DNS

Availability, scalability, and performance

DNS: Availability, scalability, and performance

- DNS should be *highly available*
  - Expectation Internet would be difficult/impossible to use without it
- DNS should be *highly scalable*
  - Expectation that it would be used heavily
- DNS should be *highly performant*
  - Above expectations make this crucial!

- All sort of intertwined because there's basically one trick to accomplishing them

  - Add more servers!

  - Done in four different ways

DNS: Availability, scalability, and performance

- Name servers for different domains are independent
  - Just because berkeley.edu fails doesn't mean mtholyoke.edu fails
  - Just because .edu fails doesn’t mean .com fails

  - *Not just one server: servers per domain*

  - Availability implication:
    - berkeley can fail; mtholyoke is fine
  - Scalability implication:
    - No one server needs to know all info for all domains
  - Performance implication:
    - Even if someone else’s domain is going viral, yours is fine

DNS: Availability, scalability, and performance

- Domains have at least two name servers
  - ns1.umass.edu
  - ns2.umass.edu
  - ns3.umass.edu
  - ns.mtholyoke.edu

  } Authoritative name servers for mtholyoke.edu

  - *Not just one server per domain: multiple servers per domain*

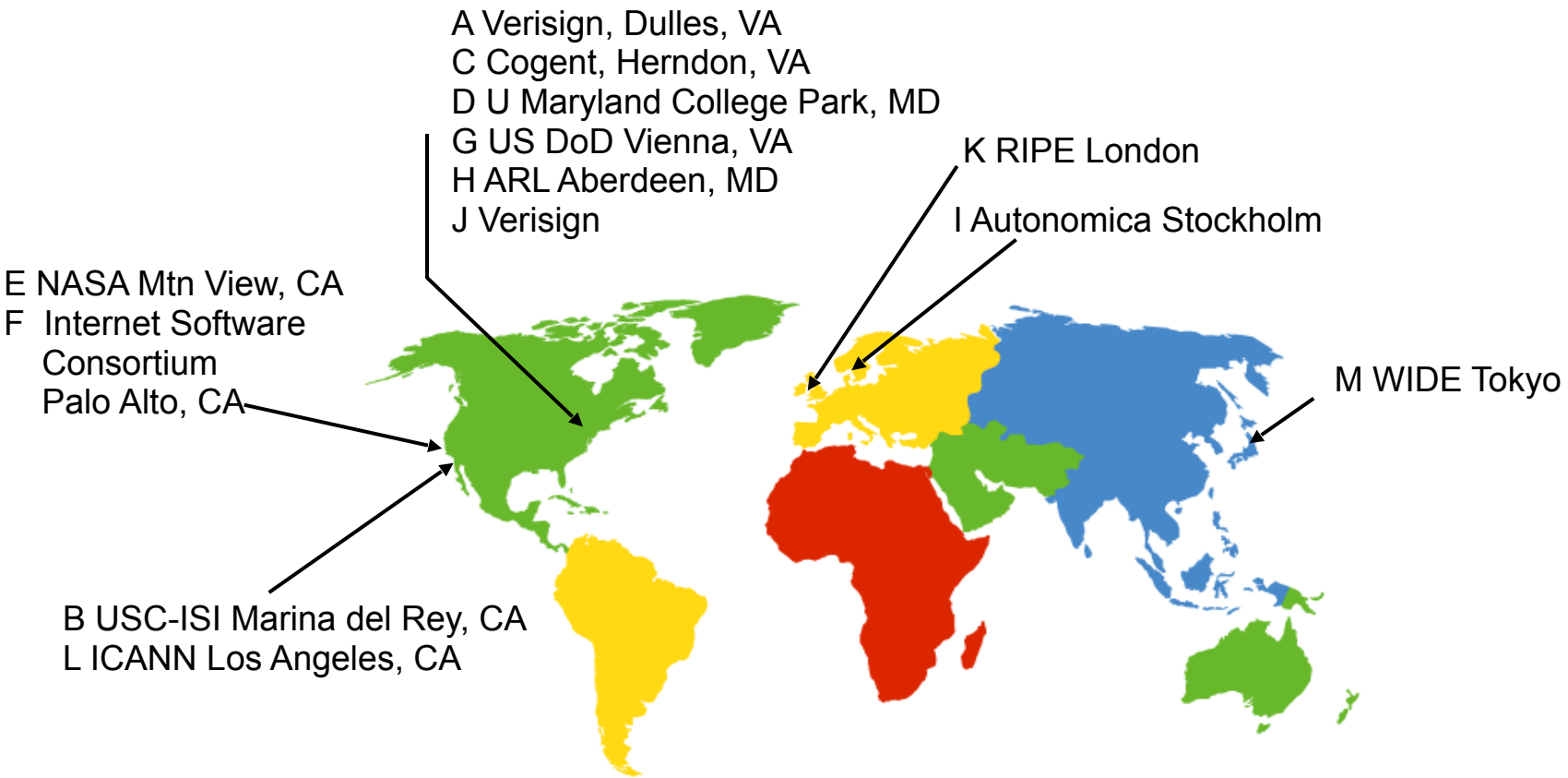
  - Availability implication:
    - One server crashes, you've got others
  - Scalability/performance implication
    - Queries get spread across; limits latency due to load

