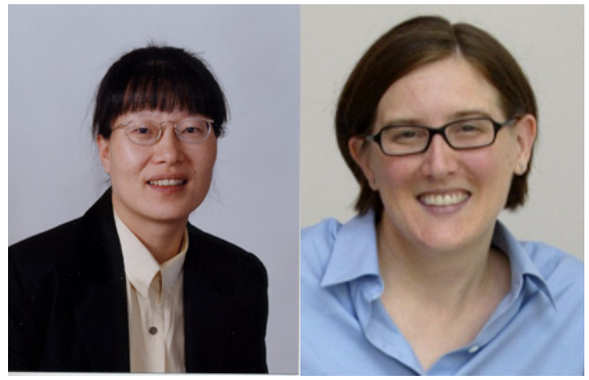


Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Routing Policy in eBGP

- Each AS chooses what routes to advertise to its neighbours
- Doesn’t need to advertise everything it receives
  - Usual to drop some routes from the advertisement – depends on the chosen routing policy
  - Common approach: the Gao-Rexford rules:





Lixin Gao Jennifer Rexford

- Gao-Rexford rules specify **what routes are advertised**, not how traffic is forwarded – difference between **control plane** and **data plane**
- Resulting graph is valley-free DAG: traffic flows up to common provider, then down
- Designed to optimise **profit**, not routing efficiency – don’t advertise expensive routes

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

BGP Routing Decision Process (1/2)

- Each AS exchanges routing information with its neighbours
  - AS-level topology information, filtered based on policies applied by each AS
  - This gives a partial view of AS-level network topology
- Each AS applies policy to choose what routes to configure for use
  - Choose what route to install for destination prefix in forwarding table based on multiple criteria – policy, shortest path, etc.
  - BGP may not find a route, even if one exists, as may be prohibited by policy
  - Routes are often not the shortest AS path
- Each AS applies policy to determine what routes to advertise to neighbours
  - Filter out routes the neighbour should not use (e.g., Gao-Rexford rules, and other reasons)
  - De-prioritise certain other routes
  - Tag routes for special processing where business agreements exist
  - Mapping business goals to BGP policies is poorly documented → many operational secrets

Table 2: Simplified BGP decision process [6, 24]. This table was also provided with the survey.

#	Criteria
1	Highest LocalPref
2	Lowest AS Path Length
3	Lowest origin type
4	Lowest MED
5	eBGP-learned over iBGP-learned
6	Lowest IGP cost to border router (hot-potato routing)
7	If both paths are external, prefer the path that was received first (i.e., the oldest path) [6]
8	Lowest router ID (to break ties)

Source: Phillipa Gill, Michael Schapira, and Sharon Goldberg, "A Survey of Interdomain Routing Policies", ACM CCR, V44, N1, January 2014, p29-34

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

BGP Routing Decision Process (2/2)

- BGP exchanges routes between autonomous systems
  - Policy, economic, and political concerns outweigh shortest path in many cases
  - Routing is between competitors – there is little trust and little public data
  - The BGP decision process is public – the input data is not, making it difficult to evaluate how routing decisions are made

