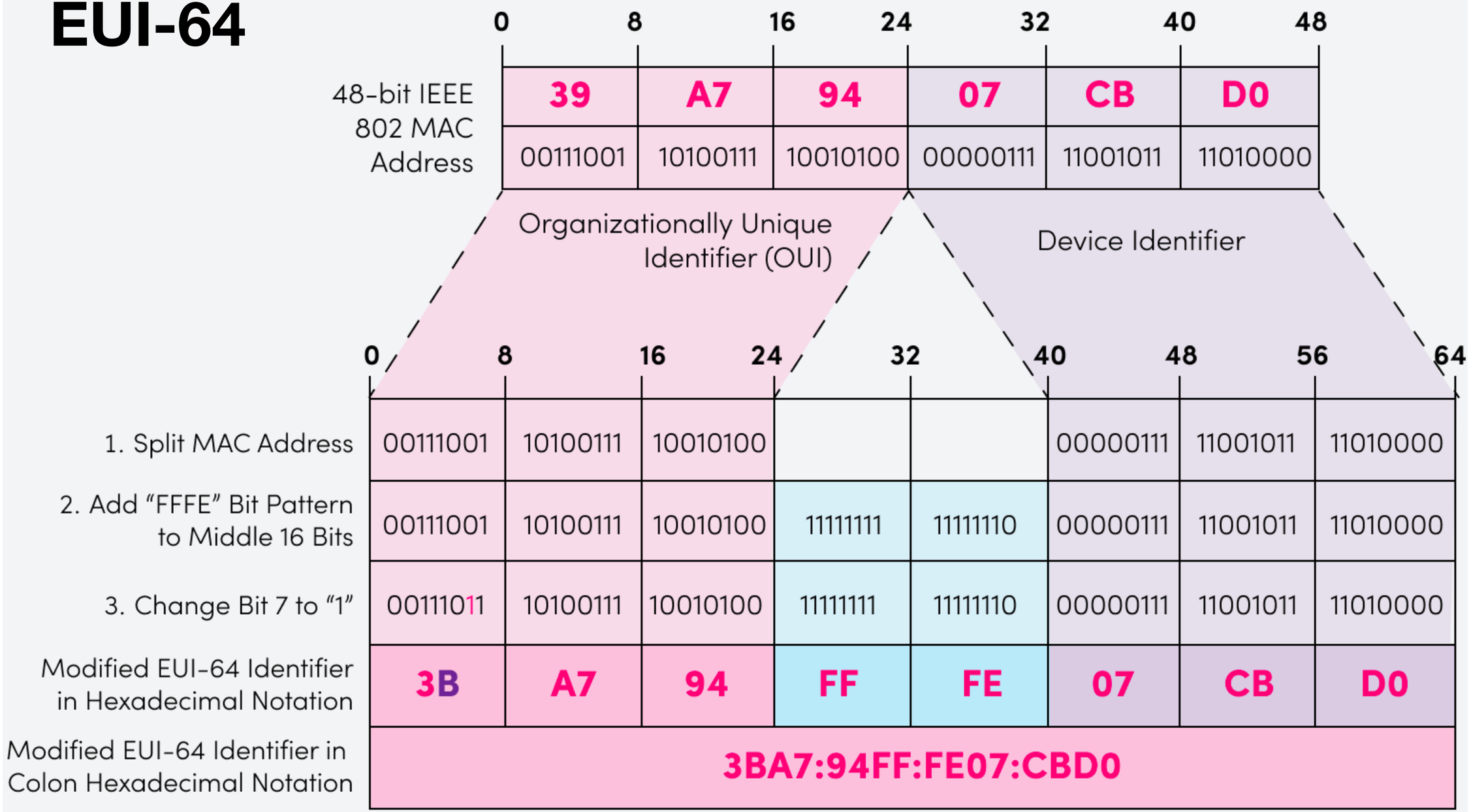
Host and Network Addressing

CS249i: The Modern Internet

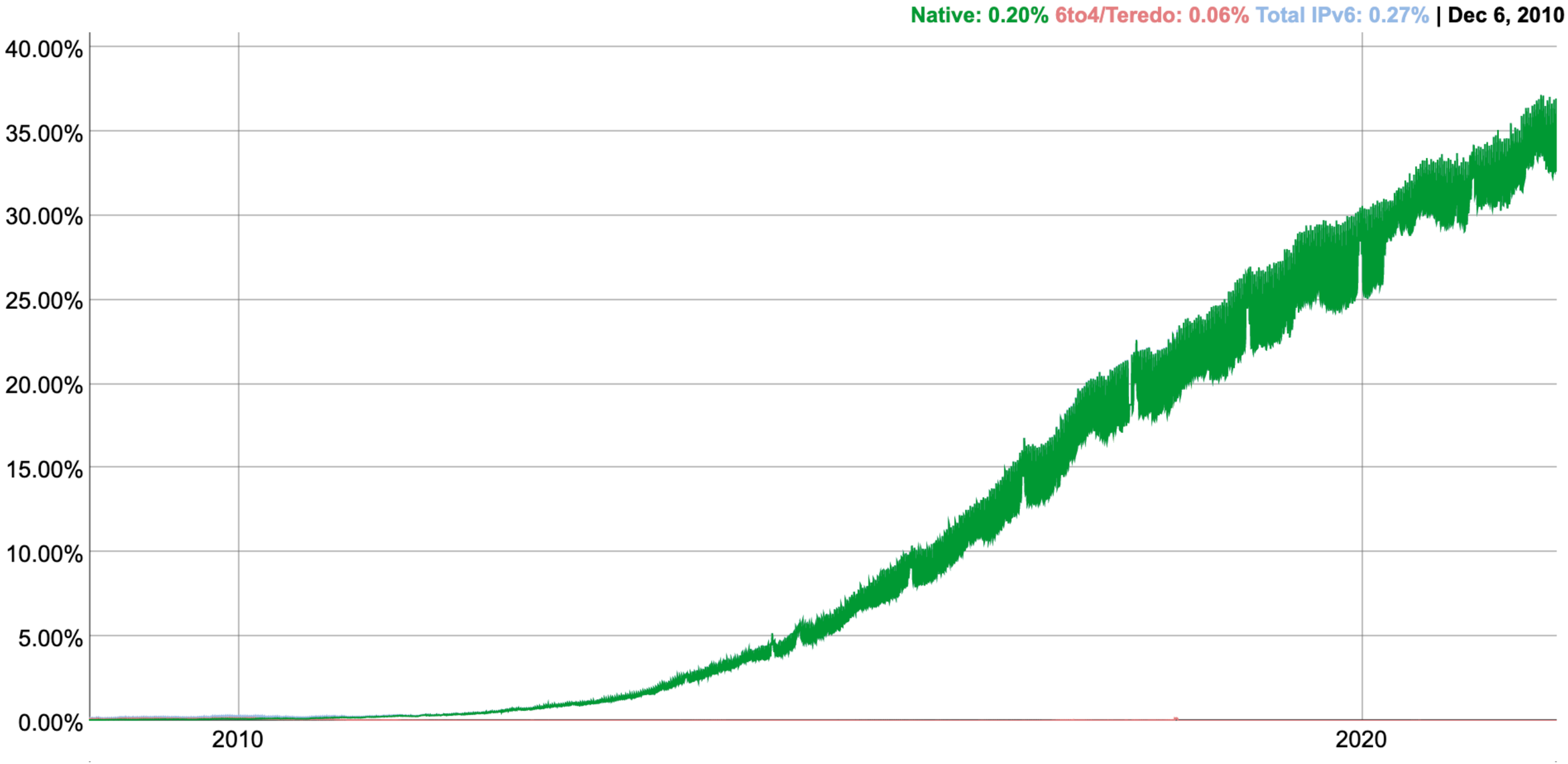
Stanford
University

EUI-64


0
8
16
24
32
40
48
48-bit IEEE 802 MAC Address
39
A7
94
07
CB
D0
00111001
10100111
10010100
00000111
11001011
11010000
Organizationally Unique Identifier (OUI)
Device Identifier
0
8
16
24
32
40
48
56
64
1. Split MAC Address
00111001
10100111
10010100
00000111
11001011
11010000
2. Add "FFFE" Bit Pattern to Middle 16 Bits
00111001
10100111
10010100
11111111
11111110
00000111
11001011
11010000
3. Change Bit 7 to "1"
00111011
10100111
10010100
11111111
11111110
00000111
11001011
11010000
Modified EUI-64 Identifier in Hexadecimal Notation
3B
A7
94
FF
FE
07
CB
D0
Modified EUI-64 Identifier in Colon Hexadecimal Notation
3BA7:94FF:FE07:CBD0

IPv6 Usage

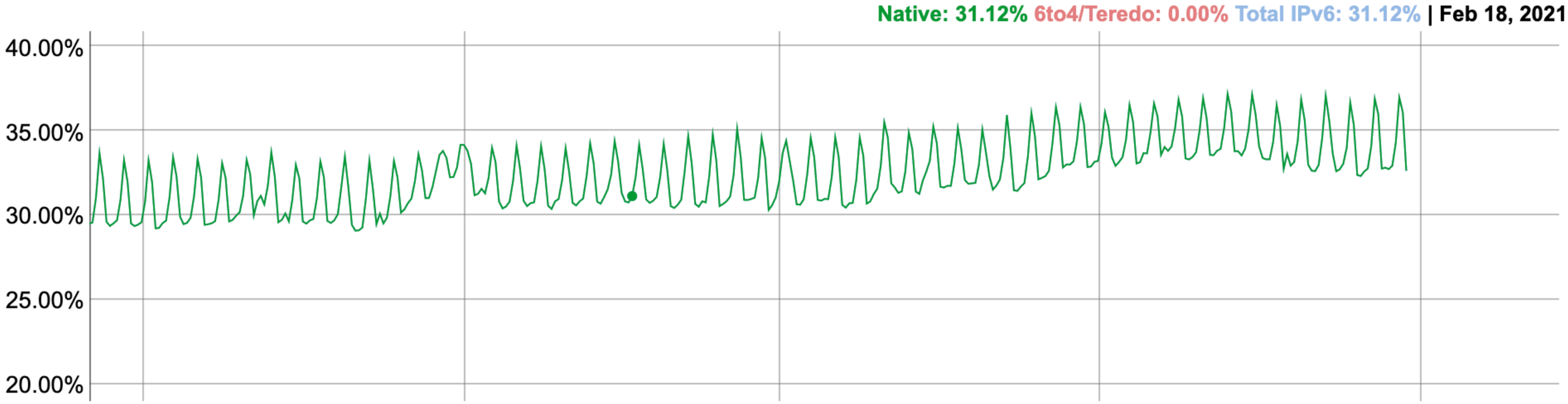
Google Observed Users



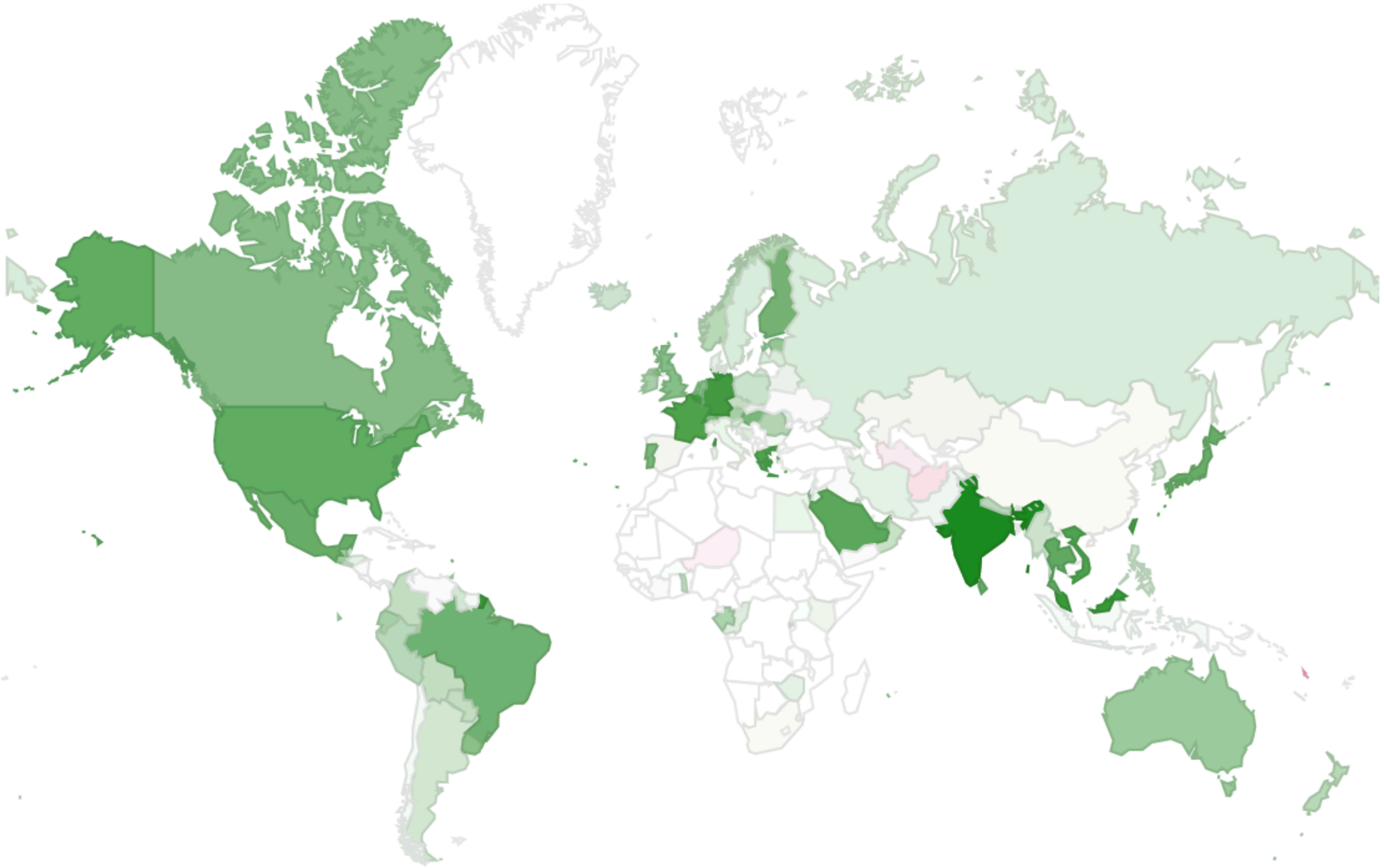
Higher Weekend Usage

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.