DNS: Looking at the Root Servers

- Already mentioned that there were 13 root name servers…

- a.root-servers.net 198.41.0.4
- b.root-servers.net 199.9.14.201
- c.root-servers.net 192.33.4.12
- d.root-servers.net 199.7.91.13
- e.root-servers.net 192.203.230.10
- f.root-servers.net 192.5.5.241
- g.root-servers.net 192.112.36.4
- h.root-servers.net 198.97.190.53
- i.root-servers.net 192.36.148.17
- j.root-servers.net 192.58.128.30
- k.root-servers.net 193.0.14.129
- l.root-servers.net 199.7.83.42
- m.root-servers.net 202.12.27.33

DNS: Looking at the Root Servers



DNS: Looking at the Root Servers

- 4.5 billion Internet users
- Are 13 root name servers enough?

- .. probably not
  - “j-root” alone got 66K queries per second in April 2018
  - 15us per query?

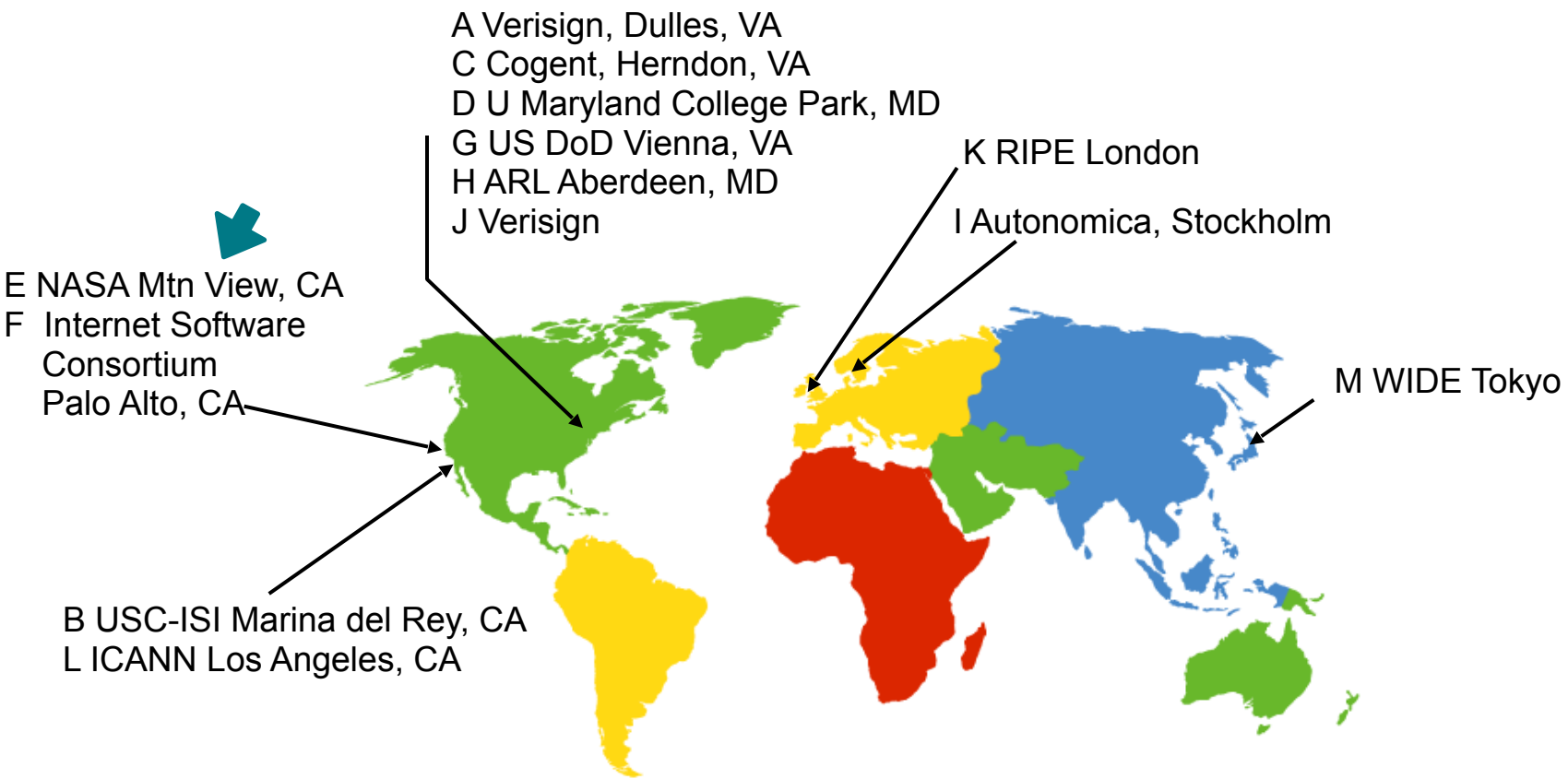
- So what's the trick?