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

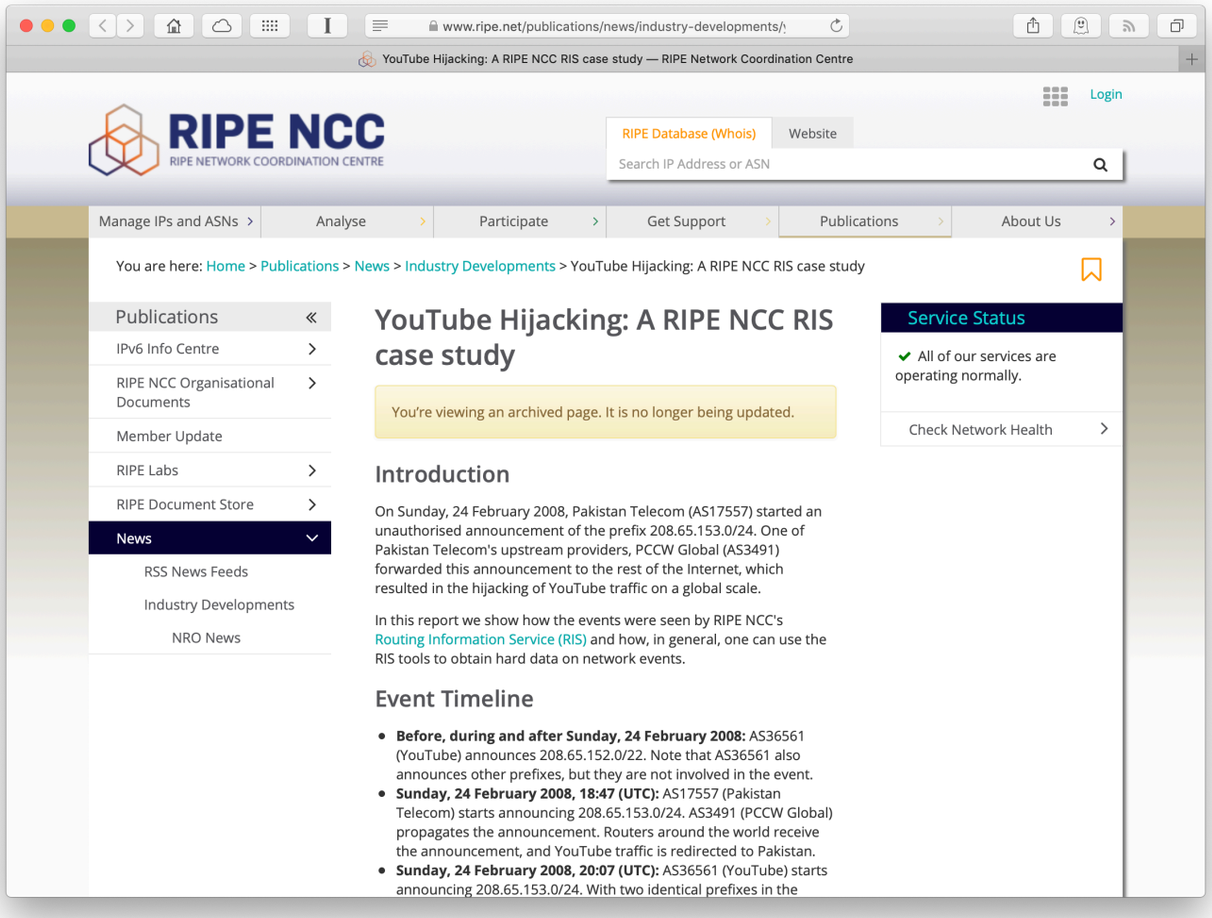
Routing Security

- RKPI
- MANRS

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

BGP Security (1/3)

- BGP announcements are not secure
- Any AS participating in BGP routing can announce any address prefix – whether or not they own it
  - Sometimes this is accidental
  - Sometimes this is malicious
- Result is that traffic is misdirected
  - Known as “BGP hijacking”
  - Security risk if traffic can be misdirected
  - Accidental hijacking is a serious stability problem for the network



https://www.ripe.net/publications/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study

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

BGP Security (2/3)

- Resource Public Key Infrastructure (RKPI)
  - Allows AS to make a signed Route Origin Authorisation (ROA)
  - Digital signature validating that an AS is authorised to announce a particular IP address prefix
  - Hierarchical delegation from Regional Internet Registry to AS to sub-domains using a chain of digital signatures
  - Allows a router to validate whether a prefix announcement is authorised – hijacked prefixes will not have a valid signature
    - High computational cost to validate signatures
    - As of 2019, only ~12% of IPv4 addresses are covered by prefixes with valid digital signatures → those are safe from hijacking, if the routers choose to check signatures



RPKI is Coming of Age

A Longitudinal Study of RPKI Deployment and Invalid Route Origins

Taejoong Chung, Rochester Institute of Technology
Emile Aben, RIPE NCC
Tim Bruijnzeels, NLNetLabs
Balakrishnan Chandrasekaran, Max Planck Institute for Informatics
David Choffnes, Northeastern University
Dave Levin, University of Maryland
Bruce M. Maggs, Duke University and Akamai Technologies
Alan Mislove, Northeastern University
Roland van Rijswijk-Deij, University of Twente and NLNetLabs
John Rula, Akamai Technologies
Nick Sullivan, Cloudflare

ABSTRACT

Despite its critical role in Internet connectivity, the Border Gateway Protocol (BGP) remains highly vulnerable to attacks such as prefix hijacking, where an Autonomous System (AS) announces routes for IP space it does not control. To address this issue, the Resource Public Key Infrastructure (RPKI) was developed starting in 2008, with deployment beginning in 2011. This paper performs the first comprehensive, longitudinal study of the deployment, coverage, and quality of RPKI.