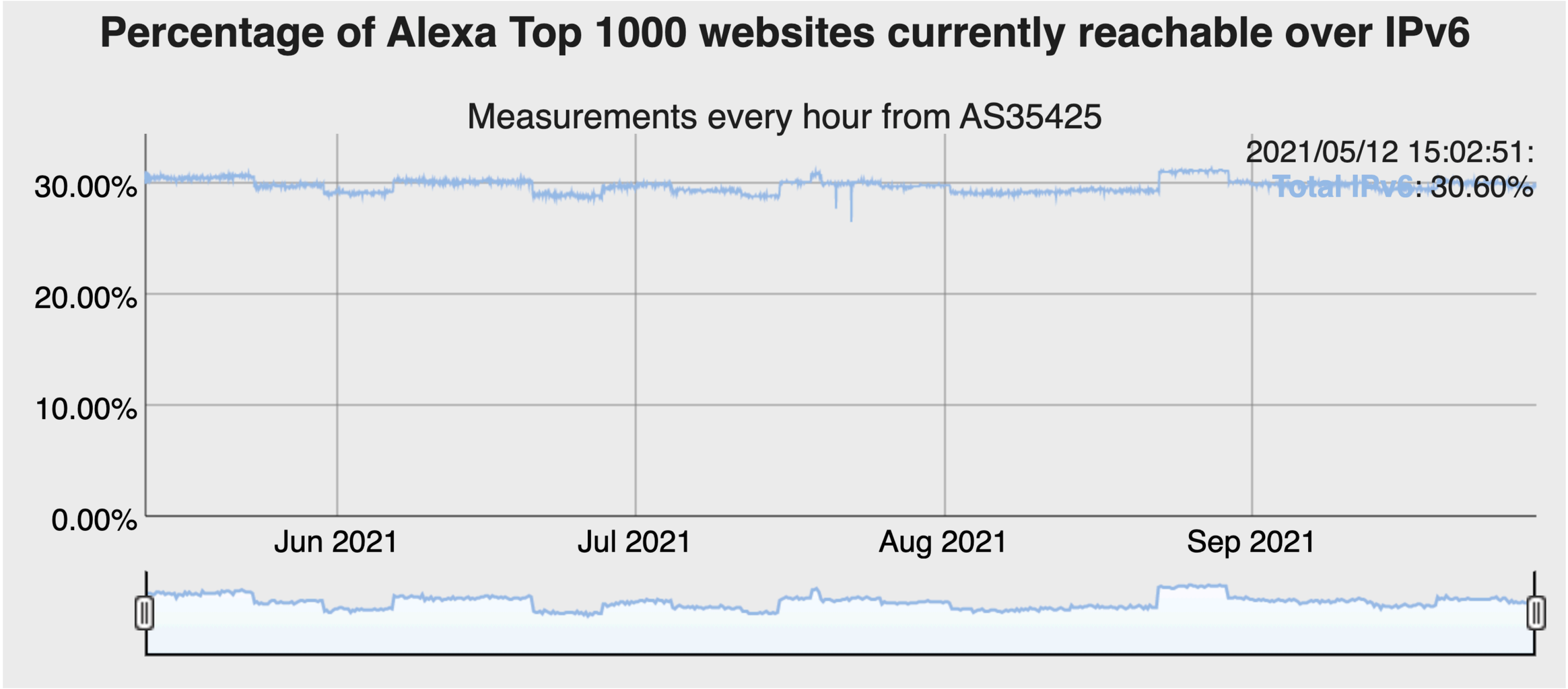
Geographic Biases



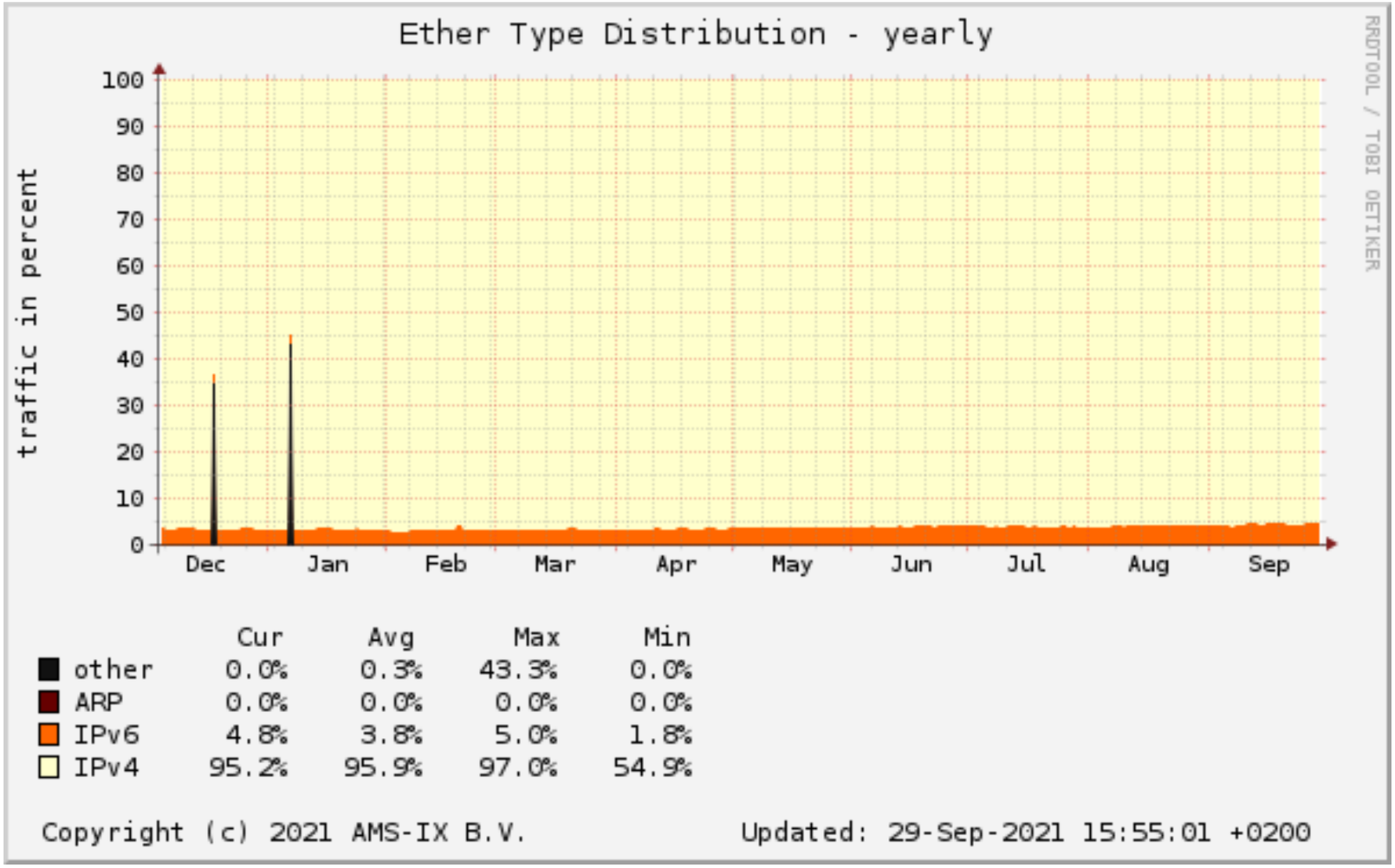
Alexa Top 1K Support

Alexa Rank	Website	AAAA Record	AAAA Record for www. Site	Site returns IPv6 source address	www. Site returns IPv6 source address
1	Google.com	✓	✓	✓	✓
2	YouTube.com	✓	✓	✓	✓
3	Facebook.com	✓	✓	✓	✓
4	Baidu.com	x	x	–	–
5	Wikipedia.org	✓	✓	✓*	✓*
6	Qq.com	x	✓	–	✓*
7	Tmall.com	x	x	–	–
8	Taobao.com	x	x	–	–
9	Yahoo.com	✓	✓	✓	✓
10	Amazon.com	x	x	–	–
11	Twitter.com	x	x	–	–
12	Sohu.com	x	x	–	–
13	Instagram.com	✓	✓	✓	✓
14	Reddit.com	x	x	–	–
15	Jd.com	x	x	–	–

Lots of Traffic != Lots of Deployment



AMS-IX Traffic Breakdown

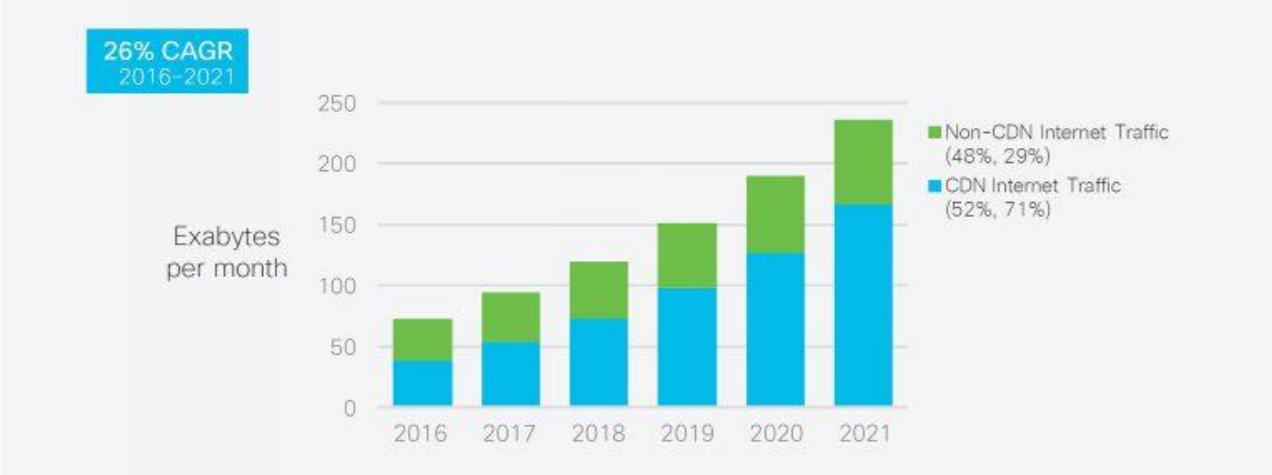


Content Delivery Strategies

CS249i

Why study content delivery strategies?

- Cisco estimates that 71% of Internet traffic traverses a content delivery network (CDN).
  - CDN's are how and *why* **The Modern Internet** works.
  - Content delivery is what Google, Amazon, Microsoft, Netflix, Cloudflare, Facebook, etc. all do.



Why study content delivery strategies?

- Cisco estimates that 71% of Internet traffic traverses a content delivery network (CDN).
  - CDN's are how and *why* **The Modern Internet** works.
  - Content delivery is at the core of Google, Amazon, Microsoft, Netflix, Cloudflare, Facebook, etc. businesses and they are constantly working to improve content delivery.

- Creating content delivery strategies is a very interesting systems/networking problem
  - Many papers at top-tier systems and networking conferences (NSDI, OSDI, IMC)

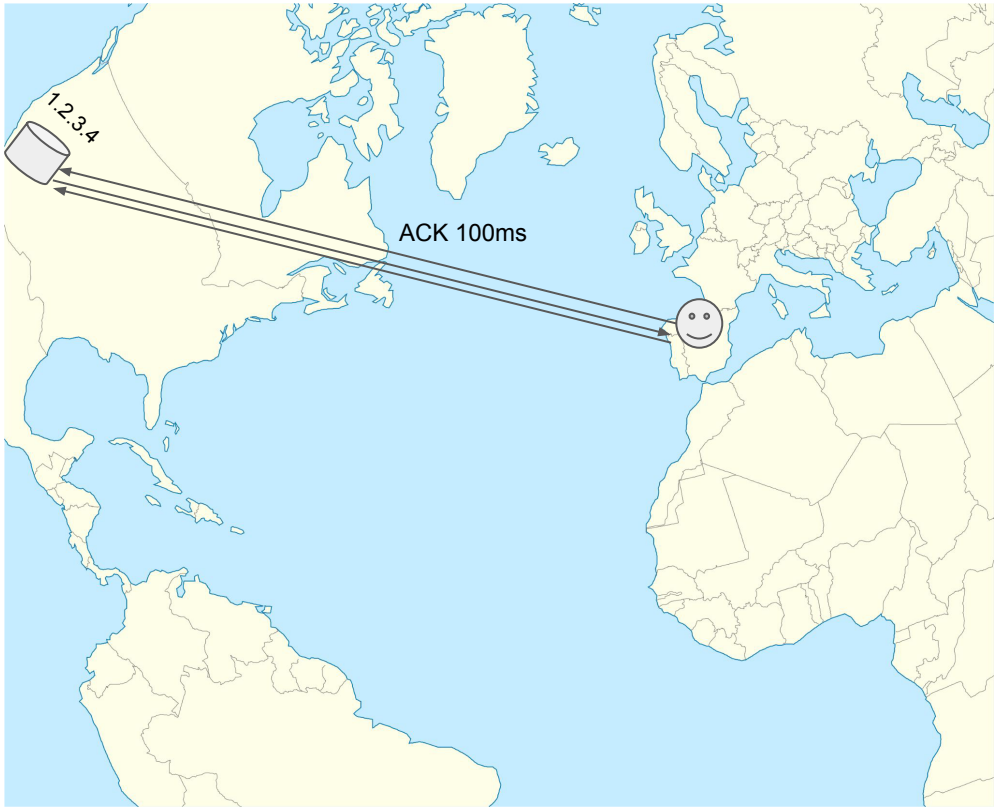
What does a world without a content delivery strategy look like?

Unicast

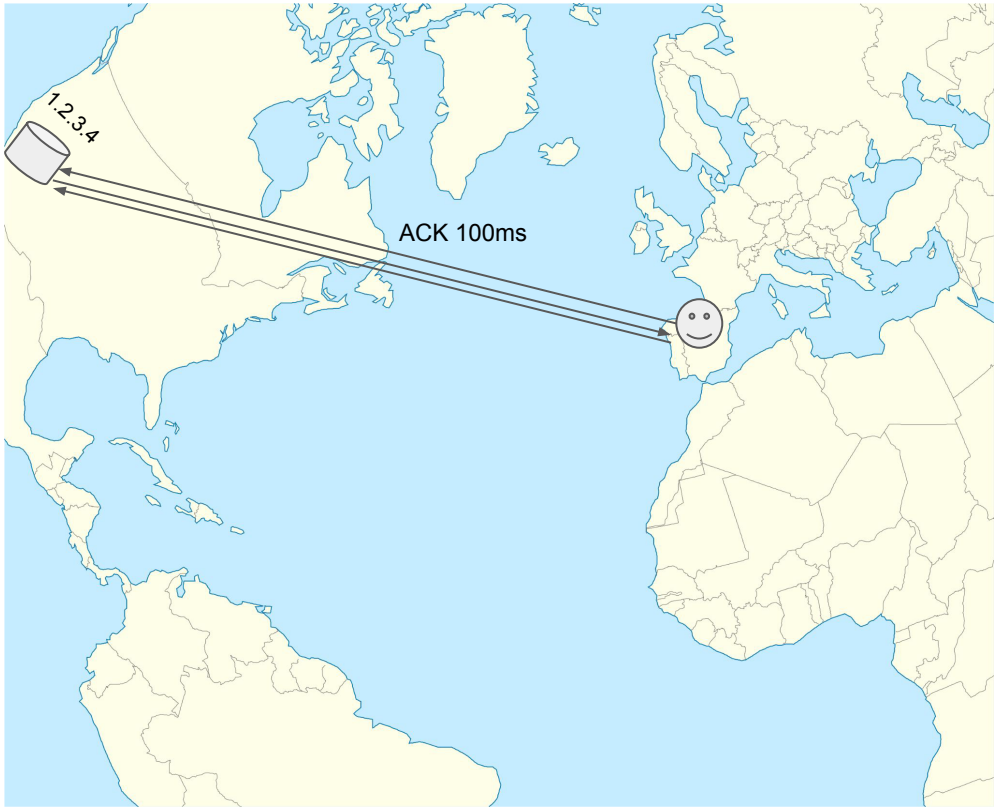
Unicast

Unicast

Unicast



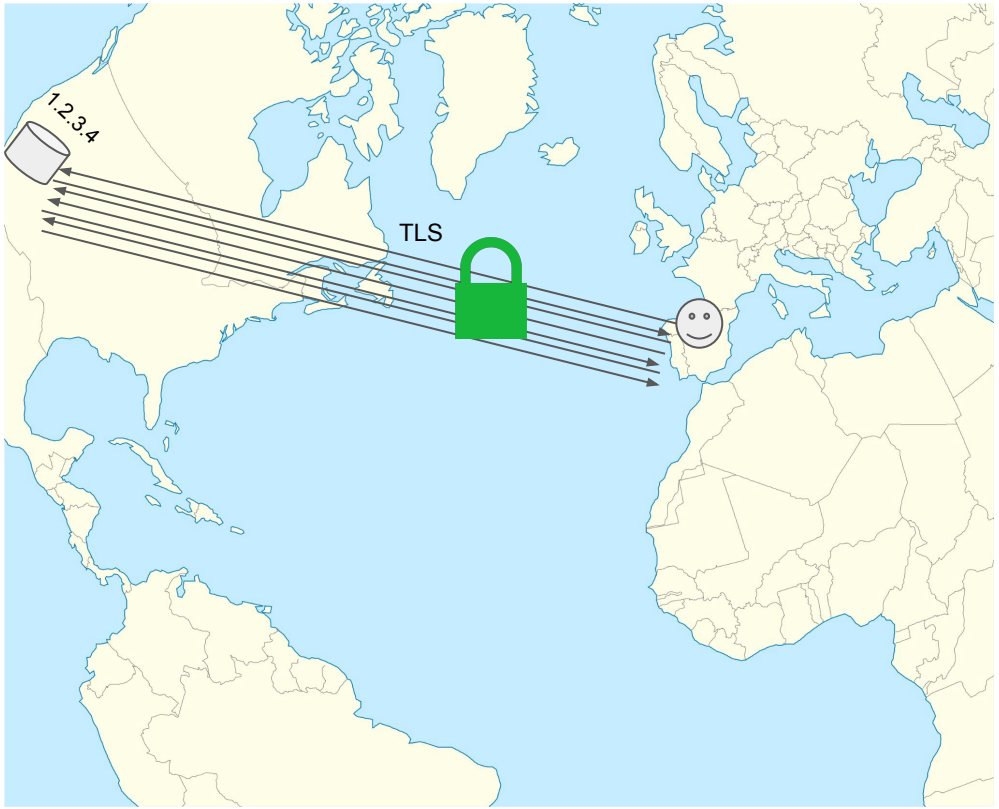
Unicast



TCP Handshake Time:

100ms *3 = 300ms

Unicast is too expensive



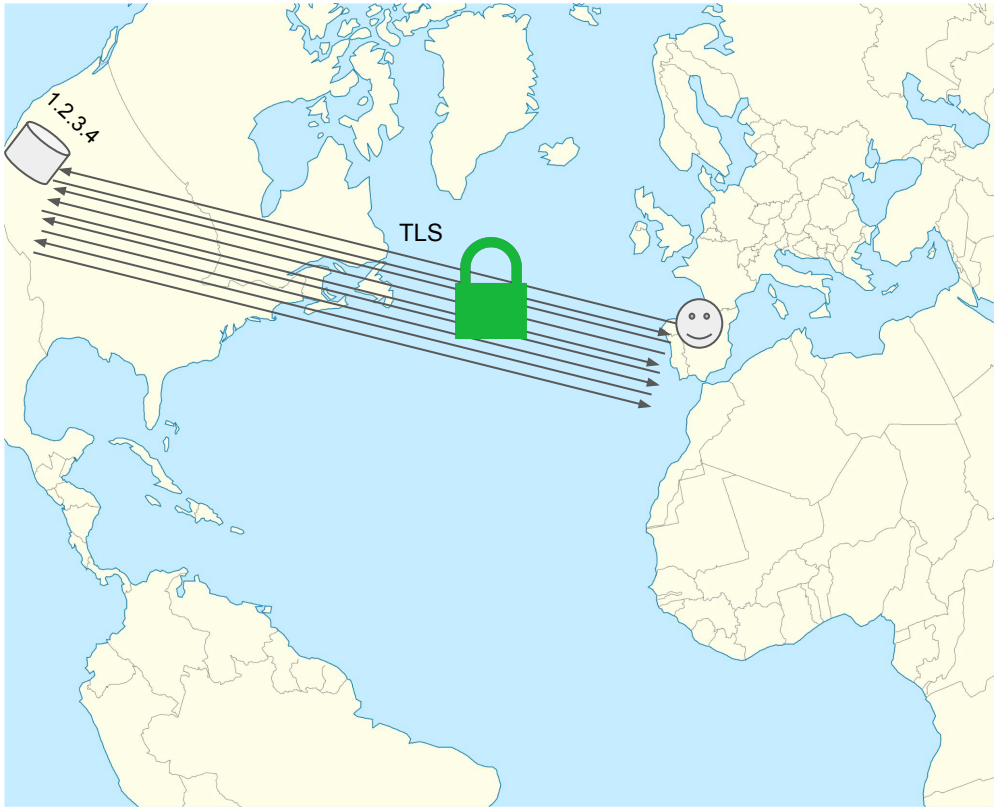
TCP Handshake Time:

100ms *3 = 300ms

TCP + TLS Handshake Time:

100ms *3 + 100ms *4 = 700 ms

Unicast is too expensive



TCP Handshake Time:

100ms *3 = 300ms

TCP + TLS Handshake Time:

100ms *3 + 100ms *4 = 700 ms

TCP + TLS + Content Time:

100ms *3 + 100ms *4 + 100ms * 2 = 900 ms

Speed Matters for ~~Google Web Search~~

Jake Brutlag
Google, Inc.
June 22, 2009

400ms ↑ load time ↓ 0.74% in searches

Internal goal of < 1 second response time

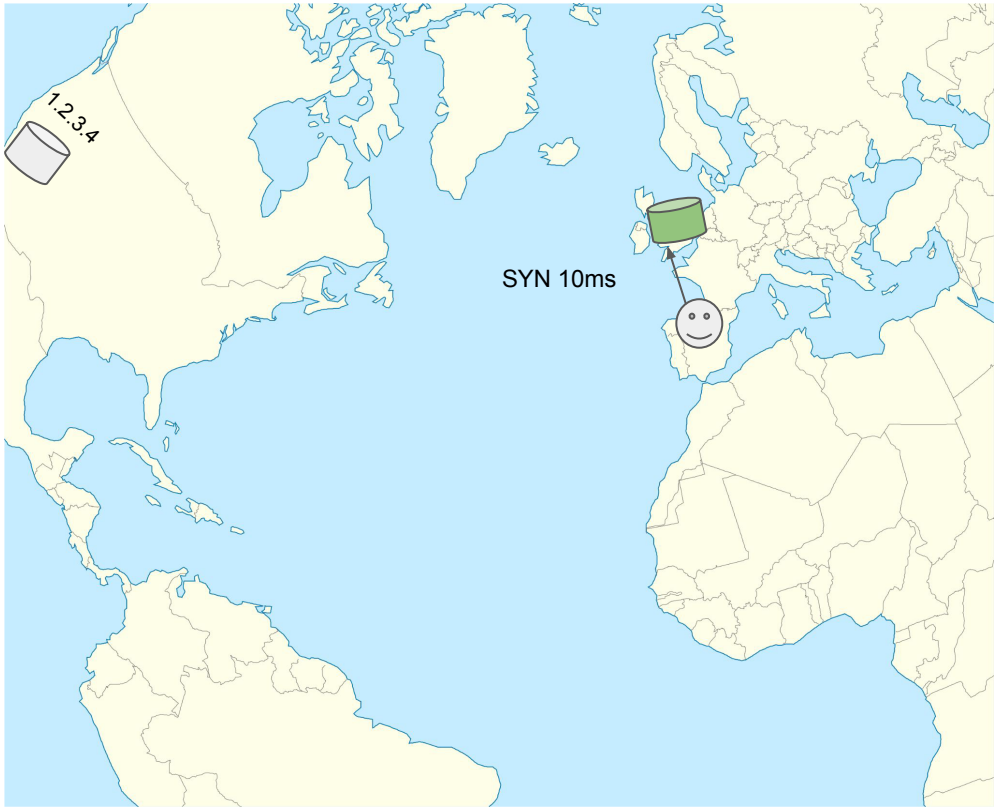
A world with a content delivery strategy

Taking advantage of edges

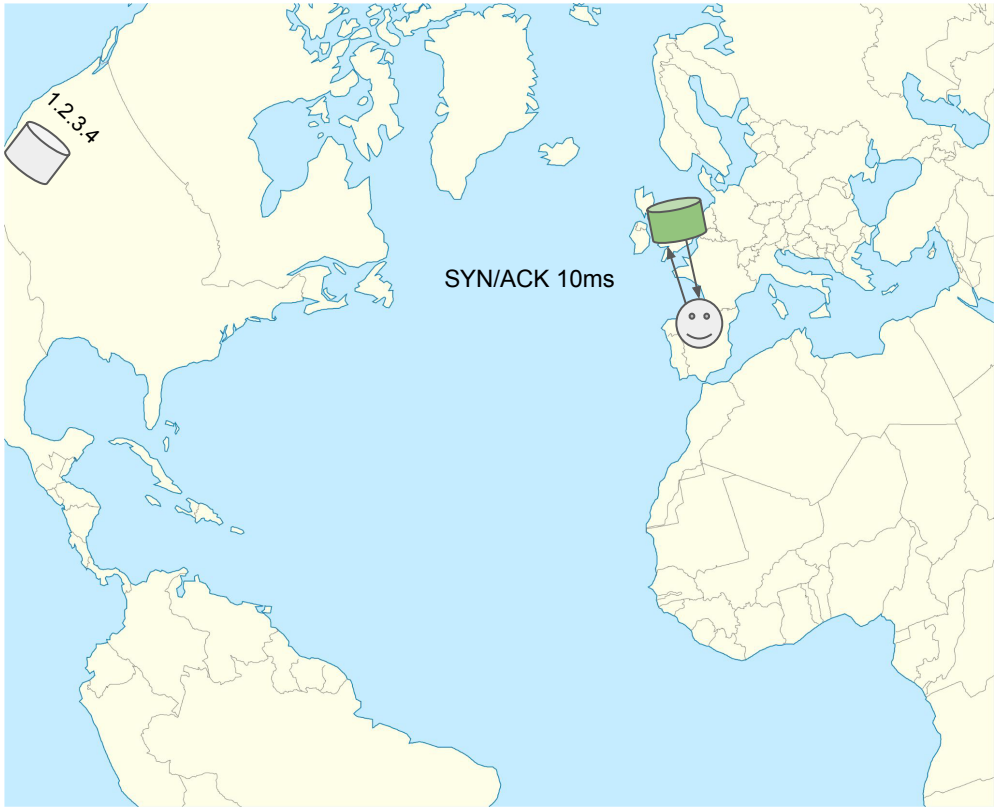
- Edge/ PoP (“Point of Presence”): server(s) located relatively near the client in order to help the client obtain the requested content

Taking advantage of edges

Taking advantage of edges



Taking advantage of edges



Taking advantage of edges

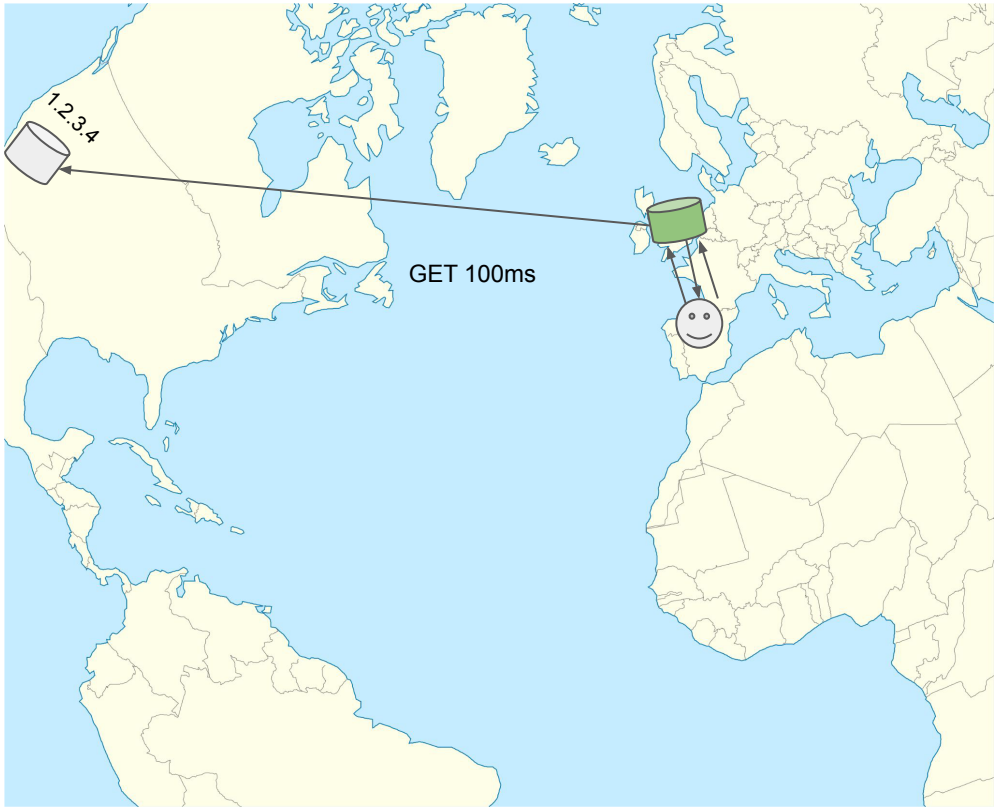
TCP Handshake Time:

10ms *3 = 30ms

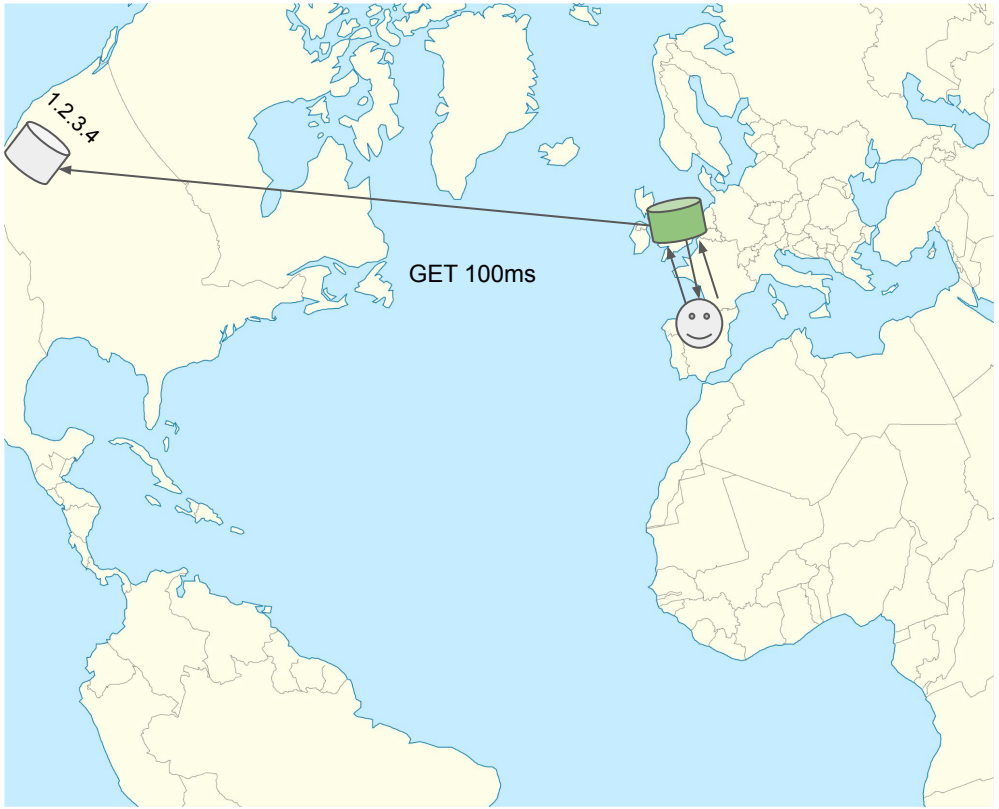
TCP + TLS Handshake Time:

10ms *3 + 10ms *4 = 70 ms

Taking advantage of edges

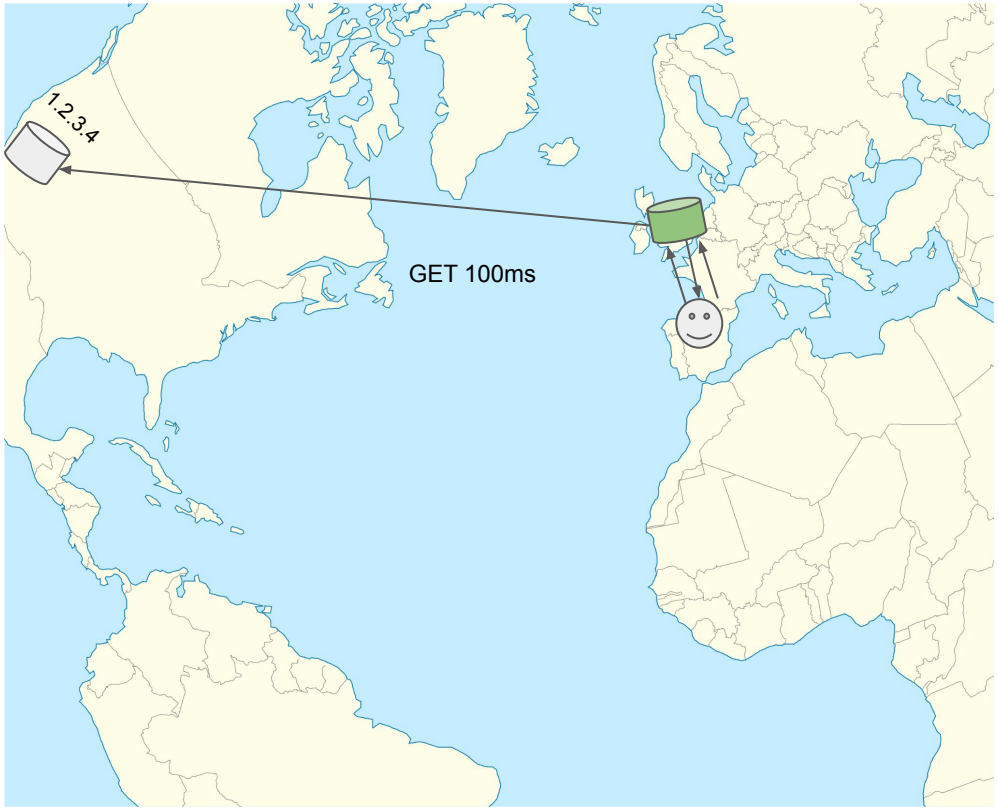


Taking advantage of edges



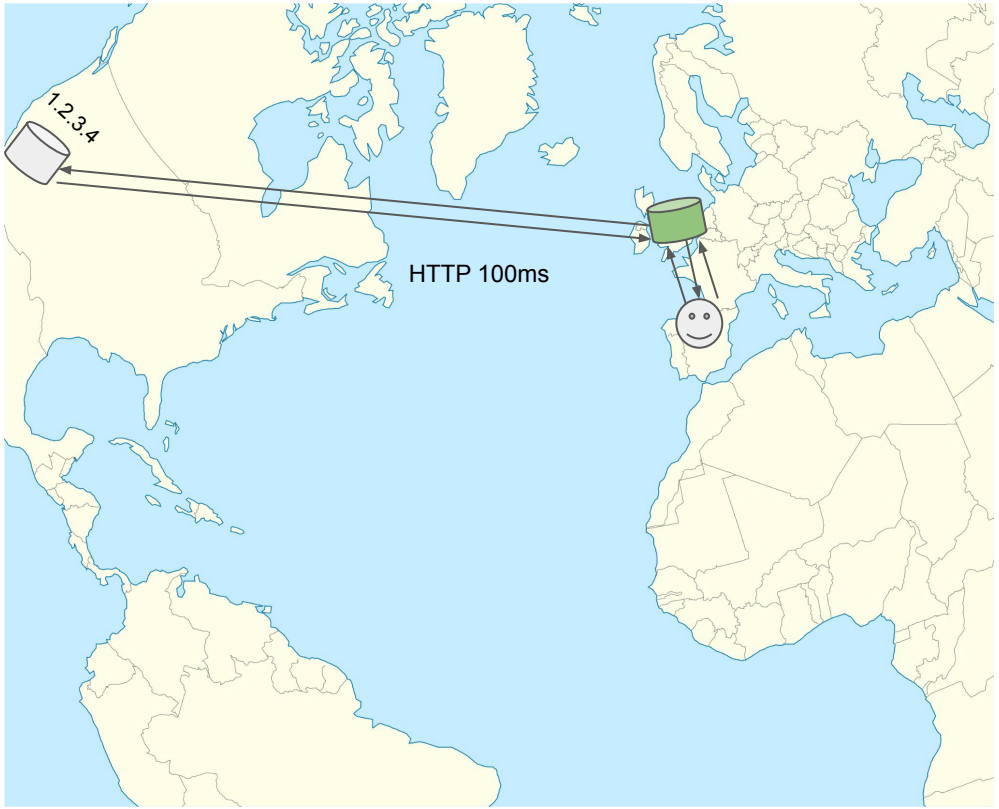
Why does the client not need to wait for the edge to establish a TCP/TLS connection?

Taking advantage of edges

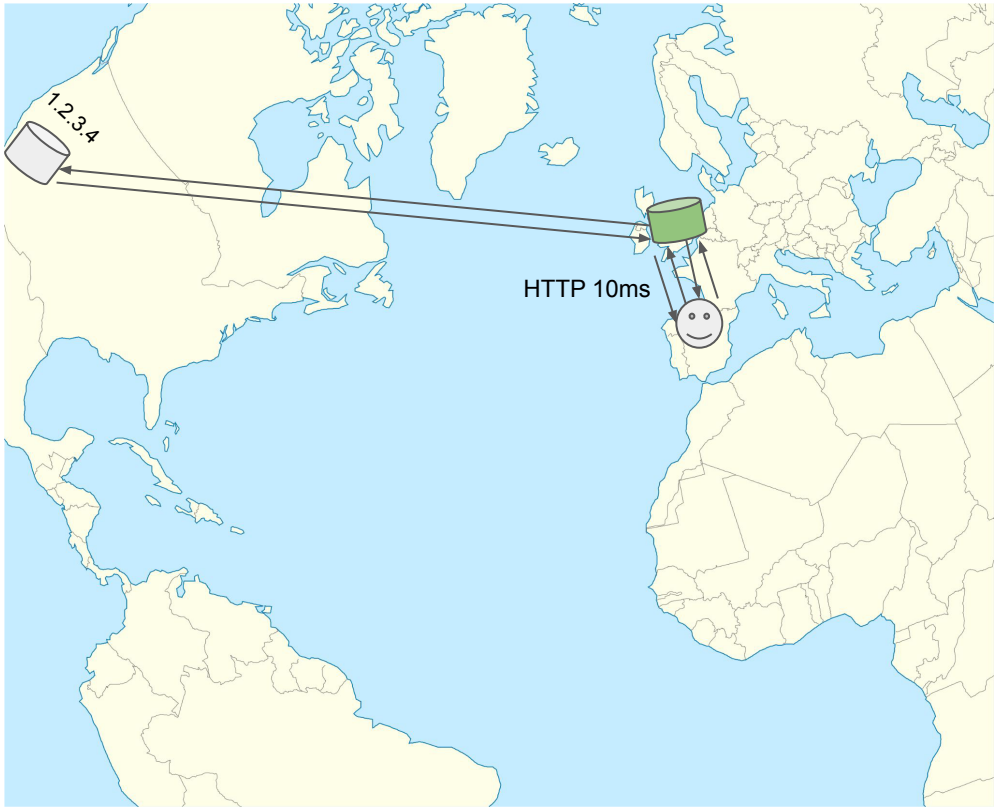


Edge can immediately send the GET request because it will have established a TCP connection before-hand and kept it alive.

Taking advantage of edges



Taking advantage of edges



TCP Handshake Time:

10ms *3 = 30ms

TCP + TLS Handshake Time:

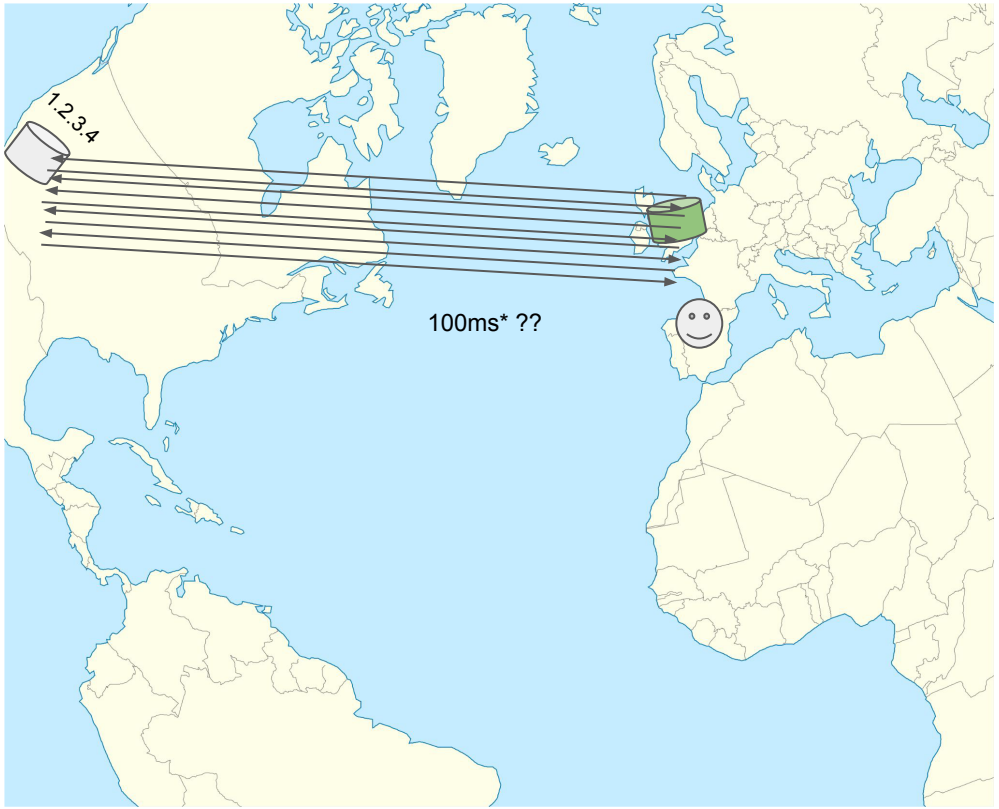
10ms *3 + 10ms *4 = 70 ms

TCP + TLS + Content Time:

70ms + 2*100ms + 10ms = 280ms

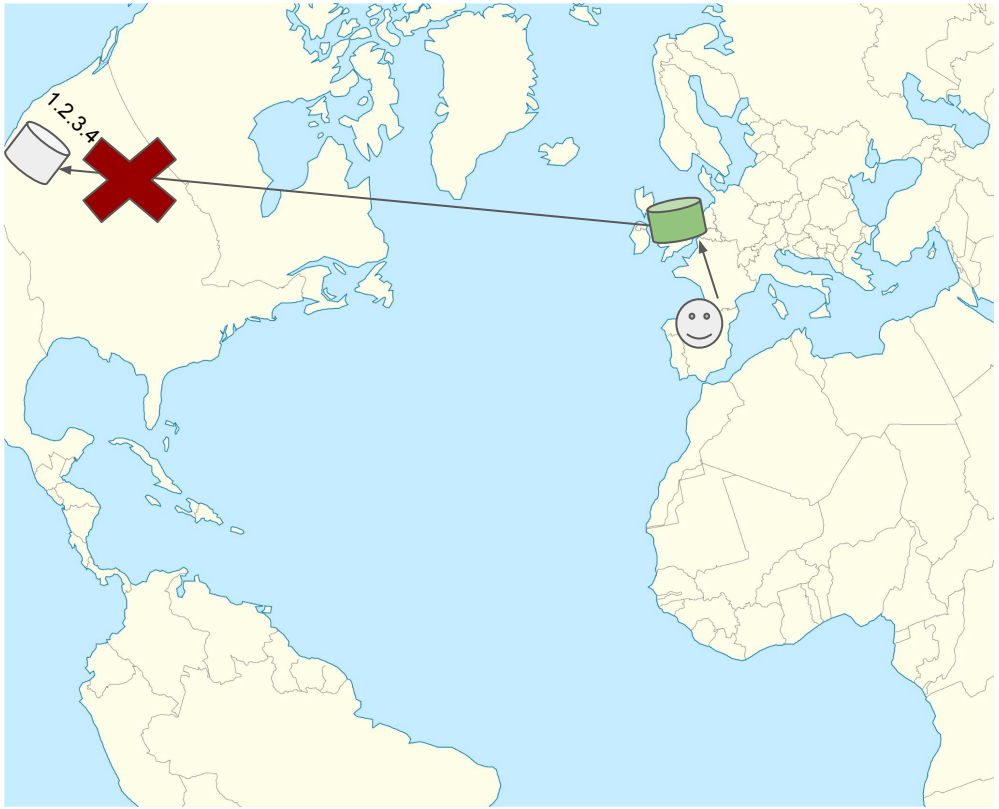
>3x savings!