- There are actually 162 duplicates of j-root in different places!
  - There’s actually about a thousand total root servers as of 2020

DNS: J-root server locations

Amsterdam, NL
Ashburn, US
Athens, GR
Atlanta, US
Bangalore , IN
Bangkok, TH
Barcelona, ES
Battle Creek, US
Beijing, CN
Belgrade, RS
Belo Horizonte, BR
Berlin, DE
Bettembourg, LU
Bloomington, US
Boston, US
Brasilia, BR
Bratislava, SK
Brisbane, AU
Bucharest, RO
Buenos Aires, AR
Cajazeiras, BR
Calgary, CA
Cape Town, ZA
Cebu City, PH
Chicago, US
Cochabamba, BO
Colombo, LK
Cordoba, AR
Dar Es Salaam, TZ
Denver, US
Des Moines, US
Dhaka, BD
Djibouti City, DJ
Eau Claire, US
Edinburgh, GB
Fayetteville, US
Frankfurt , DE
Frankfurt Griesheim, DE
Frankfurt, DE
Fremont, US
Geneva, CH
Guarabira, BR
Gurgaon, IN
Halifax, CA
Hong Kong, HK
Honolulu, US
Istanbul, TR
Jakarta, ID
Johannesburg, ZA
Juazeirinho, BR
Kansas City, US
Kathmandu, NP
Kaunas, LT
Kiev, UA
Kigali City, RW
Klagenfurt, AT
Kolkata, IN
Kuala Lumpur, MY
Lagos, NG
Leeds, GB
Lisbon, PT
Ljubljana, SI
London , GB
London, GB
Los Angeles, US
Luanda, AO
Madrid, ES
Male, MV
Melbourne, AU
Miami, US
Milan, IT
Montgomery, US
Moscow, RU
Mumbai, IN
Muscat, OM
Nairobi, KE
New Castle, US
New Delhi, IN
New York, US
Orlando, US
Oslo, NO
Oulu, FI
Paris, FR
Perth, AU
Philadelphia, US
Pirassununga, BR
Plano, US
Portland, US
Prague, CZ
Princes Town, TT
Quezon City, PH
Reno, US
Reykjavik, IS
Richmond, US
Riga, LV
Rio De Janeiro, BR
Rome, IT
Saint Petersburg, RU
Salzburg, AT
San Francisco, US
San Jose, CR
San Jose, US
San Juan, PR
Santiago, CL
Seattle, US
Seoul, KR
Singapore, SG
Sofia, BG
St. George, US
Stockholm, SE
Sydney, AU
Tallinn, EE
Tampa, US
Tamuning, GU
Tel Aviv, IL
Tokyo, JP
Turin, IT
Vancouver, CA
Vilnius, LT
Warsaw, PL
Wellington, NZ
Winnipeg, CA
Yerevan, AM
Zagreb, HR
Zurich, CH

