University of Glasgow

Naming and the Tussle for Control

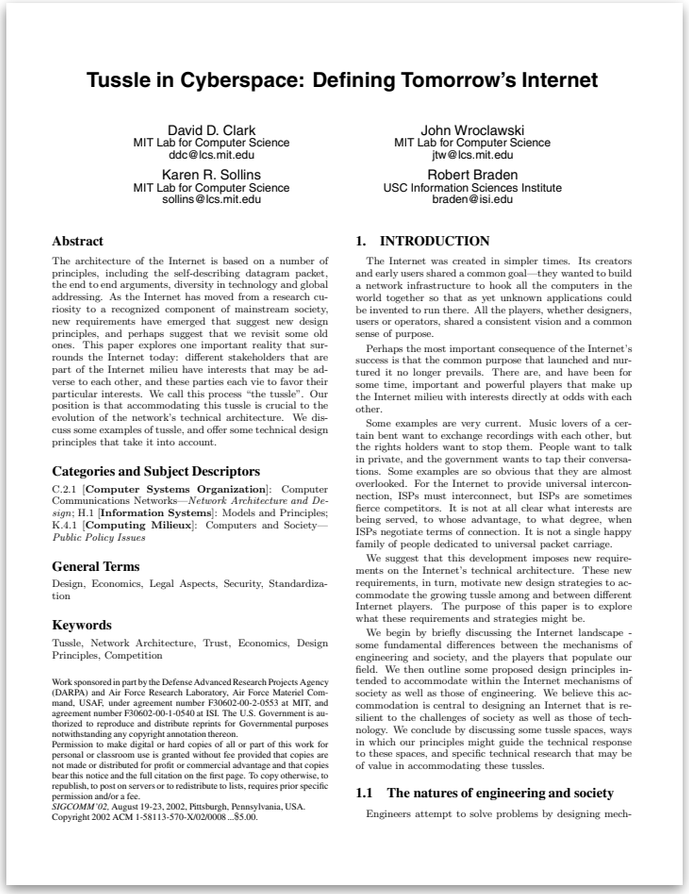
Networked Systems (H) – Lecture 8

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

- DNS structure and organisation
- Methods for DNS resolution
- The politics of names



Tussle in Cyberspace: Defining Tomorrow's Internet

David D. Clark
MIT Lab for Computer Science
ddc@lcs.mit.edu

John Wroclawski
MIT Lab for Computer Science
jtw@lcs.mit.edu

Karen R. Sollins
MIT Lab for Computer Science
sollins@lcs.mit.edu

Robert Braden
USC Information Sciences Institute
braden@isi.edu

Abstract

The architecture of the Internet is based on a number of principles, including the self-describing datagram packet, the end to-end arguments, diversity in technology and global addressing. As the Internet has moved from a research curiosity to a recognized component of mainstream society, new requirements have emerged that suggest new design principles, and perhaps suggest that we revisit some old ones. This paper explores one important reality that surrounds the Internet today: different stakeholders that are part of the Internet milieu have interests that may be adverse to each other, and these parties each vie to favor their particular interests. We call this process "the tussle". Our position is that accommodating this tussle is crucial to the evolution of the network's technical architecture. We discuss some examples of tussle, and offer some technical design principles that take it into account.

Categories and Subject Descriptors

C.2.1 [**Computer Systems Organization**]: Computer Communications Networks—*Network Architecture and Design*; H.1 [**Information Systems**]: Models and Principles; K.4.1 [**Computing Milieux**]: Computers and Society—*Public Policy Issues*

General Terms

Design, Economics, Legal Aspects, Security, Standardization

Keywords

Tussle, Network Architecture, Trust, Economics, Design Principles, Competition

Work sponsored in part by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-00-2-0553 at MIT, and agreement number F30602-00-1-0540 at ISI. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.
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.
SIGCOMM'02, August 19-23, 2002, Pittsburgh, Pennsylvania, USA.
Copyright 2002 ACM 1-58113-570-X/02/0008...$5.00.

1. INTRODUCTION

The Internet was created in simpler times. Its creators and early users shared a common goal—they wanted to build a network infrastructure to hook all the computers in the world together so that as yet unknown applications could be invented to run there. All the players, whether designers, users or operators, shared a consistent vision and a common sense of purpose.

Perhaps the most important consequence of the Internet's success is that the common purpose that launched and nurtured it no longer prevails. There are, and have been for some time, important and powerful players that make up the Internet milieu with interests directly at odds with each other.

Some examples are very current. Music lovers of a certain bent want to exchange recordings with each other, but the rights holders want to stop them. People want to talk in private, and the government wants to tap their conversations. Some examples are so obvious that they are almost overlooked. For the Internet to provide universal interconnection, ISPs must interconnect, but ISPs are sometimes fierce competitors. It is not at all clear what interests are being served, to whose advantage, to what degree, when ISPs negotiate terms of connection. It is not a single happy family of people dedicated to universal packet carriage.

We suggest that this development imposes new requirements on the Internet's technical architecture. These new requirements, in turn, motivate new design strategies to accommodate the growing tussle among and between different Internet players. The purpose of this paper is to explore what these requirements and strategies might be.

We begin by briefly discussing the Internet landscape - some fundamental differences between the mechanisms of engineering and society, and the players that populate our field. We then outline some proposed design principles intended to accommodate within the Internet mechanisms of society as well as those of engineering. We believe this accommodation is central to designing an Internet that is resilient to the challenges of society as well as those of technology. We conclude by discussing some tussle spaces, ways in which our principles might guide the technical response to these spaces, and specific technical research that may be of value in accommodating these tussles.

1.1 The natures of engineering and society

Engineers attempt to solve problems by designing mech-

D. D. Clark, J. Wroclawski, K. R. Sollins, and R. Braden, "Tussle in cyberspace: Defining tomorrow's Internet", Proceedings of the SIGCOMM Conference, Pittsburgh, PA, USA, August 2002. ACM. https://dx.doi.org/10.1145/633025.633059