Even with a connection-proxy edge, content server is far away



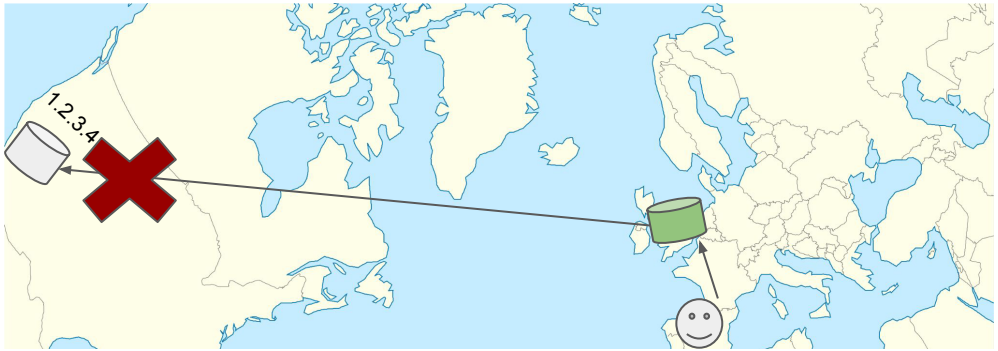
Single content server is vulnerable

- Single point of service failure



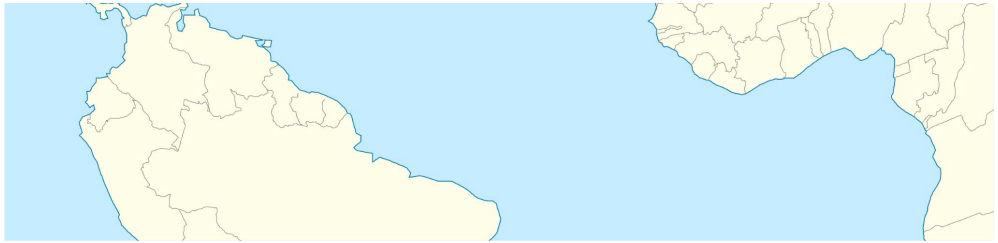
Single content server is vulnerable

- Single point of service failure

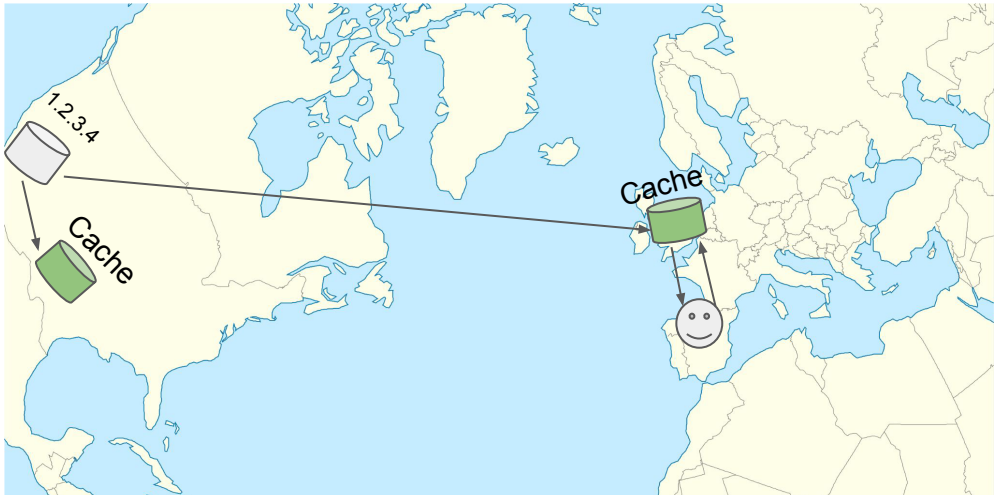


Ideally:

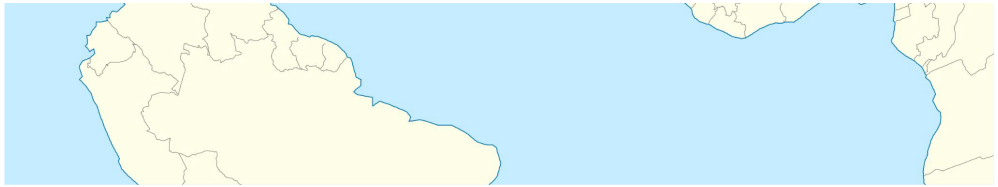
- Move content closer
- Have many content servers
- Have an edge close to the client



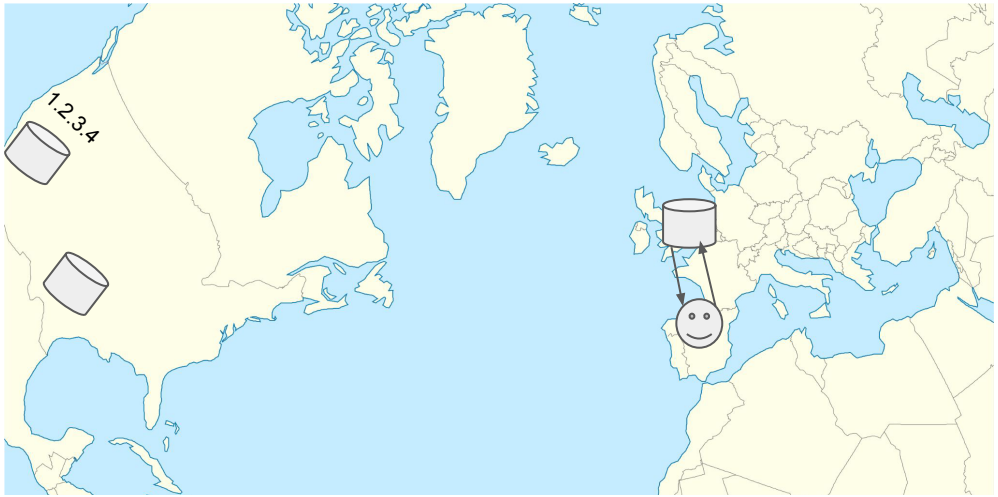
Options for moving content closer to user



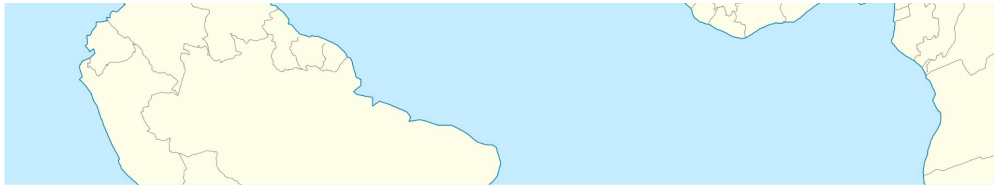
- Cache popular static content in the edge



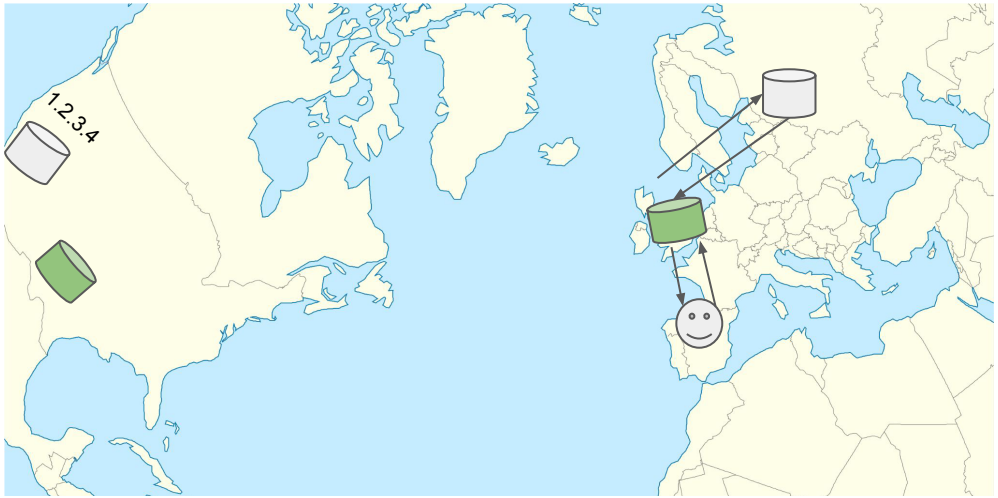
Options for moving content closer to user



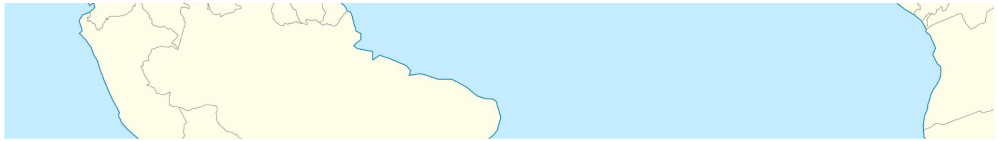
- Cache popular static content in the edge
- Make edge a replicated content server (better for dynamic content)



Options for moving content closer to user



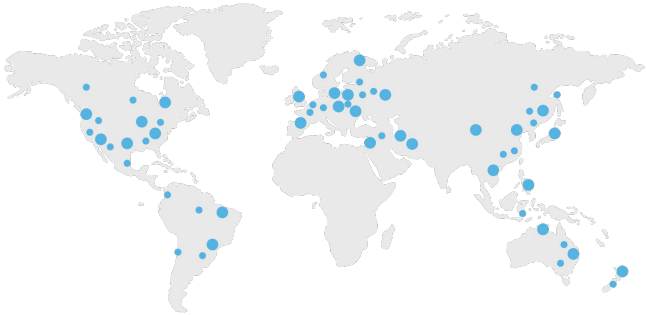
- Cache popular static content in the edge
- Make edge a replicated content server
- Some combination of both



Where should edge servers/ “PoPs” be located?

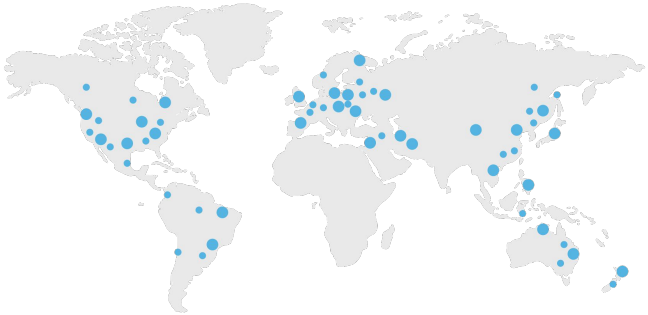
Where should edge servers/ “PoPs” be located?

- **“Scattered” Strategy**: Prioritize physical distance to client
  - Many low/medium capacity PoPs



Where should edge servers/ “PoPs” be located?

- **“Scattered” Strategy**: Prioritize physical distance to client
  - Many low/medium capacity PoPs



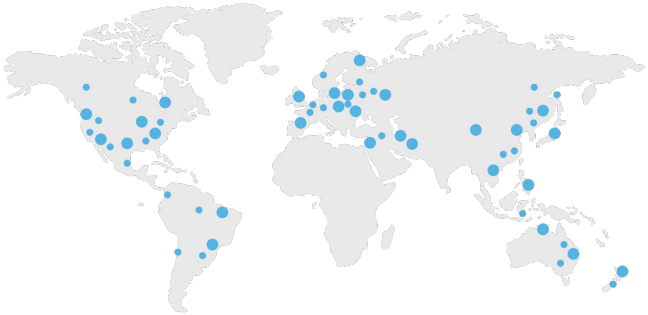
(+) Less distance to cover: less latency

(+) Effective in low-connectivity regions

(+) Easy to deploy

Where should edge servers/ “PoPs” be located?

- **“Scattered” Strategy**: Prioritize physical distance to client
  - Many low/medium capacity PoPs



(+) Less distance to cover: less latency

(+) Effective in low-connectivity regions

(+) Easy to deploy

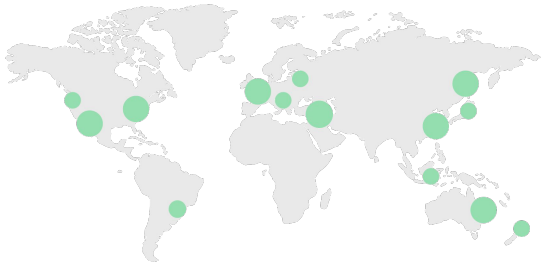
(-) Modern fiber cables makes distance less of a bottleneck

(-) Tough to maintain/update

“Copper-based transmissions currently max out at 40 Gbps, whereas fiber optics can carry data at close to the speed of light.”

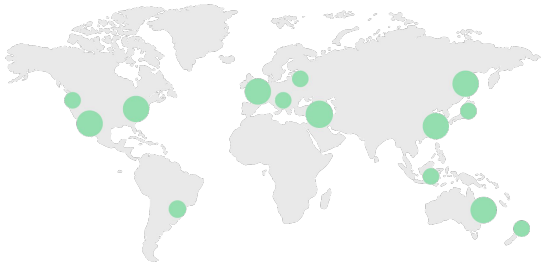
Where should edge servers/ “PoPs” be located?

“Consolidated” Strategy: Prioritize fewer, but more powerful PoPs (data centers, IXPs)



Where should edge servers/ “PoPs” be located?

“Consolidated” Strategy: Prioritize fewer, but more powerful PoPs (data centers, IXPs)



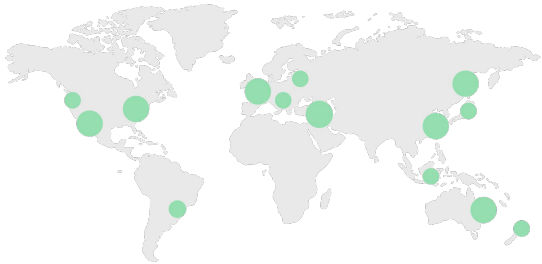
(+) Serve/cache more content

(+) Better connected to the next hop (e.g., IXP)

(+) Provides DDoS mitigation

(+) Easier to maintain/update

Where should edge servers/ “PoPs” be located?



“Consolidated” Strategy: Prioritize fewer, but more powerful PoPs (data centers, IXPs)

(+) Serve/cache more content

(+) Better connected to the next hop (e.g., IXP)