DNS: Looking at the Root Servers



DNS: E-root server locations

Accra, GH
Adelaide, AU
Amsterdam, NL
Antananarivo, MG
Arica, CL
Arusha, TZ
Ashburn, US
Athens, GR
Atlanta, US
Auckland, NZ
Baghdad, IQ
Baltimore, US
Bangkok, TH
Banjul, GM
Barcelona, ES
Beirut, LB
Belgrade, RS
Berlin, DE
Beverly, US
Blantyre, MW
Bogota, CO
Boston, US
Brisbane, AU
Brussels, BE
Bucharest, RO
Budapest, HU
Buenos Aires, AR
Buffalo, US
Burbank, US
Calgary, CA
Cape Town, ZA
Castries, LC
Cebu, PH
Charlotte, US
Chennai, IN
Chicago, US
Chittagong, BD
Colombo, LK
Columbus, US
Copenhagen, DK
Cork, IE
Curitiba, BR
Dakar, SN
Dallas, US
Dar Es Salaam, TZ
Denver, US
Detroit, US
Dhaka, BD
Djibouti, DJ
Doha, QA
Dubai, AE
Dublin, IE
Durban, ZA
Dusseldorf, DE
Edinburgh, GB
Enfidha, TN
Fortaleza, BR
Frankfurt, DE
Geneva, CH
Gothenburg, SE
Guayaquil, EC
Halifax, CA
Hamburg, DE
Helsinki, FI
Hong Kong, HK
Honolulu, US
Houston, US
Hyderabad, IN
Indianapolis, US
Istanbul, TR
Jacksonville, US
Jakarta, ID
Johannesburg, ZA
Johor Bahru, MY
Kampala, UG
Kansas City, US
Kathmandu, NP
Kigali, RW
Kingston, JM
Klagenfurt, AT
Kuala Lumpur, MY
Kuwait City, KW
Kyiv, UA
La Paz, BO
Lagos, NG
Las Vegas, US
Lausanne, CH
Leeds, UK
Lima, PE
Lisbon, PT
London, GB
Los Angeles, US
Luanda, AO
Luxembourg City, LU
Lyon, FR
Macau, MO
Madrid, ES
Manama, BH
Manchester, GB
Manchester, UK
Manila, PH
Maputo, MZ
Mar Del Plata, AR
Marseille, FR
McAllen, US
Medellin, CO
Melbourne, AU
Memphis, US
Mexico City, MX
Miami, US
Milan, IT
Minneapolis, US
Mombasa, KE
Moncton, CA
Monrovia, LR

DNS: E-root server locations

Montgomery, US
Montreal, CA
Moscow, RU
Mountain View, US
Mumbai, IN
Munich, DE
Muscat, OM
Nagpur, IN
Nairobi, KE
Nashville, US
New Delhi, IN
New York, US
Newark, US
Norfolk, US
Noumea, NC
Omaha, US
Osaka, JP
Oslo, NO
Palo Alto, US
Panama City, PA
Paris, FR
Perth, AU
Philadelphia, US
Phnom Penh, KH
Phoenix, US
Pittsburgh, US
Port Louis, MU
Port Vila, VU
Port au Prince, HT
Port of Spain, TT
Port-Au-Prince, HT
Portland, US
Porto Alegre, BR
Posadas, AR
Prague, CZ
Queretaro, MX
Quito, EC
Ramallah, PS
Rene Mouawad, LB
Reno, US
Reykjavik, IS
Richmond, US
Riga, LV
Rio De Janeiro, BR
Riyadh, SA
Rome, IT
Rosario, AR
Sacramento, US
Saint George, US
Saint-Denis, RE
Saldanha, ZA
Salt Lake City, US
San Diego, US
San Francisco, US
San Jose, CR
San Jose, US
Santa Ana, US
Santiago, CL
Sao Paulo, BR
Saskatoon, CA
Seattle, US
Seoul, KR
Seoul, ZA
Siegerland, DE
Singapore, SG
Sofia, BG
St. Georges, GD
St. Louis, US
Stockholm, SE
Sumbe, AO
Sydney, AU
Taipei, TW
Tallahassee, US
Tallinn, EE
Tampa, US
Tampere, FI
Tegucigalpa, HN
Tel Aviv, IL
Thessaloniki, GR
Tokyo, JP
Toronto, CA
Tunis, TU
Turin, IT
Ulaanbaatar, MN
Vancouver, CA
Vienna, AT
Vilnius, LT
Warsaw, PL
Washington, US
Wellington, NZ
Willemstad, CW
Windhoek, NA
Winnipeg, CA
Yerevan, AM
Zagreb, HR
Zurich, CH

DNS: Looking at the Root Servers

- There are duplicates of the a,b,c,d,… root servers in different places!

- Each “X-root” duplicate has the same IP address!
- Same address advertised through BGP at each duplicate
- BGP will always just deliver to the “closest” one (subject to BGP policy)

    - This is called *anycast*
        - Delivers to *any* one of the destinations
        - Contrast with *unicast* (our focus until now -- delivers to *single* dest.)

        - .. it “just works” — no change to routing is needed

DNS: Looking at the Root Servers

- Anycast “just works”…
- But does it have any drawbacks?