We use a unique dataset containing *all* RPKI Route Origin Authorizations (ROAs) from the moment RPKI was first deployed, more than 8 years ago. We combine this dataset with BGP announcements from more than 3,300 BGP collectors worldwide. Our analysis shows the after a gradual start, RPKI has seen a rapid increase in adoption over the past two years. We also show that although misconfigurations were rampant when RPKI was first deployed (causing many announcements to appear as invalid) they are quite rare today. We develop a taxonomy of invalid RPKI announcements, then quantify their prevalence. We further identify suspicious announcements indicative of prefix hijacking and present case studies of likely hijacks.

Overall, we conclude that while misconfigurations still do occur, RPKI is "ready for the big screen," and routing security can be increased by dropping invalid announcements. To foster reproducibility and further studies, we release all RPKI data and the tools we used to analyze it into the public domain.

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. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

IMC '19, October 21–23, 2019, Amsterdam, Netherlands

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6948-0/19/10…$15.00
https://doi.org/10.1145/3355369.3355596

T. Chung, E. Aben, T. Bruijnzeels, B. Chandrasekaran, D. R. Choffnes, D. Levin, B. M. Maggs, A. Mislove, R. van Rijswijk-Deij, J. Rula, and N. Sullivan, "RPKI is coming of age: A longitudinal study of RPKI deployment and invalid route origins", Proc. Internet Measurement Conference, Amsterdam, The Netherlands, October 2019. ACM. https://dx.doi.org/10.1145/3355369.3355596

CC BY ND

BGP Security (3/3)





- **Mutually Agreed Norms for Routing Security (MANRS)** project aims to improve routing security
  - Filtering
  - Anti-spoofing
  - Coordination
  - Global validation via RKPI
- Combination of signed route advertisements via RKPI to catch malicious attacks, with best practises for route filtering, anti-spoofing, and coordination between ISPs to protect against misconfigurations