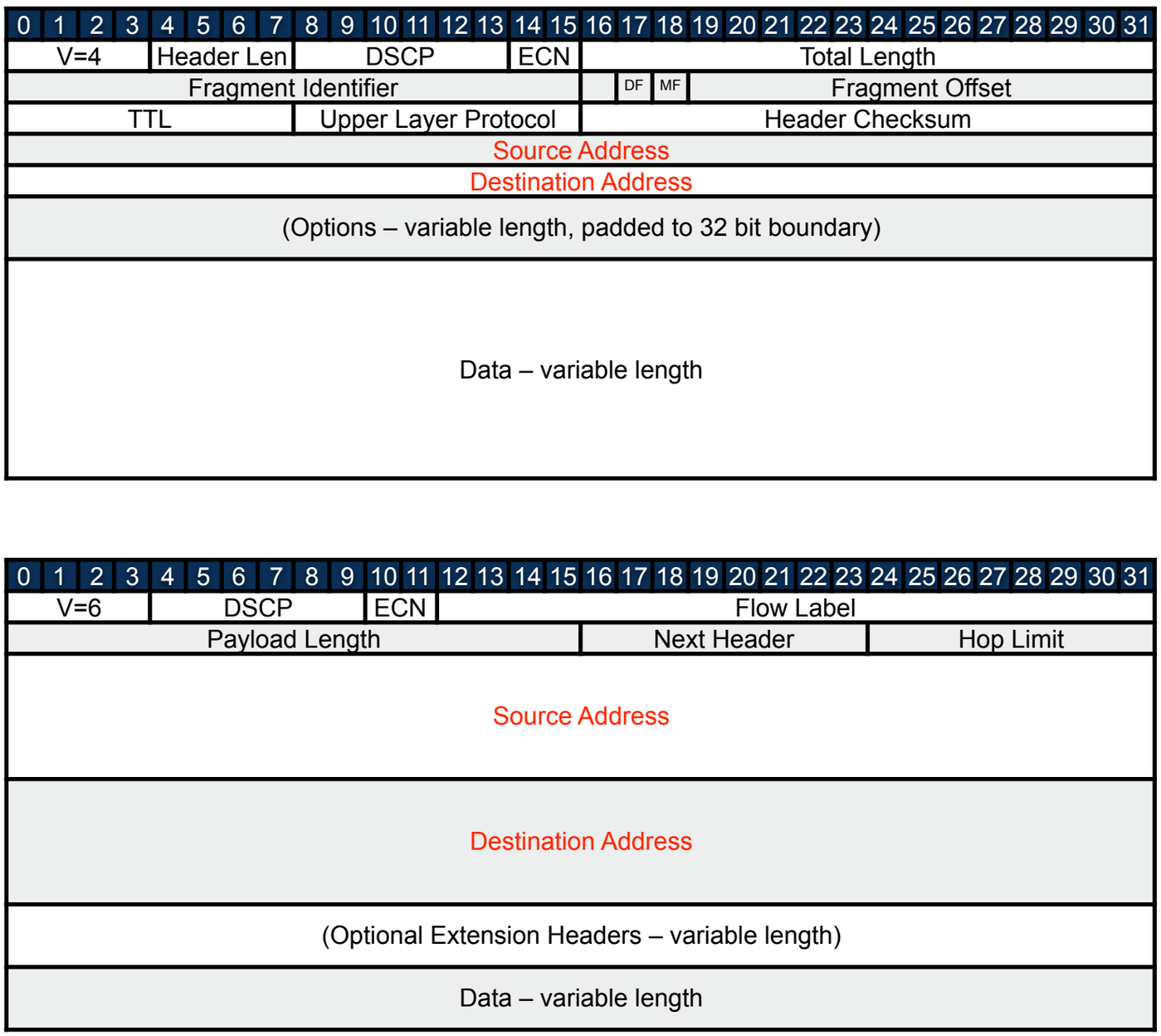
Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

DNS Structure and Organisation

- What is the DNS?
- Zones and queries
- Domains
- Security

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

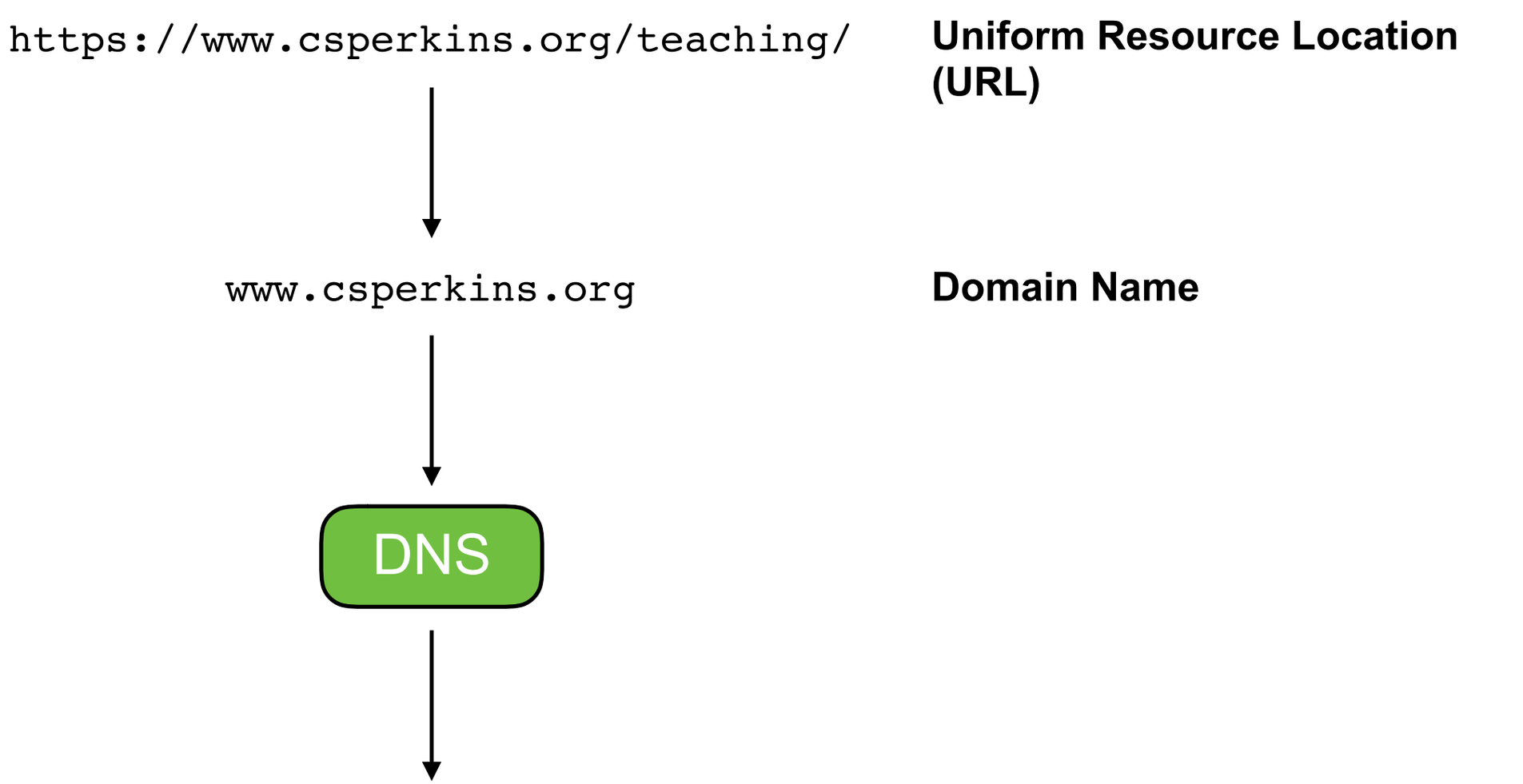
What is the DNS? (1/2)



- IP packets contain addresses rather than names
  - Designed for efficient processing by routers determining where to forward the packet
  - Not human readable – people prefer names, not addresses
- Domain name system (DNS) is a distributed database that maps names to IP addresses