    - Like multihomed prefixes, anycast prefixes can’t be aggregated
        - Can't just use it for everything!
        - But maybe really important things like DNS!

    - Doesn’t always work well with TCP…
        - Changes to BGP or client mobility may mean your connection suddenly shifts to a different server that knows nothing about it!
        - Luckily… DNS is usually UDP / stateless!

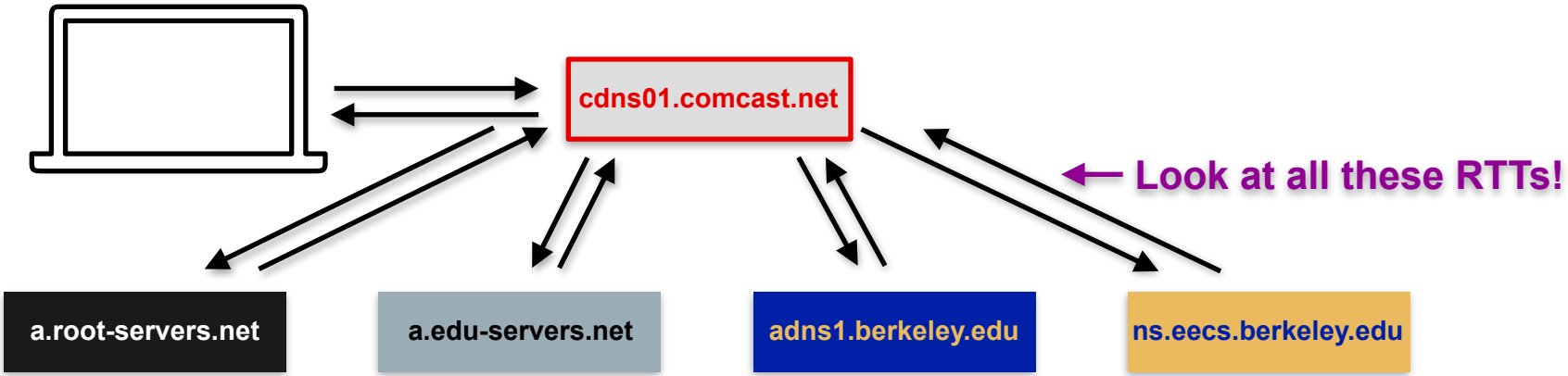
DNS: Looking at the Root Servers

- Availability implication:
  - If network partitioned, BGP will automatically route to reachable server
- Scalability implication:
  - Further spreads load across servers
- Performance implication:
  - Reduce RTT to server because BGP shouldn’t pick terrible path

DNS: Availability, scalability, and performance

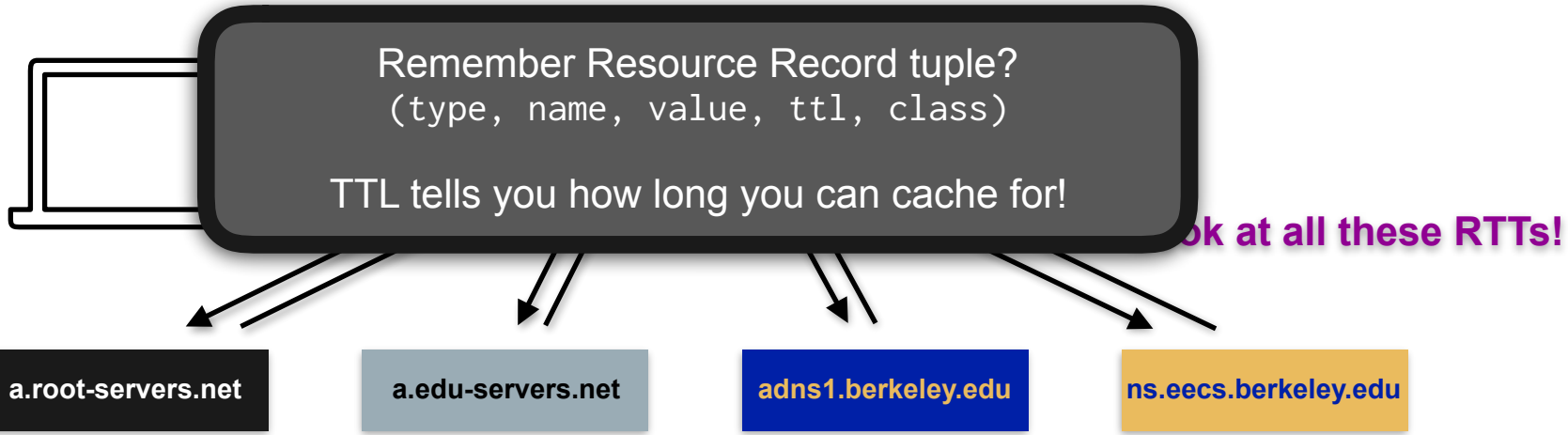
- Remember this?
- Query procedure not very fast! Many RTTs!