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

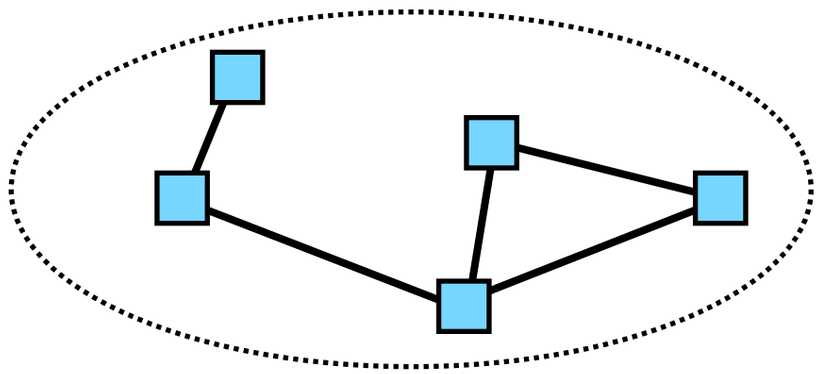
Intra-domain Routing

- Distance vector and link state algorithms

BY ND

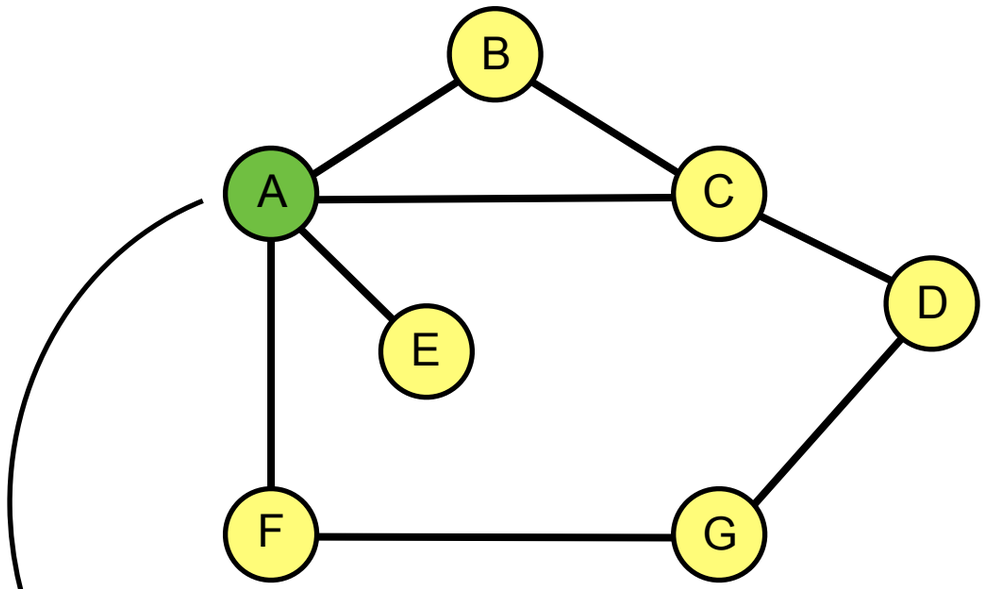
Intra-domain Unicast Routing

- BGP gives information on the path to reach other networks
- How to route traffic within an AS?
  - Single trust domain
    - No policy restrictions on who can determine network topology
    - No policy restrictions on which links can be used
  - Desire efficient routing → shortest path
    - Make best use of the network you have available
- Two approaches
  - Distance vector – the Routing Information Protocol (RIP)
  - Link state – Open Shortest Path First routing (OSPF)



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

Distance Vector Routing



Routing Table at Node A

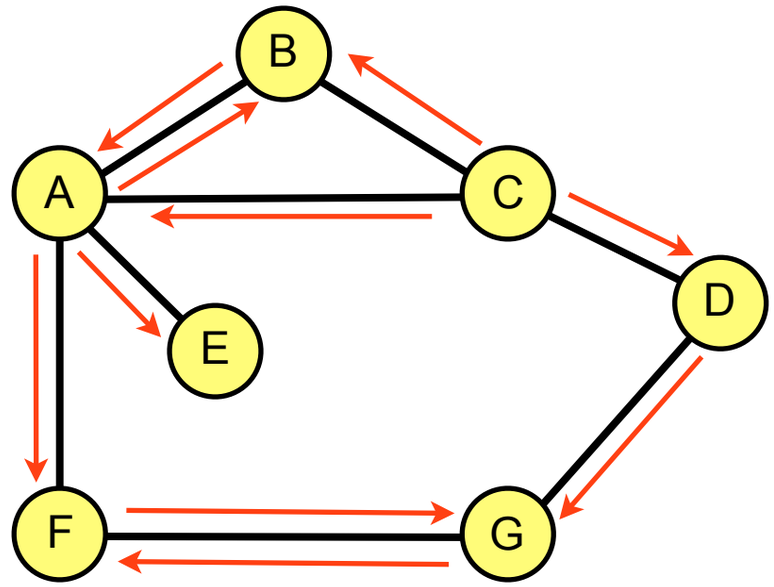
Destination	Distance	Next Hop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F

- Nodes maintain a vector containing the distance to every other node in network
  - Periodically exchanged with neighbours, eventually every node know distance and next hop to every other node
- Packets forwarded along path with least distance to destination
- Not widely used in practice:
  - Slow to converge in normal operation
  - Count-to-infinity problem makes this worse on failure in networks containing loops

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

Link State Routing

- Nodes know the links to their neighbours, and cost of using those links – **link state** information
  - Flood this information throughout the network on startup, and whenever a link is added or removed
    - Send address of node sending the update, the list of directly connected neighbours and costs for link usage, and message sequence number
    - Sequence numbers prevent loops
  - Each node then directly calculates shortest path to every other node, uses as routing table
- Flooding link state data from all nodes ensures all nodes know the entire topology
  - Each node uses Dijkstra's shortest-path algorithm to calculate optimal route to every other node
  - Optimal is assumed to be the shortest path, by cost