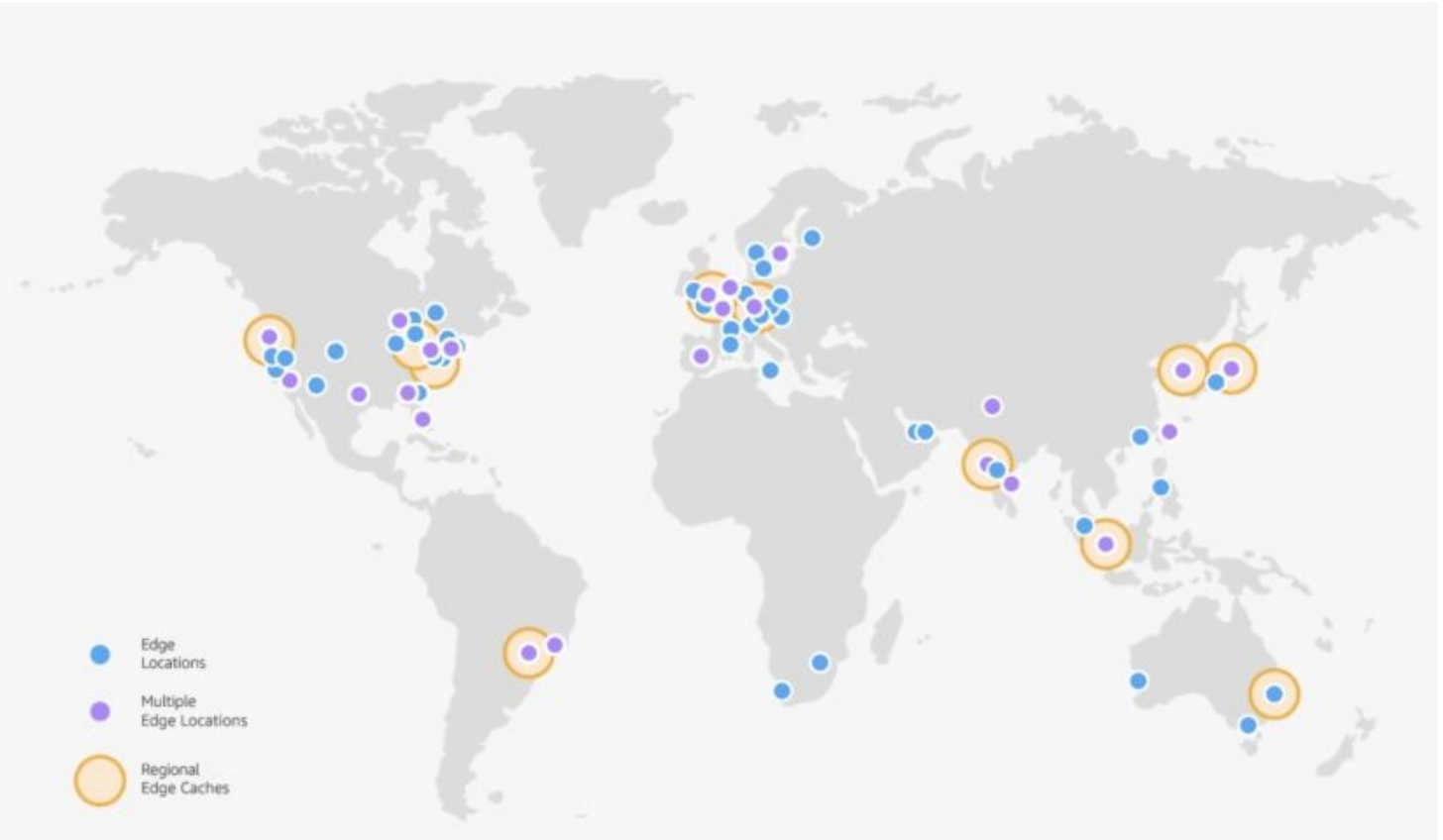
(+) Provides DDoS mitigation

(+) Easier to maintain/update

(-) Tough to deploy new PoP

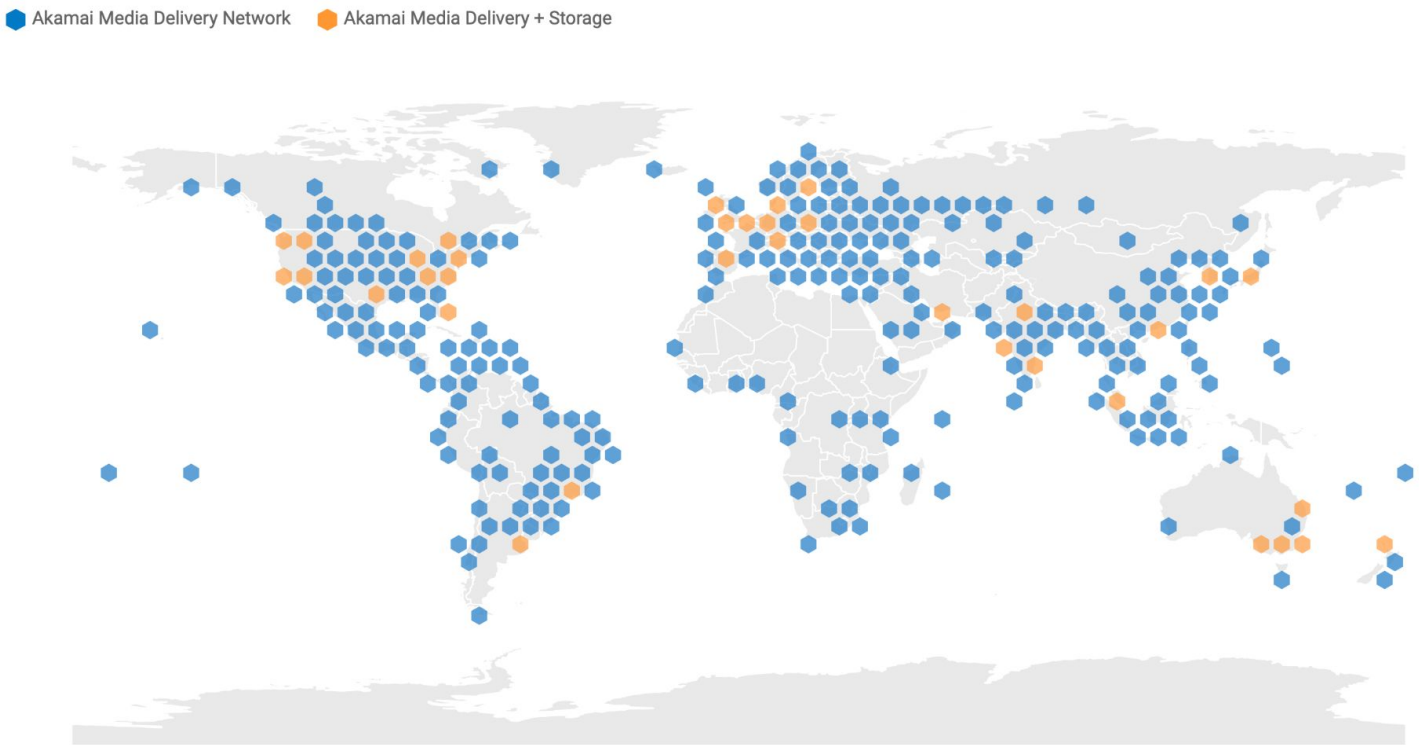
(-) Less effective in low-connectivity regions

amazon cloudfront



https://aws.amazon.com/blogs/aws/200-amazon-cloudfront-points-of-presence-price-reduction/

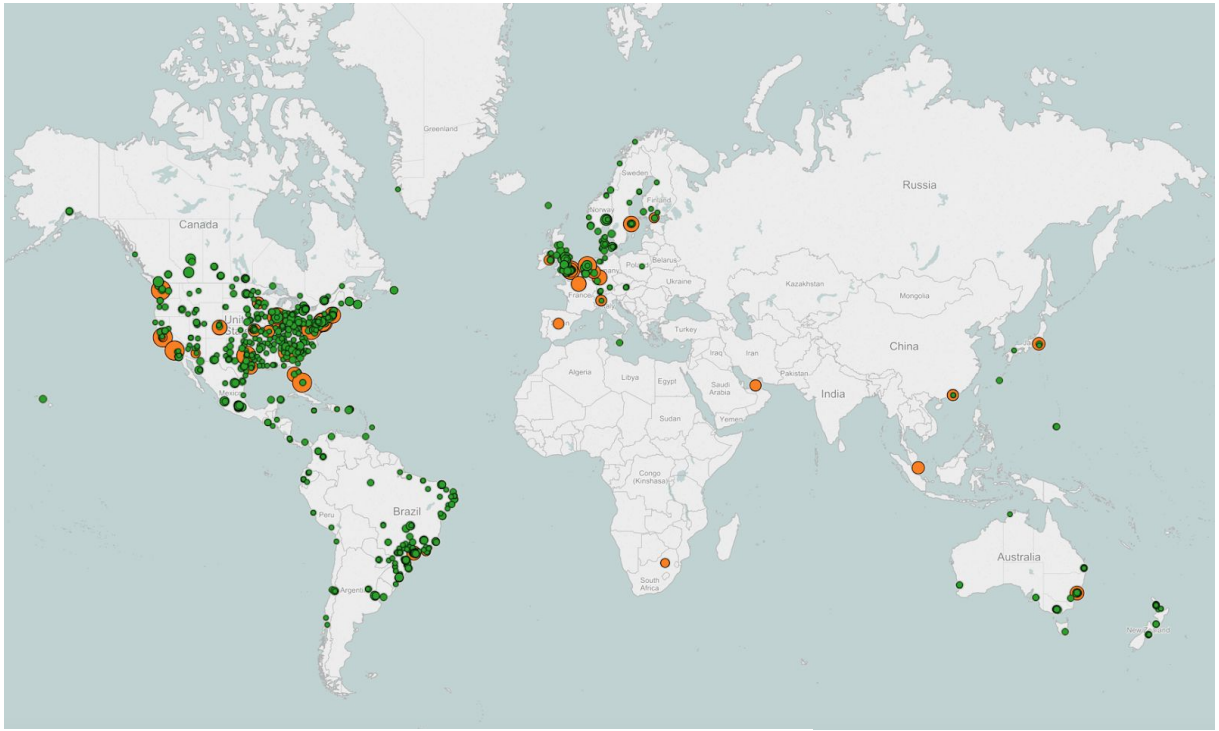
Akamai



https://www.akamai.com/visualizations/media-delivery-network-map

NETFLIX

Open Connect CDN

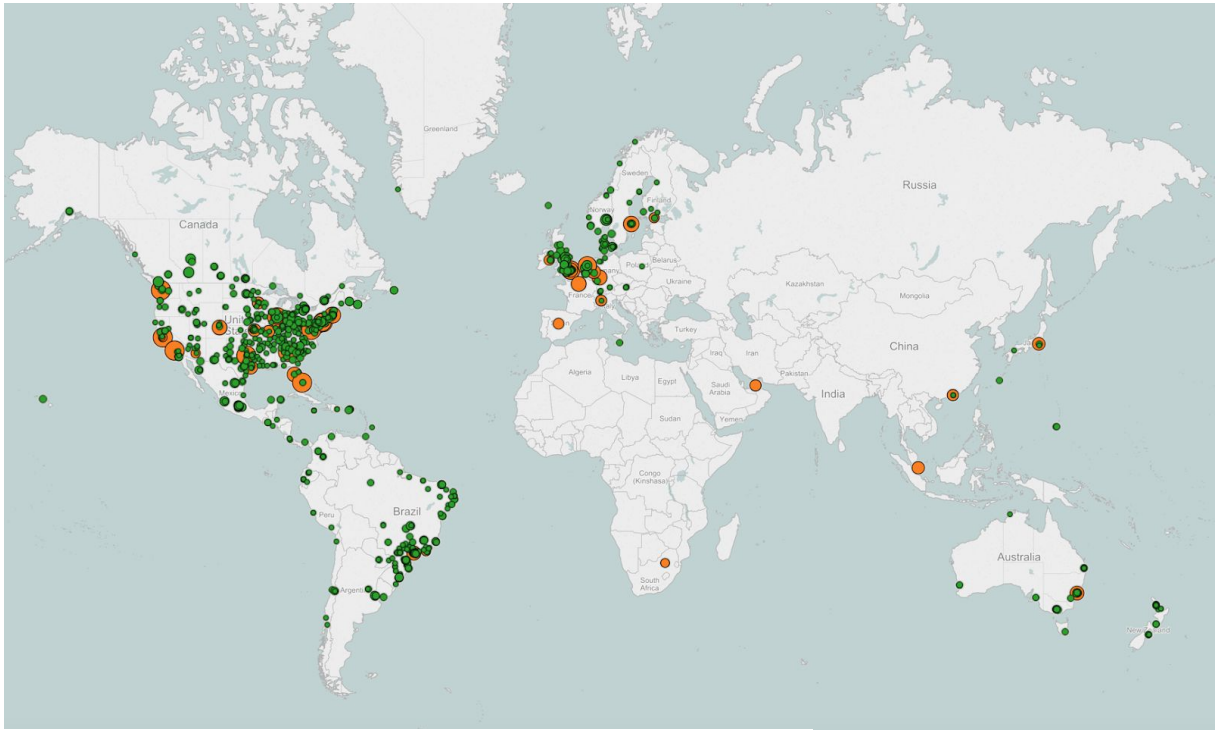


ISP Locations Internet Exchange Point (circles are sized by volume)

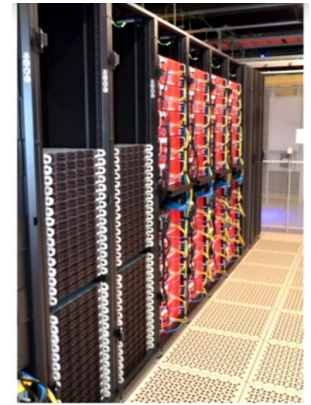
https://about.netflix.com/en/news/how-netflix-works-with-isps-around-the-globe-to-deliver-a-great-viewing-experience

NETFLIX

Open Connect CDN



ISP Locations Internet Exchange Point (circles are sized by volume)



- Netflix does not run its own network! It convinces ISPs to put its own “Open Connect Appliances” in their data centers

https://about.netflix.com/en/news/how-netflix-works-with-isps-around-the-globe-to-deliver-a-great-viewing-experience

Why do ISPs participate in Netflix's Open Connect?

- Netflix promises to pre-position content during off-peak hours, in order to reduce burden on the Internet during peak hours. Thus, ISPs do not have to worry about building more network capacity.

A brief history of Netflix's infrastructure woes

1998: Netflix is born

2007: Netflix builds two datacenters. Netflix builds its own CDNs using 5 locations within the US

- Painful process: ordering equipment, installing, never large enough..always need more

2008: Netflix goes offline for three days due to their own infrastructure.

2008: Netflix moves to AWS (North Virginia, Portland Oregon, Dublin Ireland.) and has previously said they have no intention to operate out of more regions

2009: Netflix abandons building their own CDN, turn to Akamai, Limelight, Level 3

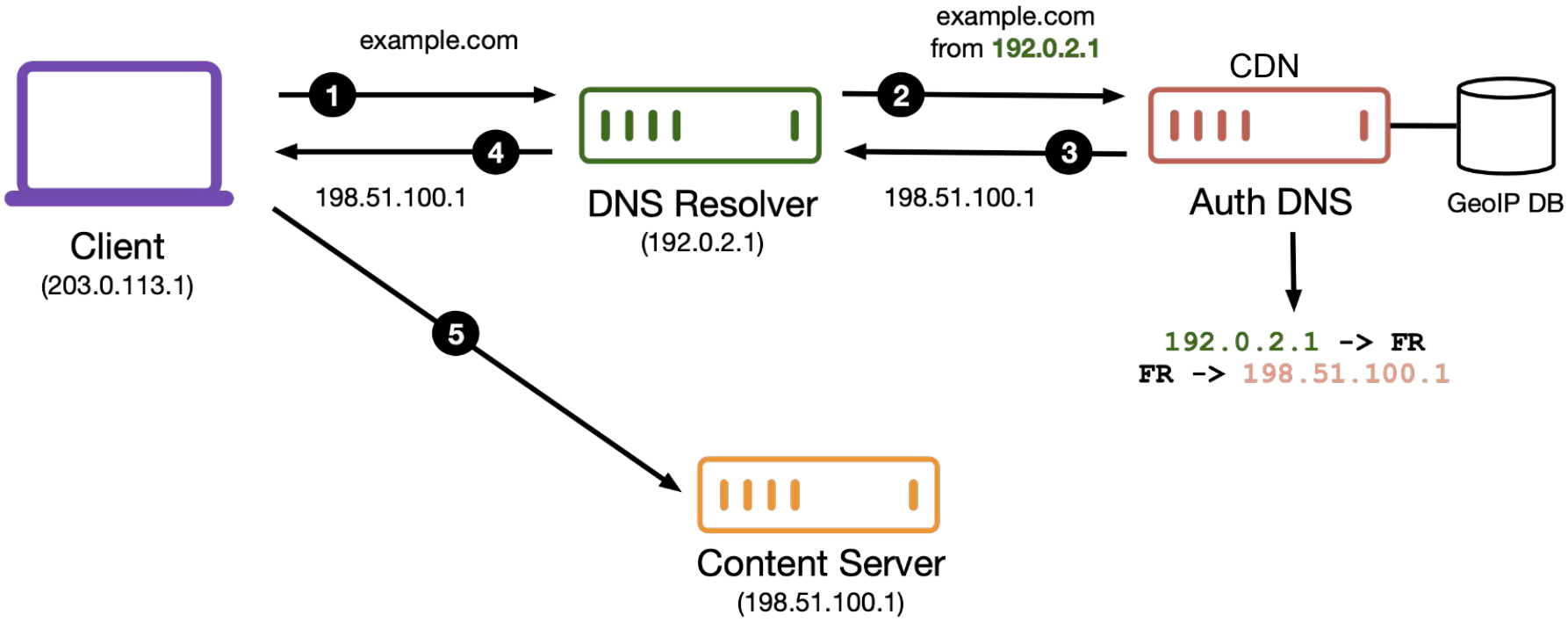
2011: Netflix decides they need a dedicated CDN to maximize network efficiency

2012: Netflix launches Open Connect (less expensive, better control, more scalable)

How does one choose the nearest PoP?