- How can we speed it up?
  - Caching!



DNS: Availability, scalability, and performance

- Servers on bottom never see anything to cache (only their own records)
- But resolving server can cache results of resolution (and share with other hosts)
- Can actually insert *caching servers* just about anywhere!
  - Don't even do iterative lookups -- just forward request and cache reply
- .. and the host can cache too, of course!



DNS: Looking at the Root Servers

- Availability implication:
  - Even during a hiccup (e.g., while BGP converging on new path), common queries can be served out of cache

- Scalability implication:
  - Reduces load on real name servers

- Performance implication:
  - No need to do multi-RTT lookup process for common entries
  - Can cache close to clients to reduce RTT

DNS: Availability, scalability, and performance

- Summing up…

- Four ways in which we add more servers:
  - Different domains have different DNS servers
  - Servers are replicated; minimum two for each domain
  - Each root server is replicated *again* using anycast
  - Add resolving/forwarding/caching servers that cache results

- With more servers:
  - It's more likely one is close to you (faster RTT)
  - They're less likely to be heavily loaded (slow)
  - It's less likely that the only one with the data you want is down
  - The amount of data that needs to be stored on each one is lessened

DNS

A little DNS skepticism

Inquiry and doubt are essential checks against deception, self-deception, and error.
— Richard Carrier
(maybe)

DNS Skepticism

- DNS is usually presented as vital, integral aspect of the Internet
- It's in every book on the Internet
- Every network course covers it
- It has been ".. an essential component of the functionality of the Internet since 1985." (Wikipedia)

- Did we name the right things?
- Does DNS actually work well?
- Does DNS put your privacy at risk?
- Is DNS even important?

Do we name the right things?

- telnet was about logging in to a remote host
  - What was the right thing to name?
  - Hosts!

Do we name the right things?

- FTP is about transferring files
  - File already had names on machines; already had name for the file part
  - “/existing/file/name on hostname” seems reasonable

  - Is it ideal?
    - What if you move the file to another machine?
    - What if you want to replicate the file on many hosts so it’s always available? Do you even care *which* host it’s stored at?

  - Better solutions? What is the right thing to name?
    - See: Information-Centric Networking, Content-Centric Networking, and **Named Data Networking**

Do we name the right things?

- Email is about communicating between *people*
  - Remember, many users used to share the same computer!
  - Early email was between users on a single computer
  - Just needed to mail <username>
  - Easy to see extension of this to mail <username> on <remote machine>

  - Is it ideal?
    - What if the user moves to a different machine?
    - What if you have too many users for single machine to handle (gmail)?

  - Better solutions?
    - Maybe unique names for *people* independent of providers?
      - Must map to whoever your current mail provider is
      - Want some sort of accountability (for spammers)?
      - Anonymity / deniability?
      - May be a tricky design challenge?

Do we name the right things?

- What about the web?
- URLs basically are hostname plus filename
  - Make sense — early web was (underneath) much like simple file transfer
  - .. but with an important change in how it was used! (hypertext)

- With that in mind, same idea as FTP might apply…
  - Want names driven by content not by which machine they're on!

- But is the web just about transferring files?
  - Is it more about accessing services? (your banking, Facebook, …)
    - Modern services certainly aren't tied to a specific host!
    - Should we be naming services directly?