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

Distance Vector vs. Link State

Distance vector routing:

- Simple to implement
- Routers only store the distance to each other node: O(n)
- Suffers from slow convergence

Link State routing:

- More complex
- Requires each router to store complete network map: O(n^2)
- Much faster convergence

Slow convergence times make distance vector routing unsuitable for large networks

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

Challenges in Intra-domain Routing

- How to rapidly recover from link failures?
  - Construction work surprisingly good at breaking network cables – trawlers do similar damage to undersea cables
  - Good network designs have multiple paths from source to destination
  - Must quickly notice failure and switch to backup path
    - If the link is used for telephony or video streaming, this must happen in $<1/60^{th}$ of a second else it interrupts the speech or video playout
  - **Fast failover** – pre-calculate alternative paths to allow rapid switchover
- How to balance load across multiple paths?
  - **Equal cost multipath routing** – if there are two paths with the same cost, alternate traffic between them
  - Some transport protocols dislike reordering; may need to ensure packets from a single flow always use the same path
    - e.g., TCP uses a triple duplicate ACK to signal loss so is insensitive to small amount of packet re-ordering, but treats large amounts of reordering as loss and slows down
    - e.g., RTP applications don't care about order, as long as packets arrive before deadline



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

Internet Routing

- Content distribution networks
- Inter-domain routing and security
- Intra-domain routing

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

Wrap-up

- Remaining Lecture
- Remaining laboratory sessions
- Examination
- Projects

CC BY ND

Remaining Lecture

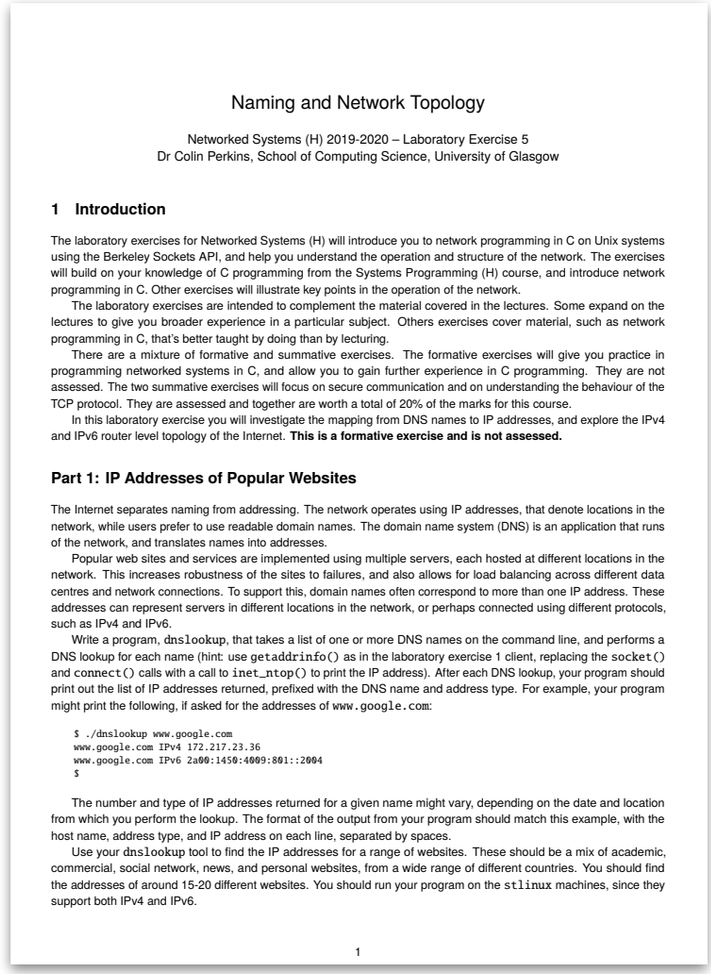
- Lecture 10 is cancelled – was to be on Future Directions
  - A **non-assessed** overview of open issues in networking
    - Path-aware networking
    - Computation in the network
    - Quantum Internet
    - Information centric networking
    - Human rights and network protocols
    - …
  - If interested, the Internet Research Task Force (https://irtf.org/) considers and develops such ideas