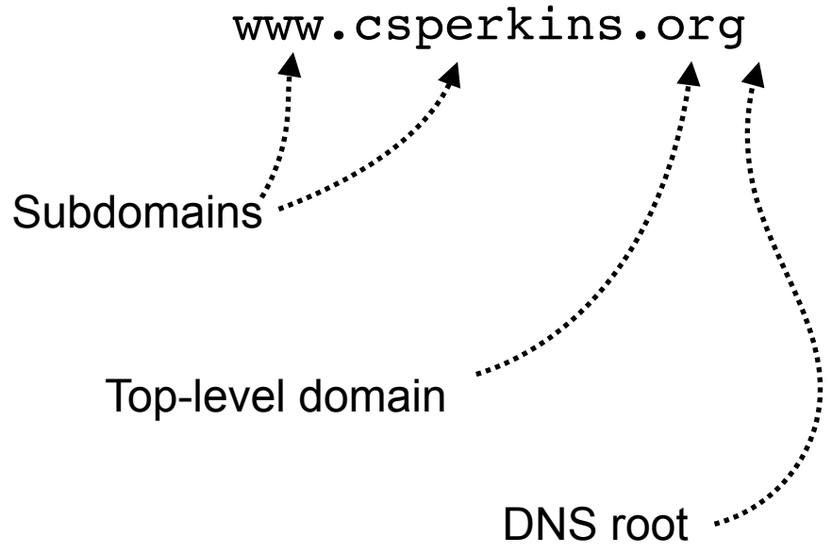
Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

What is the DNS? (2/2)



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

Structure of DNS Names

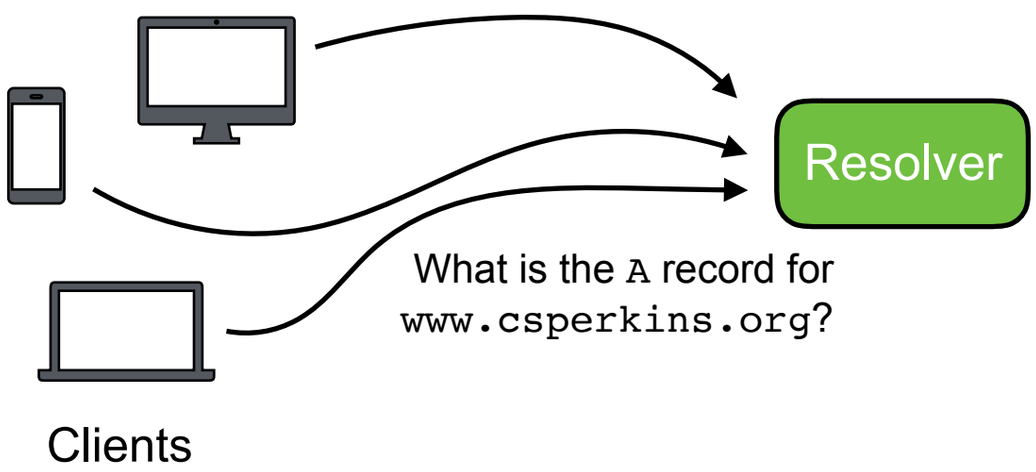


- Naming is hierarchical
  - Sub-domains first
  - Concludes with a **top-level domain** (TLD)
    - Country-code top-level domains (ccTLDs)
      - `.uk`, `.de`, `.cn`, `.io`, `.ly`, …
    - Generic top-level domains (gTLDs)
      - `.com`, `.org`, `.net`, …
  - Top-level domains live within the **DNS root**
    - The **root servers** advertise the top-level domains
    - They have well-known, fixed, IP addresses – new DNS resolvers need to reach them to find the TLDs before they can answer DNS queries
- Each level is independently administered and operated
  - DNS is a distributed database – in authority and implementation
  - Each level in the hierarchy controls its own data

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

DNS Name Resolution (1/3)

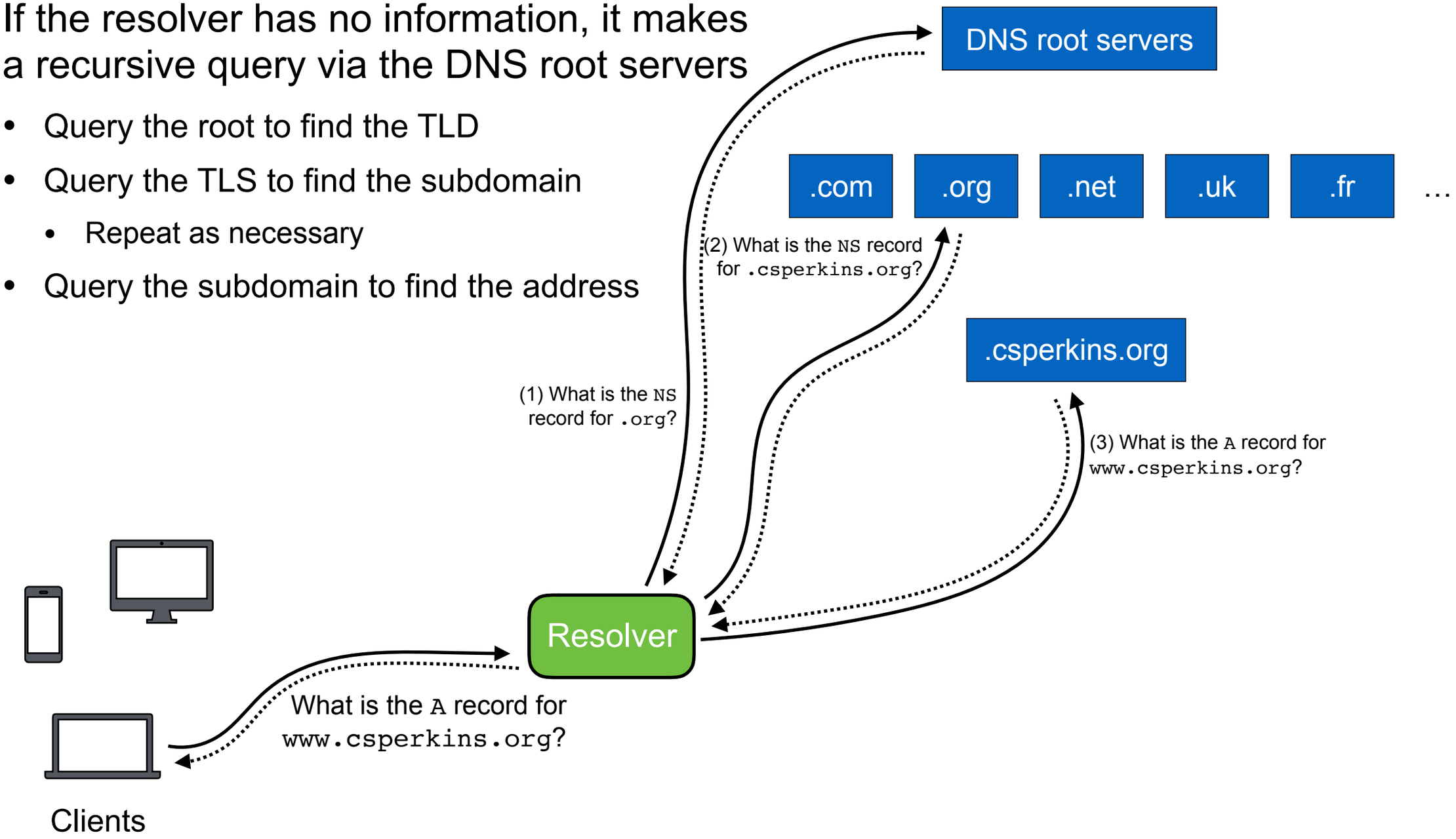
- The DNS is used for **name resolution**
  - Given a name, lookup a particular type of **record** relating to that name
  - Many different types of record: `A`, `AAAA`, `CNAME`, `MX`, `NS`, `SRV`, …
  - Most common are `A` and `AAAA` records, that map hostnames to IPv4 and IPv6 addresses, and `NS` records that identify the **name server** for a domain
- A DNS **client** asks its **resolver** to perform the lookup
  - By calling `getaddrinfo()`, or similar
  - The resolver could be a process running on the client, it more commonly runs on a machine provided by the network operator