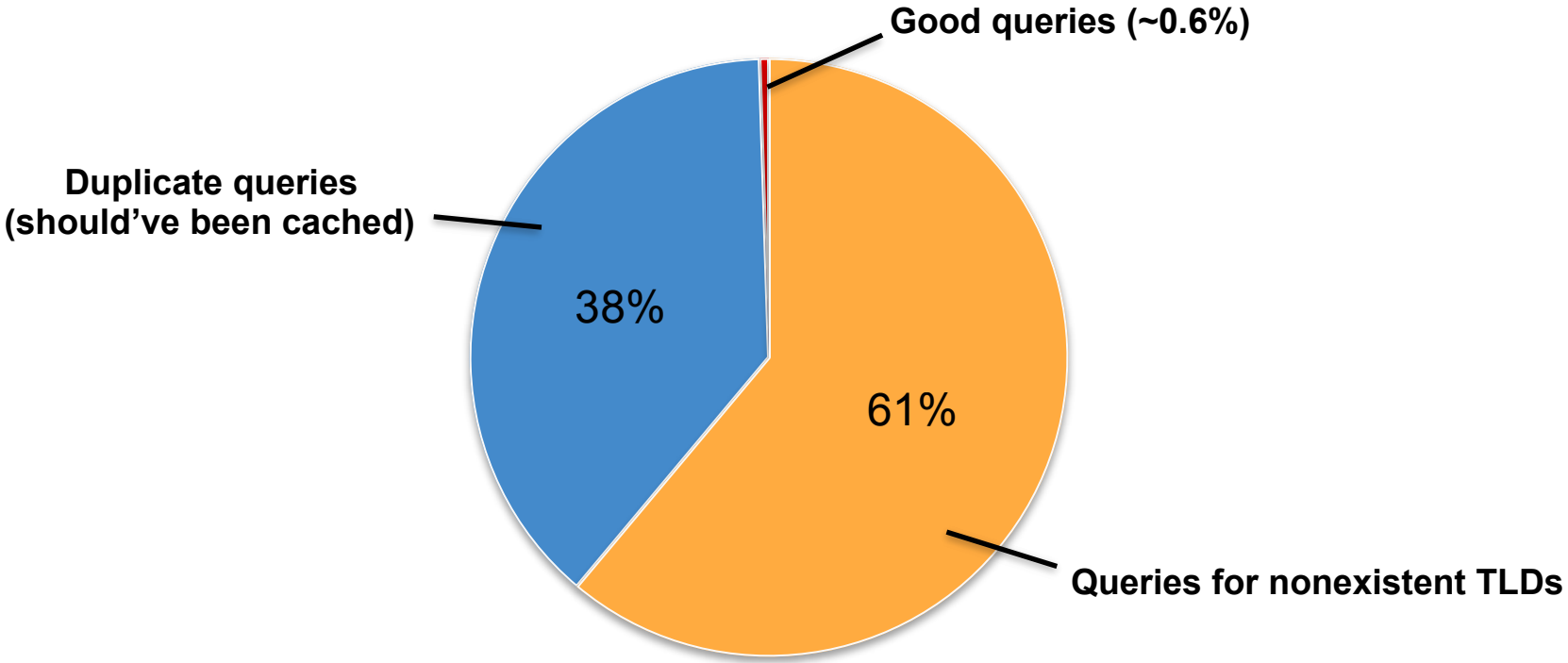
Do we name the right things?

- Naming hosts made *perfect* sense for telnet
- For files, people, and services… hosts are less good of a fit
- If we stepped back, could we design better schemes and mechanisms?
    - e.g., A highly available, scalable, performant system for addressing *people* no matter where they are or who their email provider is?

- *My* gut says yes.

Does DNS work well?