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

Remaining Laboratory Sessions

- Laboratory sessions timetabled for 12th and 19th March are cancelled
  - Lab 5 is a formative exercise on Naming and Network Topology – **not assessed**
  - Recommended that you complete in your own time, to help understand Internet routing
  - If you have questions, please email



Naming and Network Topology

Networked Systems (H) 2019-2020 – Laboratory Exercise 5
Dr Colin Perkins, School of Computing Science, University of Glasgow

1 Introduction

The laboratory exercises for Networked Systems (H) will introduce you to network programming in C on Unix systems using the Berkeley Sockets API, and help you understand the operation and structure of the network. The exercises will build on your knowledge of C programming from the Systems Programming (H) course, and introduce network programming in C. Other exercises will illustrate key points in the operation of the network.

The laboratory exercises are intended to complement the material covered in the lectures. Some expand on the lectures to give you broader experience in a particular subject. Others exercises cover material, such as network programming in C, that's better taught by doing than by lecturing.

There are a mixture of formative and summative exercises. The formative exercises will give you practice in programming networked systems in C, and allow you to gain further experience in C programming. They are not assessed. The two summative exercises will focus on secure communication and on understanding the behaviour of the TCP protocol. They are assessed and together are worth a total of 20% of the marks for this course.

In this laboratory exercise you will investigate the mapping from DNS names to IP addresses, and explore the IPv4 and IPv6 router level topology of the Internet. **This is a formative exercise and is not assessed.**

Part 1: IP Addresses of Popular Websites

The Internet separates naming from addressing. The network operates using IP addresses, that denote locations in the network, while users prefer to use readable domain names. The domain name system (DNS) is an application that runs of the network, and translates names into addresses.

Popular web sites and services are implemented using multiple servers, each hosted at different locations in the network. This increases robustness of the sites to failures, and also allows for load balancing across different data centres and network connections. To support this, domain names often correspond to more than one IP address. These addresses can represent servers in different locations in the network, or perhaps connected using different protocols, such as IPv4 and IPv6.

Write a program, `dnslookup`, that takes a list of one or more DNS names on the command line, and performs a DNS lookup for each name (hint: use `getaddrinfo()` as in the laboratory exercise 1 client, replacing the `socket()` and `connect()` calls with a call to `inet_ntop()` to print the IP address). After each DNS lookup, your program should print out the list of IP addresses returned, prefixed with the DNS name and address type. For example, your program might print the following, if asked for the addresses of `www.google.com`:

```
$ ./dnslookup www.google.com
www.google.com IPv4 172.217.23.36
www.google.com IPv6 2a00:1450:4009:801::2004
$
```

The number and type of IP addresses returned for a given name might vary, depending on the date and location from which you perform the lookup. The format of the output from your program should match this example, with the host name, address type, and IP address on each line, separated by spaces.

Use your `dnslookup` tool to find the IP addresses for a range of websites. These should be a mix of academic, commercial, social network, news, and personal websites, from a wide range of different countries. You should find the addresses of around 15-20 different websites. You should run your program on the `stlinux` machines, since they support both IPv4 and IPv6.

1

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

Examination and Coursework

- Course is examined – final exam is worth 20% of the marks
  - Past papers available on Moodle and course website
  - **Significant curriculum revision this year** – sample exam with answers will be provided before the end of March

- Exercise 1 marks and feedback available via Student LTC system
- Exercise 2 marks and feedback should be available by end March

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

Projects

- If you are interested in Level 4 or MSci projects relating to networked systems, please contact me – colin.perkins@glasgow.ac.uk
  - Keen to work with motivated students to develop projects on topics including:
    - Improving Internet protocol standards and specifications
    - Transport protocols, real-time applications, and QUIC
    - New networking APIs in high-level languages (e.g., Rust)
    - Network measurement
- Strong focus on interactions with IETF standards and research communities:
  - Some potential projects are strongly technical
  - Others focus heavily on the standardisation process

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

The End

BY ND