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

DNS Name Resolution (2/3)

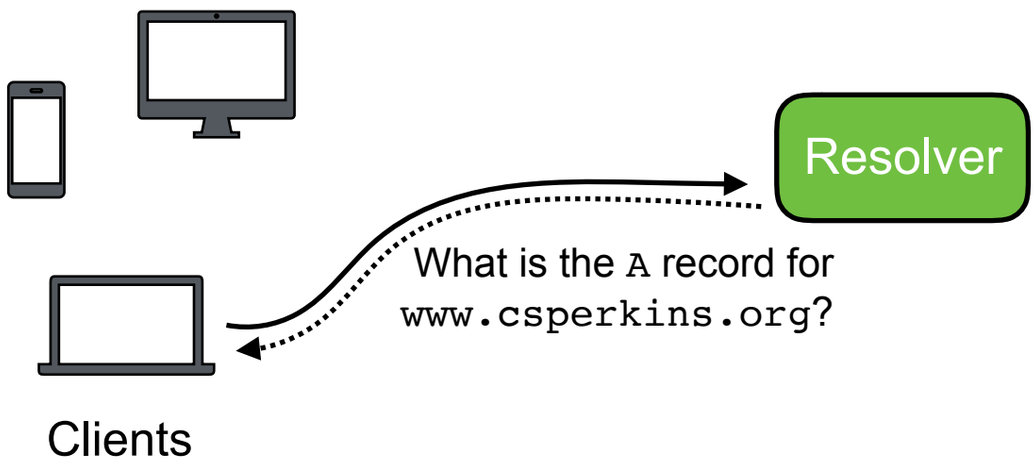
- If the resolver has no information, it makes a recursive query via the DNS root servers
  - Query the root to find the TLD
  - Query the TLS to find the subdomain
    - Repeat as necessary
  - Query the subdomain to find the address



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

DNS Name Resolution (3/3)

- Responses include a **time to live** that allows a resolver to cache the value for a certain time period
  - Subdomains can set a short TTL and give different answers each time a particular name is requested → load balancing; CDNs directing queries to local caches
- Subsequent queries are answered from the cache, where possible
  - If the cached entry times it, it's refreshed from the next level up in the hierarchy
  - Eventually reaching the DNS root
- The IP addresses for the root servers are well known, and never time out



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

DNS Zones

```
$TTL 3600          ; 1 hour
example.org.    IN      SOA     ns1.example.org. admin.example.org. (
                                2006051501      ; Serial
                                10800           ; Refresh
                                3600            ; Retry
                                604800          ; Expire
                                86400           ; Minimum TTL
                        )
; DNS Servers
                IN      NS      ns1.example.org.
                IN      NS      ns2.example.org.
; MX Records
                IN      MX 10   mx.example.org.
                IN      MX 20   mail.example.org.
; Machine Names
ns1             IN      A       192.168.1.2
ns2             IN      A       192.168.1.3
mx              IN      A       192.168.1.4
mail            IN      A       192.168.1.5
mail            IN      AAAA    2001:200:1000:0:25f:23ff:fe80:1234
server1         IN      A       192.168.1.10
server2         IN      A       192.168.1.11
; Aliases
www             IN      CNAME   server1
```

Each domain configures a **DNS zone** holding names and addresses for the domain

The name servers for the domain

The mail exchanger (email) servers for the domain

The IP addresses of the hosts within the domain

An alias for a host within the domain

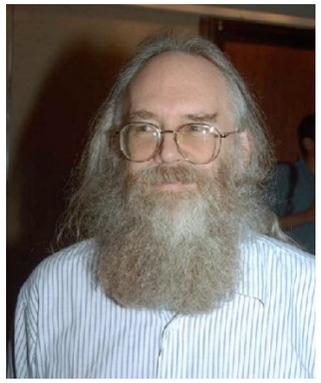
Source: adapted from The FreeBSD Handbook

- The root zone contains `NS`, `A`, and `AAAA` records for the TLDs
- The zone for each TLD contains `NS`, `A`, and `AAAA` records for its sub-domains
- …

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

Top-Level Domains (1/4)

- The set of top-level domains is controlled by ICANN
  - Complex history
    - ARPAnet project → IANA
    - Notional US government control via ARPAnet funding
    - Jon Postel at USC/ISI (http://www.ietf.org/rfc/rfc2468.txt – “I remember IANA”)
    - The Internet Corporation for Assigned Names and Numbers (ICANN)
      - US Government contractual control of ICANN ended on 1 October 2016
      - A US not-for-profit corporation based in Los Angeles
  - Complex governance model
    - Board of governors
    - Generic Names Supporting Organisation – gTLDs
    - Country Code Names Supporting Organisation – ccTLDs
    - Address Supporting Organisation – IP address allocation to regional registries
    - Governmental Advisory Committee – representatives from each of the 112 UN-recognised countries
    - At Large Advisory Committee
    - Root Sever Advisory Committee
    - Security and Stability Advisory Committee
    - Technical Liaison Group
    - Regular public meetings



Jon Postel

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

Top-Level Domains (2/4)

- ISO standard 3166-1 defines two-letter country names
  - To be included a country must either:
    - Be a member state of the United Nations
    - Be a member of one of the UN special agencies (ITU, IMF, UNESCO, WHO, …)
    - Be party to the International Court of Justice
- Every code included in ISO 3166-1 is address to the DNS root zone as a Country Code Top-Level Domain (ccTLD)
  - `.uk`, `.fr`, `.de`, `.cn`, `.us`, `.io`, `.ly`, …
    - Each country has it's own policy for sub-domains of the CCTLD
    - .cs – JANET Name Resolution Systems and the "Czechoslovakia problem"
  - Exceptions:
    - `.gb` – The UK should use `.gb` (`.gov.uk` used to be `.hmg.gb`, but `.gb` never widely used)
    - `.su` – The domain for the Soviet Union still exists and accepts new registrations
    - `.eu` – The EU is not an ISO 3166-1 country, but has a ccTLD
    - `.oz` – Australia, sadly, changed from `.oz` to `.au`

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

Top-Level Domains (3/4)

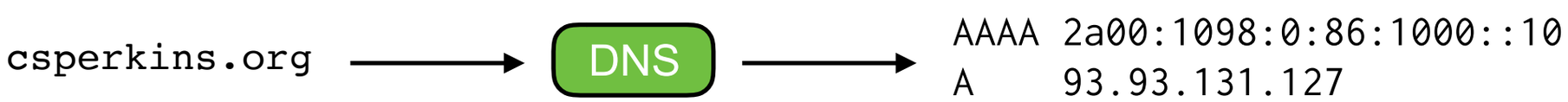
- DNS also includes Generic Top-level Domains (gTLDs)
  - Core set of gTLDs:
    - `.com`, `.org`, `.net` – unrestricted use
    - `.edu` – higher educational organisations (restricted use; primarily US-based)
    - `.mil` – US military
    - `.gov` – US government
    - `.int` – International Treaty Organisations (United National, Interpol, NATO, Red Cross, …)
  - ICANN has since massively expanded the set of gTLD registrations
    - e.g., `.scot` is a gTLD