(1) (DNS routing, “Map the Internet”, Unicast) Approach

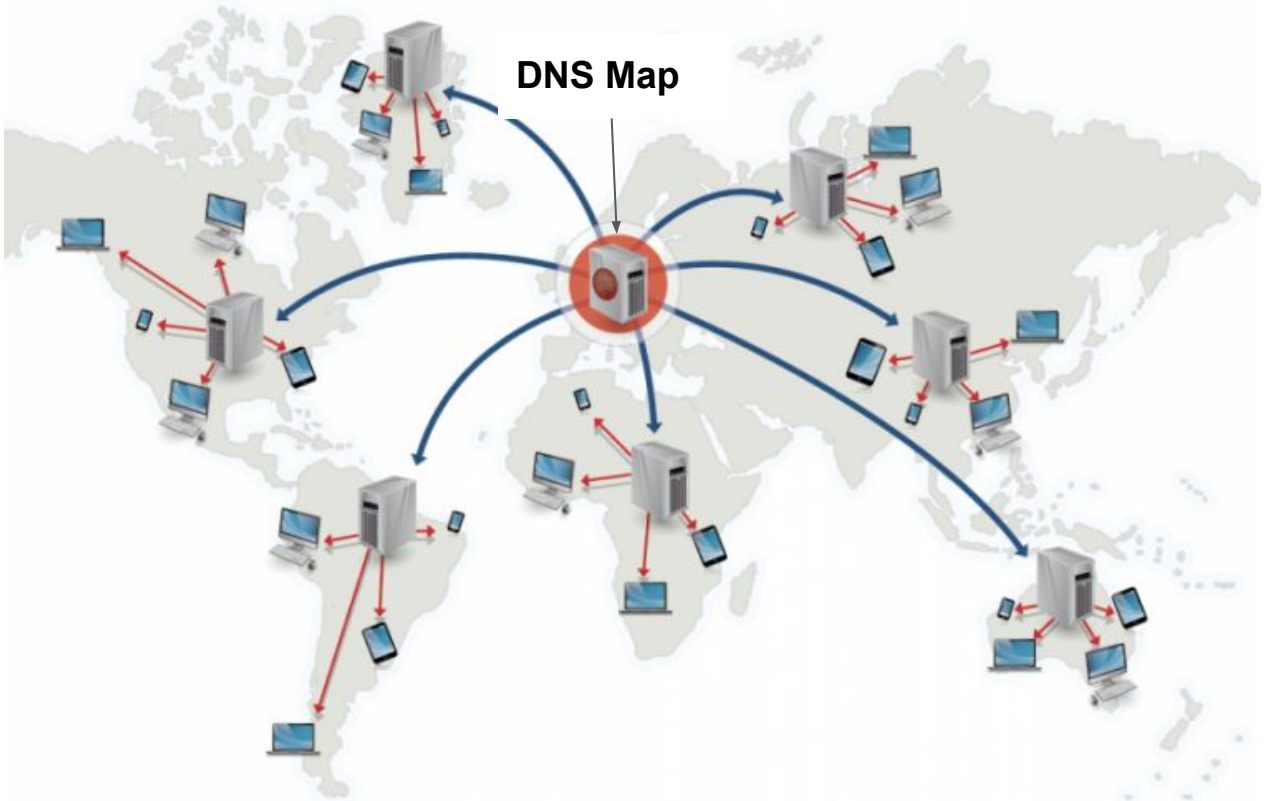
DNS uses the client resolver IP to return the edge/PoP IP that is closest to the client



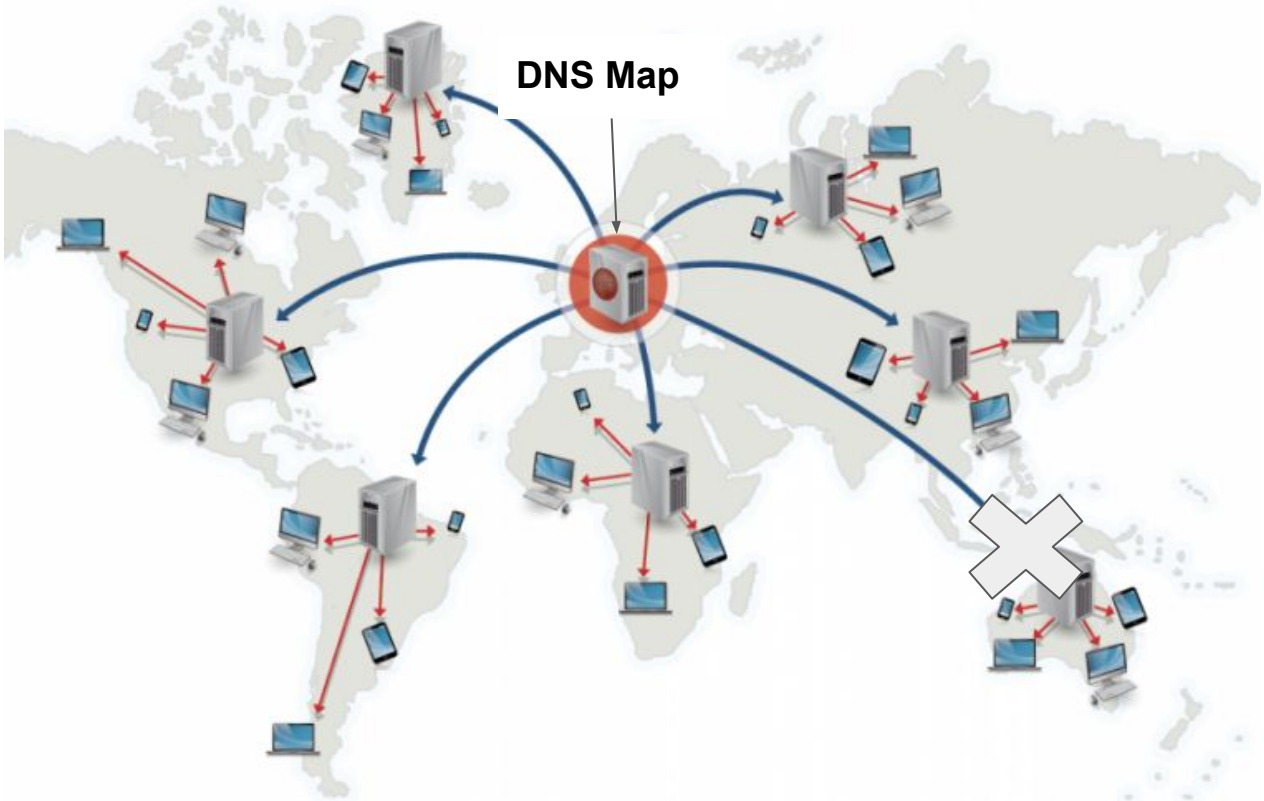
(1) (DNS routing, “Map the Internet”, Unicast) Approach

Every edge/PoP is assigned its own unique IP address

(1) (DNS routing, “Map the Internet”, Unicast) Approach



Upon failure of an edge/PoP, DNS must detect and re-route



(1) (DNS routing, “Map the Internet”, Unicast) Approach

(+) direct control of which edge is chosen
(+) “real-time”* re-routing

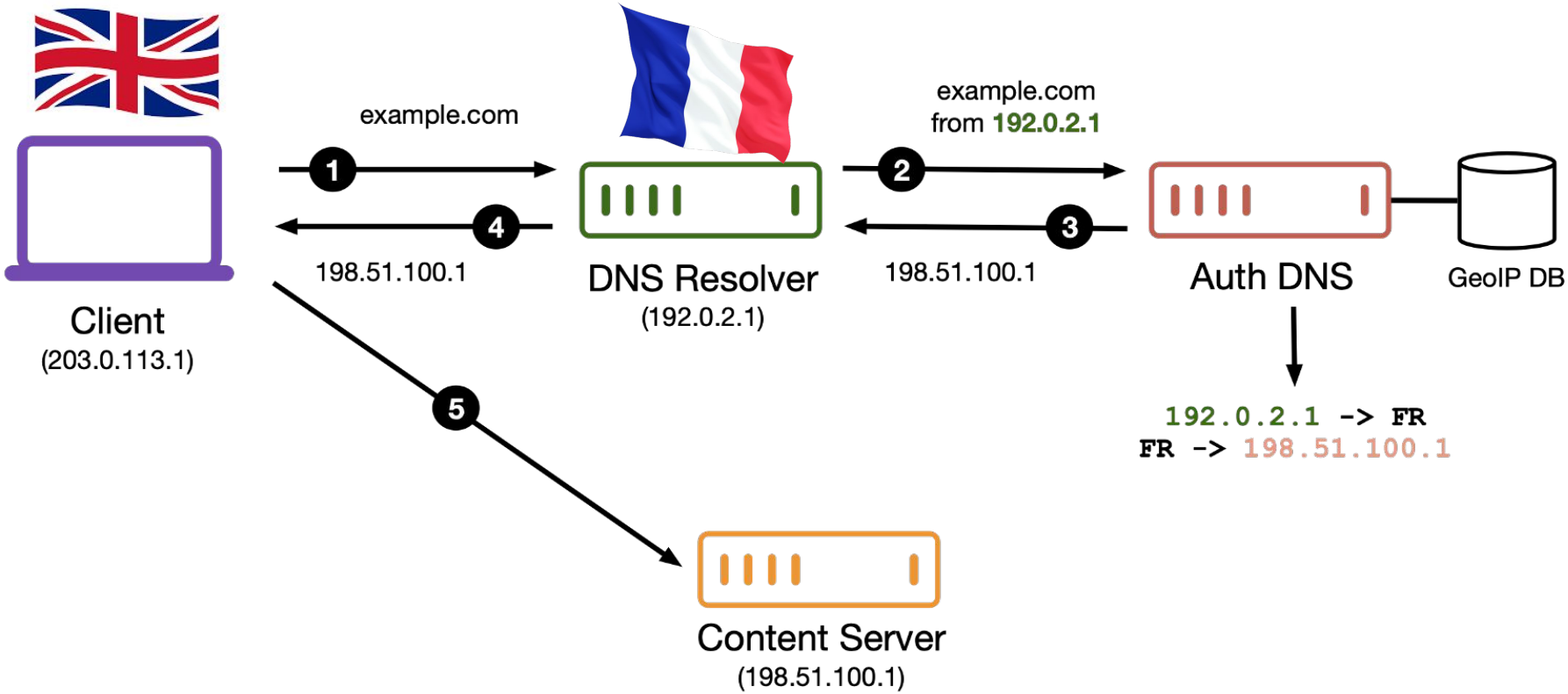
(1) (DNS routing, “Map the Internet”, Unicast) Approach

(+) direct control of which edge is chosen
(+) “real-time”* re-routing

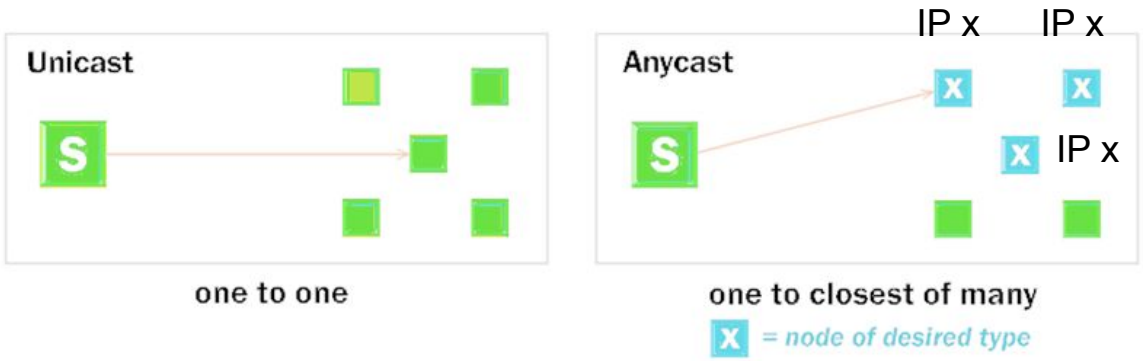
(-) Extra infrastructure/operations required
- “health” monitoring of edges/PoPs

(-) Availability requires short TTLs (increases the amount of DNS lookups..)
(-) DNS doesn’t always know where the client actually is (resolver location != client location) e.g., Google Public DNS and OpenDNS

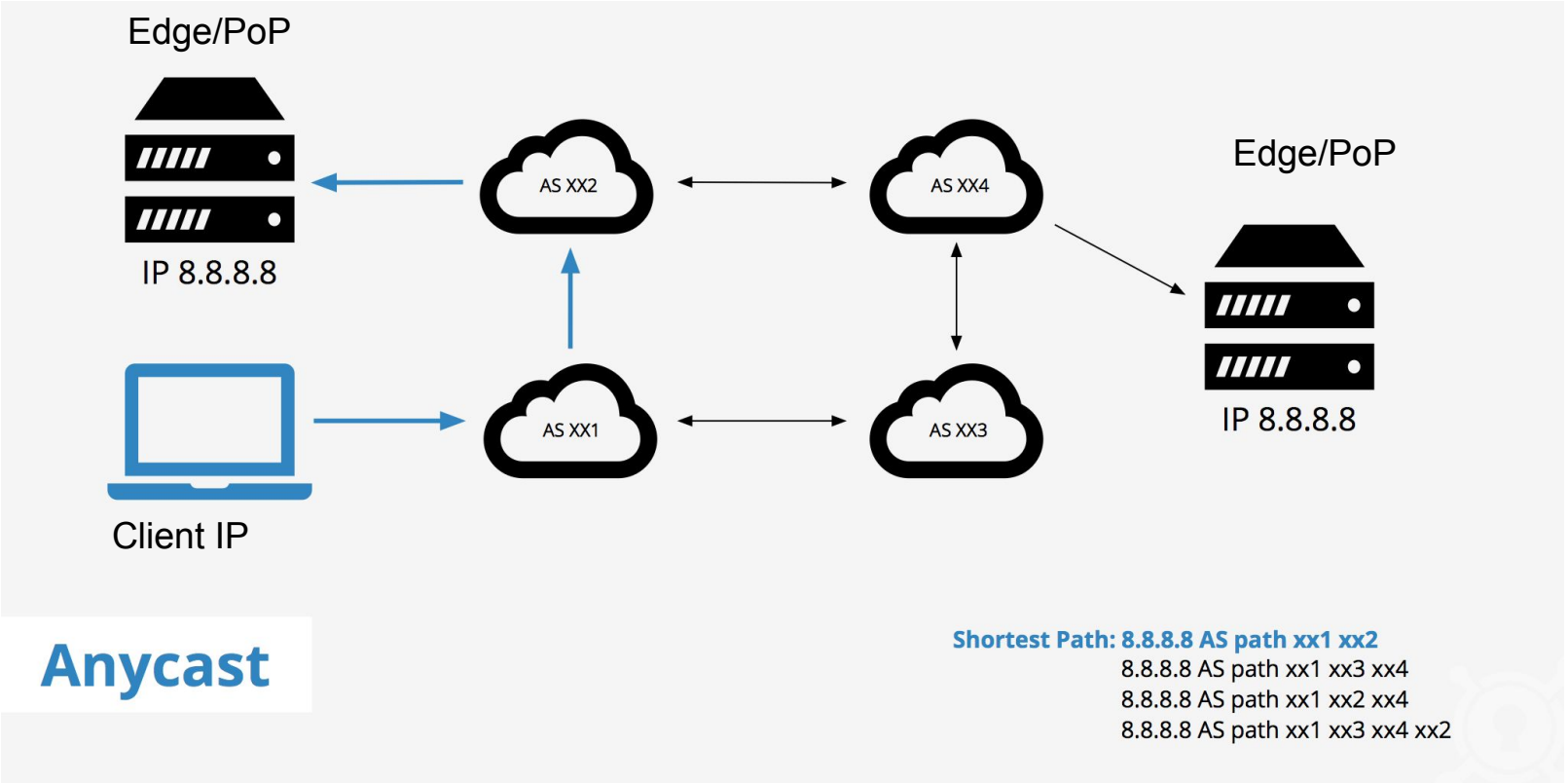
DNS doesn't always know where the client actually is



(2) Anycast routing approach



Anycast routing for CDNs



Anycast routing for CDNs

(+) Clients choose the edge location; CDN does not need to guess

(+) Naturally reactive to failures, thanks to BGP

Anycast routing for CDNs

(+) Clients choose the edge location; CDN does not need to guess

(+) Naturally reactive to failures, thanks to BGP

> **Global Thermonuclear War:** would be bad, but CloudFlare may continue to be able to route traffic to whatever portion of the Internet is left. As facilities

https://blog.cloudflare.com/cloudflares-architecture-eliminating-single-p/

Anycast routing for CDNs

(+) Clients choose the edge location; CDN does not need to guess

(+) Naturally reactive to failures, thanks to BGP

(-) Manipulating traffic can be slow: rely on BGP propagation

(-) BGP route flaps: TCP SYN and ACK can theoretically get diverted to different servers. Though route flap damping should take care of this.