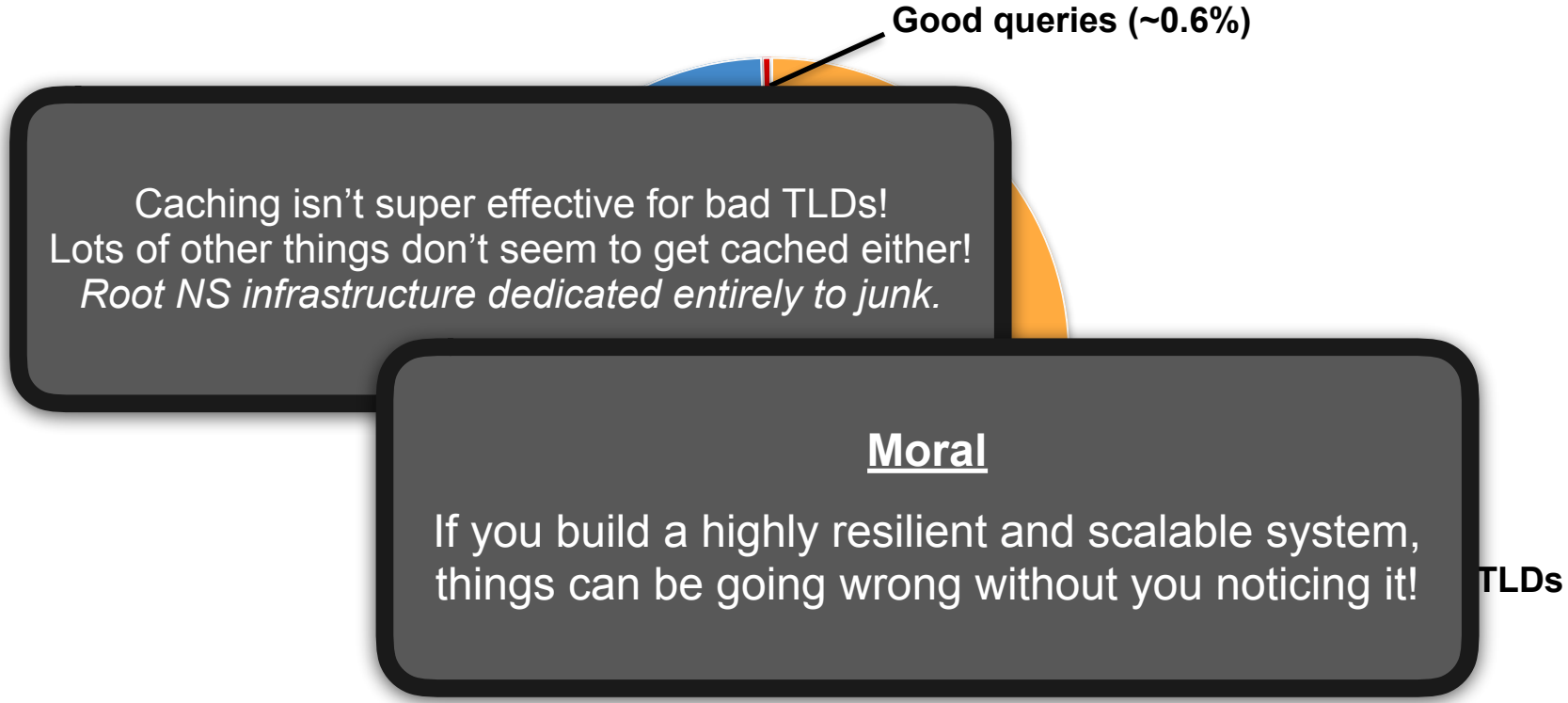
- Analyzed 5.7B queries in 24 hours at J-root



[Data from Allman 2019]

Does DNS work well?

- Analyzed 5.7B queries in 24 hours at J-root



[Data from Allman 2019]

Is DNS a privacy problem?

- If you browse sites using https, the data is encrypted
- .. but the DNS lookups aren't

- Nobody knows what you’re reading or writing, but everyone knows where you’re doing it

- Trivially easy for ISPs to just log requests at DNS server
- They don't even need to "snoop" the traffic -- you send it right to them!

Is DNS a privacy problem?

- Currently two competing projects to make DNS more private:
  - DNS over TLS
  - DNS over HTTPS

- It’s causing a fair amount of debate
- Current versions of Firefox are using DNS over HTTPS by default
  - No longer uses your own ISP
  - Uses third party for DNS resolution — Cloudflare
    - Do you know/trust them, or are you just securely giving your DNS info to an untrusted third party?

Is DNS a privacy problem?

- Currently two competing projects to make DNS more private:
  - DNS over TLS
  - DNS over HTTPS

Who needs names anyway?

- When is the last time you typed a hostname?

- Get the weather then (1992):
  - telnet rainmaker.wunderground.com
- Get the weather now:
  - Open Weather Underground app

- Finding HTTP protocol information then (1989):
  - Type http://info.cern.ch/hypertext/WWW/ into browser
  - Take links for Technical, Protocols, HyperText Transfer Protocol
- Finding HTTP protocol information now:
  - Put “http standard” in search bar of browser

Who needs names anyway?

- Is human-readability useful / secure?
- Wait, is it examplesite.com or example-site.com?
- Is it safe to take a link to welllsfargo.com?
- What about instagram.com?

- You’re probably better off searching than typing an address!

Is DNS actually important?

- DNS maps from hostnames to addresses (mostly)
  - But *people, services*, and *data* are in many ways more important than particular hosts today

- Human-supplied names…
  - Matter less today due to web search, apps, etc.

- Human-readable names…
  - Have little guarantee that you're reading what you think you're reading

- .. I'm not convinced original purpose of DNS stands the test of time!

Is DNS actually important?

- But, DNS also provides…

- Load balancing…
  - One name can map to multiple addresses to spread traffic

- An indirection layer…
  - Can keep the same name but map it to a different address
    - Useful if you move your server, switch ISPs, etc.
  - Can do the mapping dynamically — inspect source IP of DNS request, and direct to a server you think is *near* the client!

- .. very important!
  - .. but have nothing to do with the original driver of human-readability!
  - .. could you design a better system if you gave up human-readability?