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

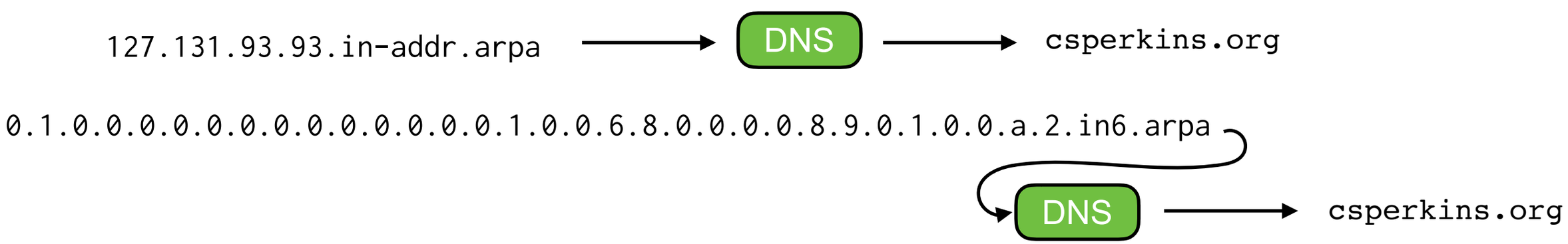
Top-Level Domains (4/4)

- The `.arpa` domain is a historical relic
  - Used in the transition from the ARPANET – the precursor to the Internet – to the Internet
  - It has one current use: **reverse DNS**

- **Forward DNS lookup:**



- **Reverse DNS lookup:**



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

Internationalised DNS

- DNS names should be available in any language
  - Initial TLDs and sub-domains were in ASCII, but UTF-8 ought to be allowed
  - DNS protocol **in principle** should work with UTF-8 name – in practice it doesn't, due to protocol ossification
- Internationalised DNS works around this by translating non-ASCII names into ASCII for encoding in the DNS
  - Punycode [https://datatracker.ietf.org/doc/rfc3492/]
    - Encodes **any** unicode text as a sequence of ASCII letters, digits, and hyphens
    - `München` → `Mnchen-3ya`
    - `Bahnhof München-Ost` → `Bahnhof Mnchen-Ost-u6b`
  - Internationalised DNS names use Punycode, prefixed with `xn--`
    - e.g., the name part of `http://日本語.jp/` is represented in the DNS as `xn--wgv71a119e.jp`

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

The DNS Root (1/3)

- ICANN decides the set of legal top-level domains – the root servers then advertise the name servers for these domains

- What are the root servers?
  - The set of 13 servers that advertise the name servers for the top level domains
    - a.root-servers.net → m.root-servers.net
  - Also have well-known IPv4 and IPv6 addresses
- Why 13 servers?
  - Want to be able to ask a DNS resolver to return a list of the root servers
  - Old versions of DNS had a size limit on the relies → 13 root servers was all that fit

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

The DNS Root (2/3)

- Who operates the root servers?

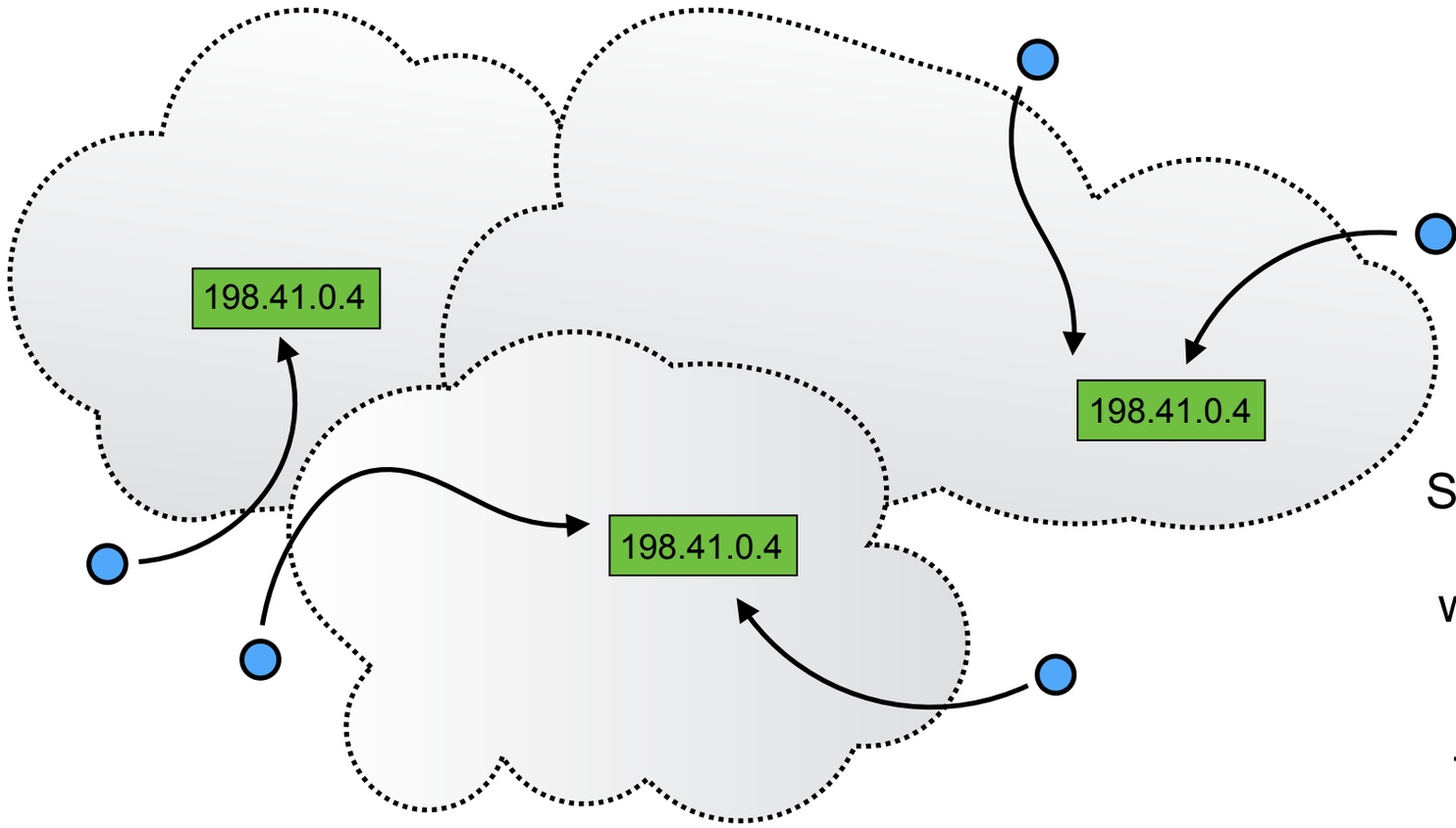
Server	IPv4 Address	IPv6 Address	Operator
A	198.41.0.4	2001:503:ba3e::2:30	Verisign
B	199.9.14.201	2001:500:200::b	USC-ISI
C	192.33.4.12	2001:500:2::c	Cogent Communications
D	199.7.91.13	2001:500:2d::d	University of Maryland
E	192.203.230.10	2001:500:a8::e	NASA Ames Research Center
F	192.5.5.241	2001:500:2f::f	Internet Systems Consortium
G	192.112.36.4	2001:500:12::d0d	US Defense Information Systems
H	198.97.190.53	2001:500:1::53	US Army Research Lab
I	192.36.148.17	2001:7fe::53	Netnod
J	192.58.128.30	2001:503:c27::2:30	Verisign
K	193.0.14.129	2001:7fd::1	RIPE NCC
L	199.7.83.42	2001:500:9f::42	ICANN
M	202.12.27.33	2001:dc3::35	WIDE Project