Penalizes constant route changing
https://labs.ripe.net/author/clemens_mosig/route-flap-damping-in-the-wild/

Anycast routing for CDNs

(+) Clients choose the edge location; CDN does not need to guess

(+) Naturally reactive to failures, thanks to BGP

(-) Manipulating traffic can be slow: rely on BGP propagation

(-) BGP route flaps: TCP SYN and ACK can theoretically get diverted to different servers. Though route flap damping should take care of this.

(-) Have to predict which PoPs will likely receive the most traffic...predictions can change over time/ be wrong...but infrastructure is already there.

- Can create overload/ underload

FastRoute: A Scalable Load-Aware Anycast Routing Architecture for Modern CDNs

Ashley Flavel, Pradeepkumar Mani, David A. Maltz, and Nick Holt, *Microsoft;*
Jie Liu, *Microsoft Research;* Yingying Chen and Oleg Surmachev, *Microsoft*

https://www.usenix.org/conference/nsdi15/technical-sessions/presentation/flavel

FastRoute: How to handle anycast overload

System goal: To be able to re-route anycast traffic when an edge gets overloaded.

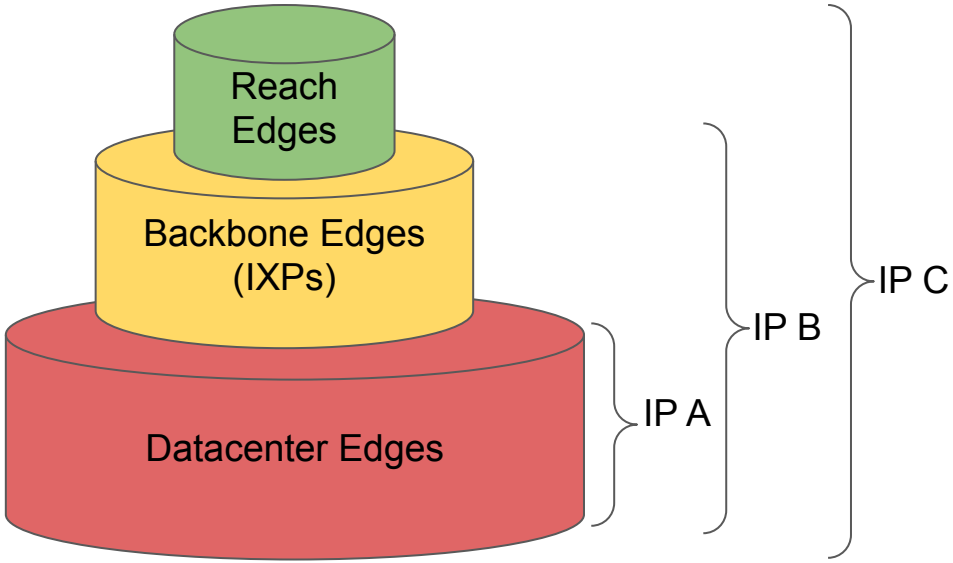
Solution design goal: Just as simple as anycast, but just enough control to re-route.

FastRoute: How to handle anycast overload

Microsoft

System goal: To be able to re-route anycast traffic when an edge gets overloaded.

Solution design goal: Just as simple as anycast, but just enough control to re-route.



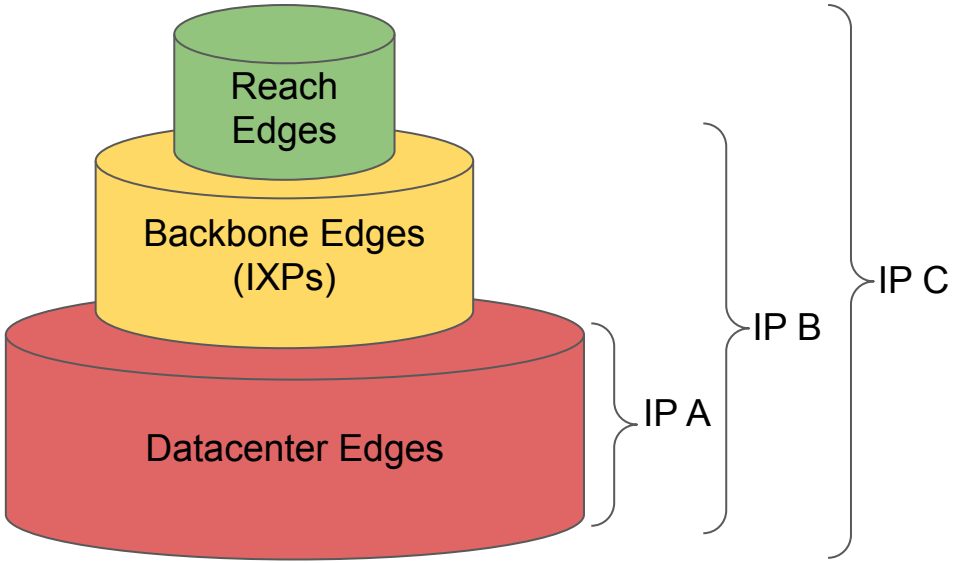
Anycast “layers”: b/c not all edges are created equal

FastRoute: How to handle anycast overload

Microsoft

System goal: To be able to re-route anycast traffic when an edge gets overloaded.

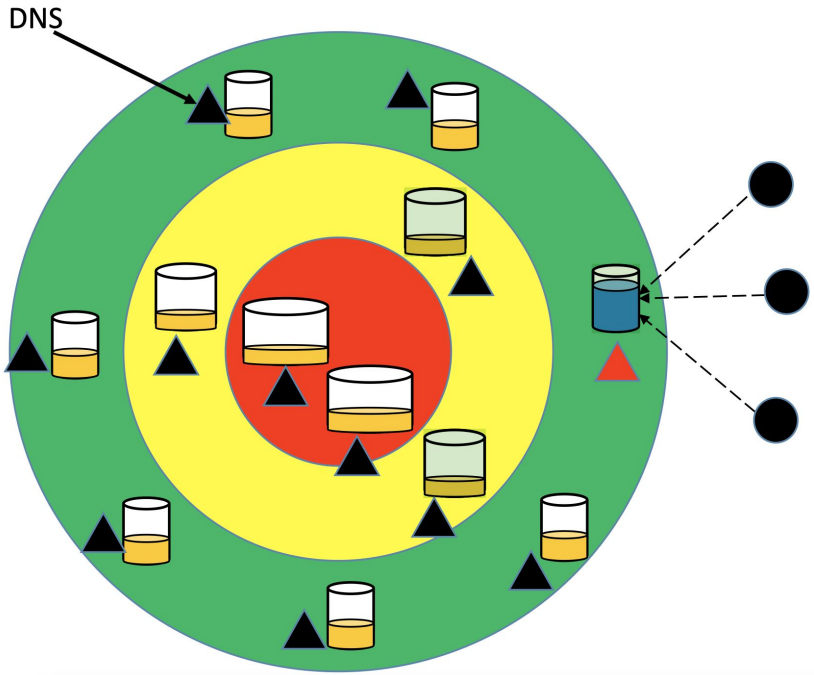
Solution design goal: Just as simple as anycast, but just enough control to re-route.



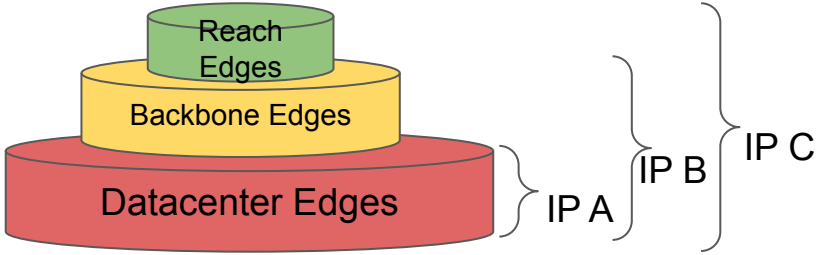
Anycast “layers”: b/c not all edges are created equal

Solution: DNS chooses which IP address layer to hand out to client. Then uses anycast to route to that group of nodes.

FastRoute: How to handle anycast overload

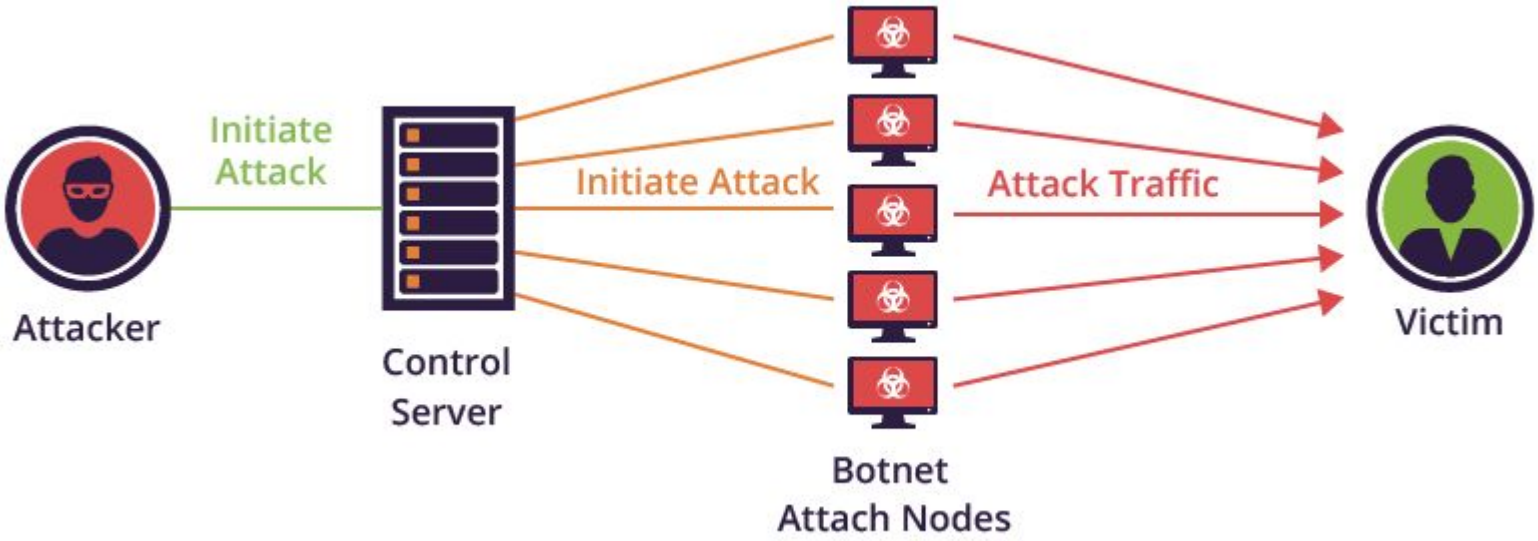


- DNS servers located in the same location as edges (e.g., reach edges) and they talk to each other.
- If DNS server detects that its neighboring edge is being overloaded, it starts handing out the IP address for the next layer of edges
- Still a decentralized approach!



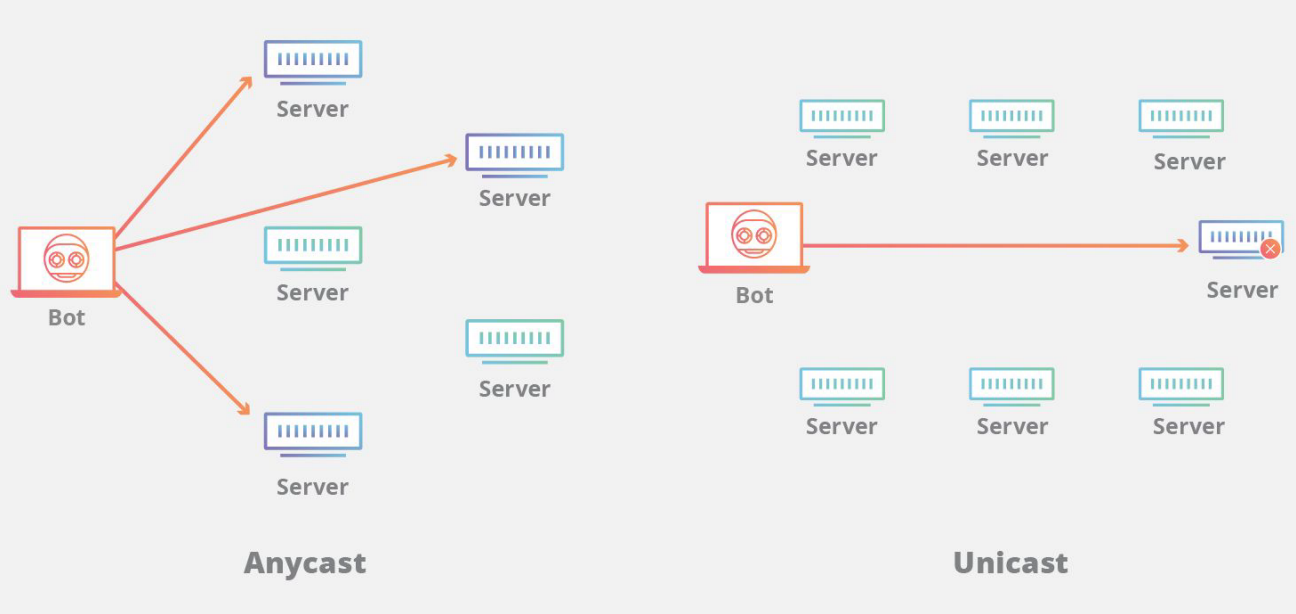
Anycast CDNs are perfect for mitigating DDoS attacks

DDoS attack: “Distributed Denial of Service”



Botnets are often made up of IoT devices that are distributed all over the world

Anycast CDNs are perfect for mitigating DDoS attacks



Scattered bot traffic will be distributed amongst many servers, thereby mitigating the denial of service

Cloudflare thwarts 17.2M rps DDoS attack — the largest ever reported (08/19/2021)