- Heavily US-based, for historical reasons
- The set of root servers cannot, in practice, be changed – it's too widely known

cc BY ND

The DNS Root (3/3)

- There are not really 13 servers → **anycast routing** (lecture 9)



Same IP address advertised from multiple places in the network – which replica you reach depends where you're located

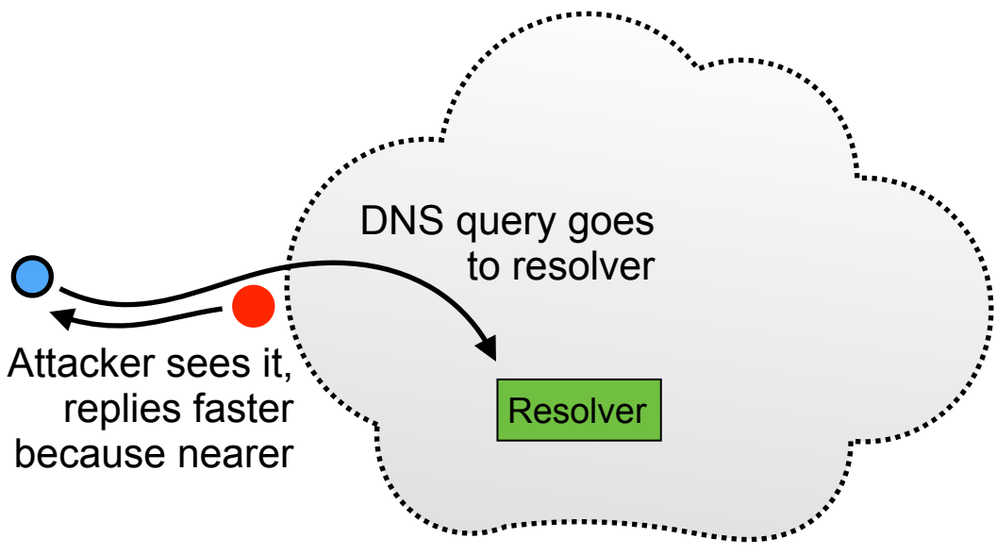
There are 13 IP addresses used by the root servers, but many more physical servers

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

DNS Security (1/2)

- **DNS has historically been completely insecure**
  - Responses delivered via unencrypted and unauthenticated protocol
  - Responses include no digital signature to verify authenticity of data

- Trivial to eavesdrop on who is looking up what name
- Trivial for on-patch attackers, or malicious resolvers, to forge replies



e.g., possible if victim and attacker are in the same cafe, using insecure Wi-Fi

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

DNS Security (2/2)

- Two approaches to securing DNS:
  - Transport security
    - Make DNS requests, and receive replies, over TLS (or some other secure channel)
    - Requests are responses are encrypted, so can't be understood or modified by attacker
    - If you trust the resolver, this protects against attack
  - Record security – **DNSSEC**
    - Add a digital signature to DNS responses that client can verify to check the data is valid
      - ICANN signs the root zone
      - Root servers sign information they provide about TLDs
      - TLDs sign information they provide about sub-domains
      - …
    - Allows a client to verify signatures back to the root, providing a chain of trust to demonstrate ownership of a domain – protects against malicious resolvers
    - Makes extensive use of public key cryptographic techniques – details are complex
    - Implemented, but not widely used
- **Need both transport and record security for fully secure DNS**

CC BY ND

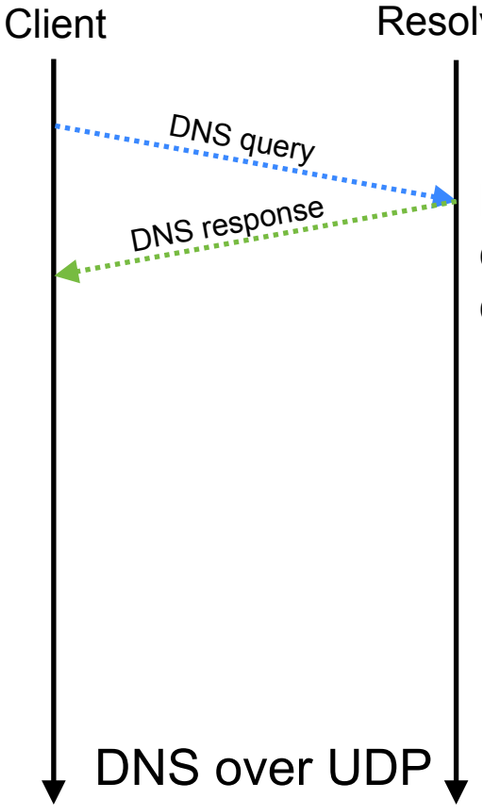
Methods for DNS Resolution

- DNS over UDP
- DNS over TLS
- DNS over HTTPS
- DNS over QUIC

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

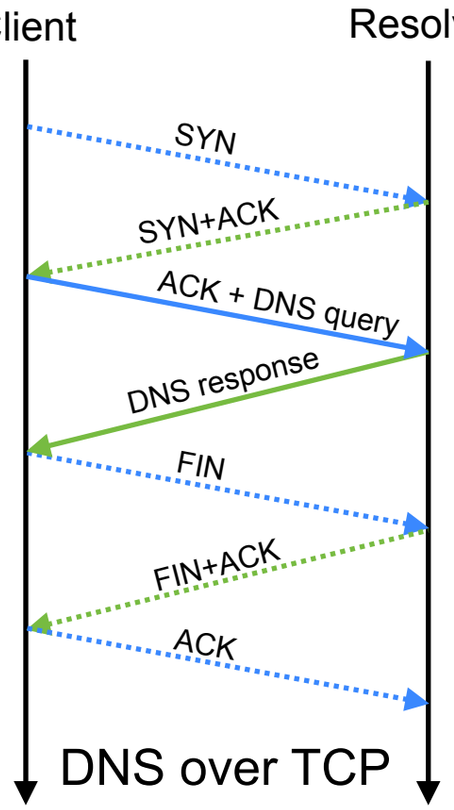
DNS Over UDP (1/4)

- DNS queries generally made over UDP port 53
  - Requests and responses are generally small enough to fit into a single packet
  - TCP reliability isn't needed – if no answer, retransmit the request
  - Congestion control isn't needed – can't adjust the rate you send a single packet



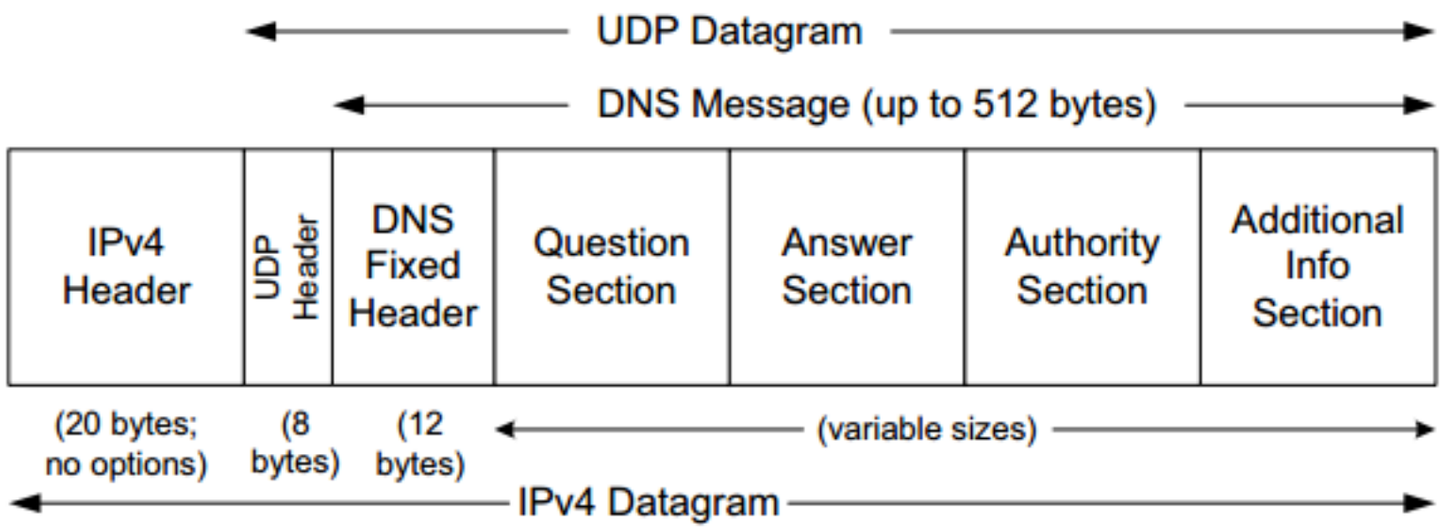
DNS query over UDP completes in one RTT – one packet for request, one for response

DNS over TCP would be much slower; initial and final TCP handshakes take several RTT and send extra packets



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

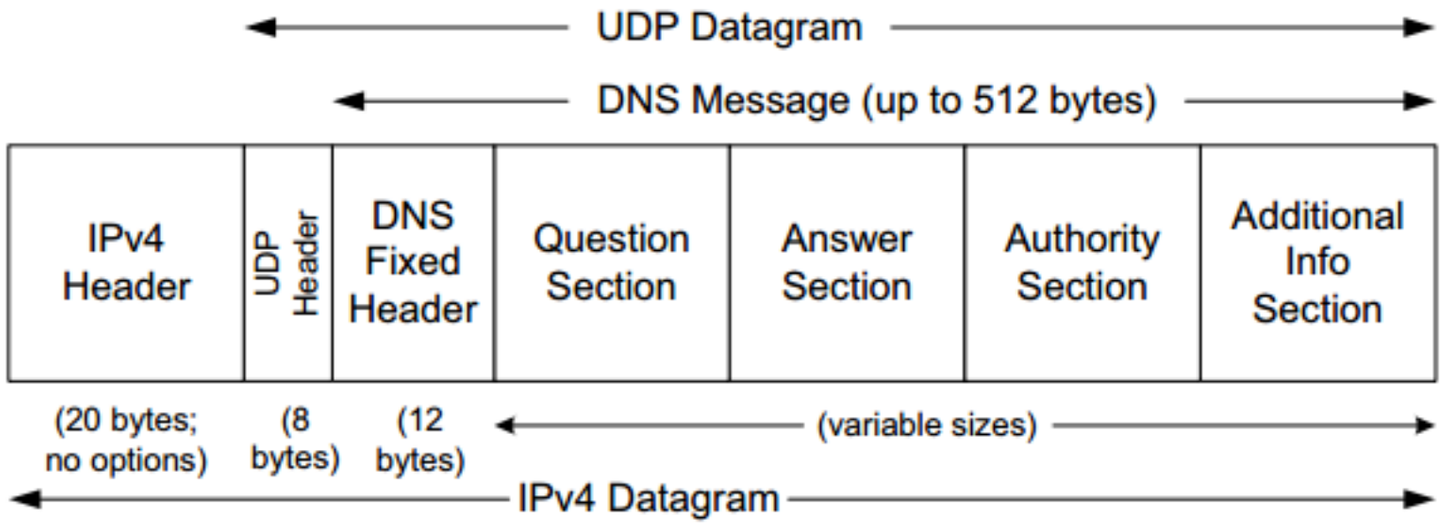
DNS Over UDP (2/4)



- Question section:
  - List of domain names and requested record type
    - e.g., what is the `AAAA` record for domain `csperkins.org`
  - Can include more than one question

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

DNS Over UDP (3/4)



- Answer, authority, and additional information sections:
  - List of domain names and record type, record data, and time-to-live
  - Answer section answers a question made in a previous request
    - e.g., the `AAAA` record for domain csperkins.org is 2a00:1098:0:86:1000::10 and it's valid for 1 hour
  - Authority describes where the answer came from

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

DNS Over UDP (4/4)

```
[stlinux02] > dig csperkins.org

; <<>> DiG 9.11.4-P2-RedHat-9.11.4-9.P2.el7 <<>> csperkins.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51409
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;csperkins.org.         IN  A

;; ANSWER SECTION:
csperkins.org.      2681   IN  A   93.93.131.127

;; AUTHORITY SECTION:
csperkins.org.      78278  IN  NS  ns2.mythic-beasts.com.
csperkins.org.      78278  IN  NS  ns1.mythic-beasts.com.

;; ADDITIONAL SECTION:
ns1.mythic-beasts.com.  70870  IN  A   45.33.127.156
ns2.mythic-beasts.com.  157301 IN  A   93.93.128.67
ns1.mythic-beasts.com.  70870  IN  AAAA   2600:3c00:e000:19::1
ns2.mythic-beasts.com.  157301 IN  AAAA   2a00:1098:0:80:1000::10

;; Query time: 0 msec
;; SERVER: 130.209.244.1#53(130.209.244.1)
;; WHEN: Wed Mar 04 18:26:53 GMT 2020
;; MSG SIZE  rcvd: 199
```

- The `dig` tool on the Linux servers can perform DNS queries
- Prints the contents of the response packet

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

DNS Over TLS (DoT)

- DNS over UDP is insecure
  - The packets are not encrypted or authenticated
  - Devices on the path between client and resolver can see DNS queries and responses – and can forge responses

- DNS over TLS solves this problem
  - DNS client opens a TCP connection to the resolver (port 853)
  - DNS client and resolver negotiate a TLS 1.3 session on the TCP connection
  - DNS client sends query, and receives response, over the TLS connection
    - DNS over TLS messages are formatted exactly the same as DNS over UDP, and contain exactly the same information – only difference is that they're sent over TLS not UDP
    - Slower and higher overhead than DNS over UDP – due to need to negotiate TCP and TLS – but more secure

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

DNS over HTTPS (DoH)

- DoH allows a client to send queries to a resolver using HTTPS
  - Can use with GET or POST methods in HTTPS

```
GET /dns-query?dns=AAABAAABAAAAAAAAA3d3dwdleGFtcGxlA2NvbQAAAQAB HTTP/1.1
Accept: application/dns-message
```

Base-64 encoded version of the data that would be sent in a DNS-over-UDP request

Uses /dns-query as the URL path

```
POST /dns-query HTTP/1.1
Accept: application/dns-message
Content-type = application/dns-message
Content-length = 33

<33 bytes of UDP query, exactly as if sent in a UDP packet>
```

- HTTP response has `Content-Type: application/dns-message` and contains the exact same data that would be sent in a UDP-based DNS response

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

DNS over QUIC (DoQ)

- Work in progress to define DNS over QUIC:
  - https://datatracker.ietf.org/doc/draft-huitema-quic-dnsoquic/
  - Same principle as DNS over TLS:
    - Client opens a QUIC connection to the resolver
      - Negotiates TLS security as part of the connection setup
    - Sends the request and receives the response over that connection
      - Requests and response contain exactly the same data as DNS over UDP – just sent via QUIC

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

Methods for DNS Resolution

- Increasingly many ways of making DNS queries:
  - DNS over UDP
  - DNS over TLS
  - DNS over HTTPS
  - DNS over QUIC
- The contents of the query and the response are **identical** in all cases
  - They change how the query is delivered to the resolver and how the response is returned, but not the contents of the messages
  - They change the security guarantees provided
  - They potentially gives clients more flexibility to query different resolvers

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

The Politics of Names

- Choice of DNS resolver
- Intellectual property and the DNS
- What domains should exist?
- Who controls the DNS root?

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

Implications of Choice of DNS Resolver (1/3)

- How is the DNS resolver chosen?
  - When connecting to a network, most hosts use DHCP (dynamic host configuration protocol) to discover network settings and configuration
    - DHCP tells the host what DNS resolver to use for the network
    - If a host has multiple network interfaces, it may use a different DNS resolver for each
      - The `getaddrinfo()` call takes a `hints` parameter, that can include the local IP address
      - e.g., consider phone connected to 4G cellular network and private company WiFi – the WiFi might make available names of internal company services that are not accessible to the public
  - Possible to configure the DNS resolver manually
    - e.g., to talk to Google public DNS resolver (IPv4 address 8.8.8.8)

- DNS resolution has typically been a system-wide service
  - Operating system implements a DNS resolution service (DNS-over-UDP, sometimes also DNS-over-TLS) and all DNS queries use that service
  - **DoH is changing this** – JavaScript web applications can now easily perform DNS queries via any HTTP website

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

Implications of Choice of DNS Resolver (2/3)

- Is flexibility for each application to perform DNS queries differently a concern?

No

Applications should have the ability to use a secure DNS server they trust to avoid phishing attacks, malware, monitoring, etc.

Why should network operators be able to see DNS queries and modify responses? This is a privacy and security risk

Yes

Network operators filter DNS responses to block access to malicious sites and prevent malware spreading – allowing applications to bypass this is a security risk

Network operators filter DNS responses to enforce legal or societal constraints – e.g., UK Internet Watch Foundation publishes a DNS block list to prevent access to sites hosting child sexual abuse material

- Giving applications ability to securely access arbitrary DNS servers allows them to avoid local observation and/or filtering of DNS traffic; with both positive and negative consequences

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

Implications of Choice of DNS Resolver (3/3)

- Can a network restrict the choice of DNS resolver?
  - Firewalls can block access to DNS-over-UDP and DNS-over-TLS resolvers
    - Block access to UDP port 53
    - Block access to TCP port 853
    - For all destination IP addresses except those of allowed DNS resolvers
  - Difficult to block DNS-over-HTTPS
    - Network cannot distinguish DNS-over-HTTPS from any other traffic over HTTPS
    - May be able to tell from the destination IP address
      - e.g., Google use IPv4 address 8.8.8.8 for public DoH services, but not other traffic
    - But if a web server handles a mix of web and DNS traffic over HTTPS, cannot block one without blocking the other
    - Many ISPs and governments concerned that DoH prevents use of DNS as a control point

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

Intellectual Property and the DNS

- Intellectual property is managed on a national basis
  - e.g., a company might own a trademark in the UK while a different company owns the same trademark in the Republic of Ireland
  - Which company owns *`trademark`*`.ie` and which owns *`trademark`*`.co.uk` is a straightforward legal question
  - Which company owns *`trademark`*`.com` is likely harder to decide
    - Especially since `.com` is operated by a US-based organisation, and a third company might own the trademark in the US

- A ccTLD clearly operates under legal regime of a particular country
- Use of a gTLD might lead to legal complications

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

What Domains Should Exist?

- Should a particular gTLD be allowed to exist?
  - e.g., should `.xxx` exist to host "adult" content?
    - If so, who gets to decide what content should (must?) sit within that gTLD?
    - Different countries have very different norms and standards in this area
  - Who controls what TLDs ICANN permits?
- Should a particular subdomain be allowed to exist?
  - Significant differences around freedom of speech and permissible topics in different parts of the world
  - A ccTLD can enforce local conventions and rules
  - A gTLD is potentially accessible worldwide – what should it allow?
    - If a particular country/group finds a site objectionable, should it be taken down?
    - If country X decides particular content is illegal, but it is legal in country Y, should a gTLD operated out of country Y but accessible in country X permit such content?
    - e.g., Holocaust denial is illegal in Germany but not in the US – should a `.com` permit such content?

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

Who Controls the Root Servers?

Server	IPv4 Address	IPv6 Address	Operator
A	198.41.0.4	2001:503:ba3e::2:30	Verisign
B	199.9.14.201	2001:500:200::b	USC-ISI
C	192.33.4.12	2001:500:2::c	Cogent Communications
D	199.7.91.13	2001:500:2d::d	University of Maryland
E	192.203.230.10	2001:500:a8::e	NASA Ames Research Center
F	192.5.5.241	2001:500:2f::f	Internet Systems Consortium
G	192.112.36.4	2001:500:12::d0d	US Defense Information Systems Agency
H	198.97.190.53	2001:500:1::53	US Army Research Lab
I	192.36.148.17	2001:7fe::53	Netnod
J	192.58.128.30	2001:503:c27::2:30	Verisign
K	193.0.14.129	2001:7fd::1	RIPE NCC
L	199.7.83.42	2001:500:9f::42	ICANN
M	202.12.27.33	2001:dc3::35	WIDE Project

- DNS root servers are mostly controlled by US-based organisations
  - Is this a national security risk for other countries?
- Should the root servers be controlled by a broader mix of countries?
  - If so, who gets to decide – ICANN? The United Nations?
  - Is there a benefit in a country controlling a DNS root server?
  - Is there a benefit in a country controlling a gTLD server?
    - e.g., is it a geopolitical or economic benefit to the US that `.com` is based there?

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

Should There Be a Single DNS Root?

- Should all TLDs be accessible from everywhere?
  - Should there be a single global DNS?
  - Should the same name always resolve to the same site?
    - With global content distribution networks, how can you tell?
  - Should different countries be allowed to filter DNS?
    - If so, how should such restrictions be implemented?
    - It is difficult to distinguish modifications to DNS responses made to conform to government-mandated filtering requirements from those made by malware, phishing attacks, etc. – is this a feature or a bug?

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

Naming and the Tussle for Control

- What is the DNS?
- How are DNS queries made?
- Who controls the names?

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