EDNS0 Client-Subnet for DNS based CDNs

Matt Jansen
Akamai Technologies

SANOG 24, Delhi, August 2nd 2014

The Akamai Intelligent Platform

The world's largest on-demand, distributed computing platform delivers all forms of web content and applications

The Akamai Intelligent Platform:

150,000+	2,000+	1,200+	700+	92
Servers	Locations	Networks	Cities	Countries

Typical daily traffic:

- More than **2 trillion** requests served
- Delivering over **21 Terabits/second**
- **15-30%** of all daily web traffic

How CDNs Work

When content is requested from CDNs, the user is directed to the optimal server to serve this user

There’s 2 common ways to do that:

- anycast: the content is served from the location the request is received (easy to build, requires symmetric routing to work well)
- DNS based: the CDN decides where to best serve the content from based on the resolver it receives the request from, and replies with the optimal server

How DNS based CDNs Work

Users querying a DNS-based CDNs will be returned different A (and AAAA) records for the same hostname depending on the resolver the request comes from

This is called “mapping”

The better the mapping, the better the CDN

How Akamai’s CDN works

Example of Akamai mapping

- Notice the different A records for different locations:

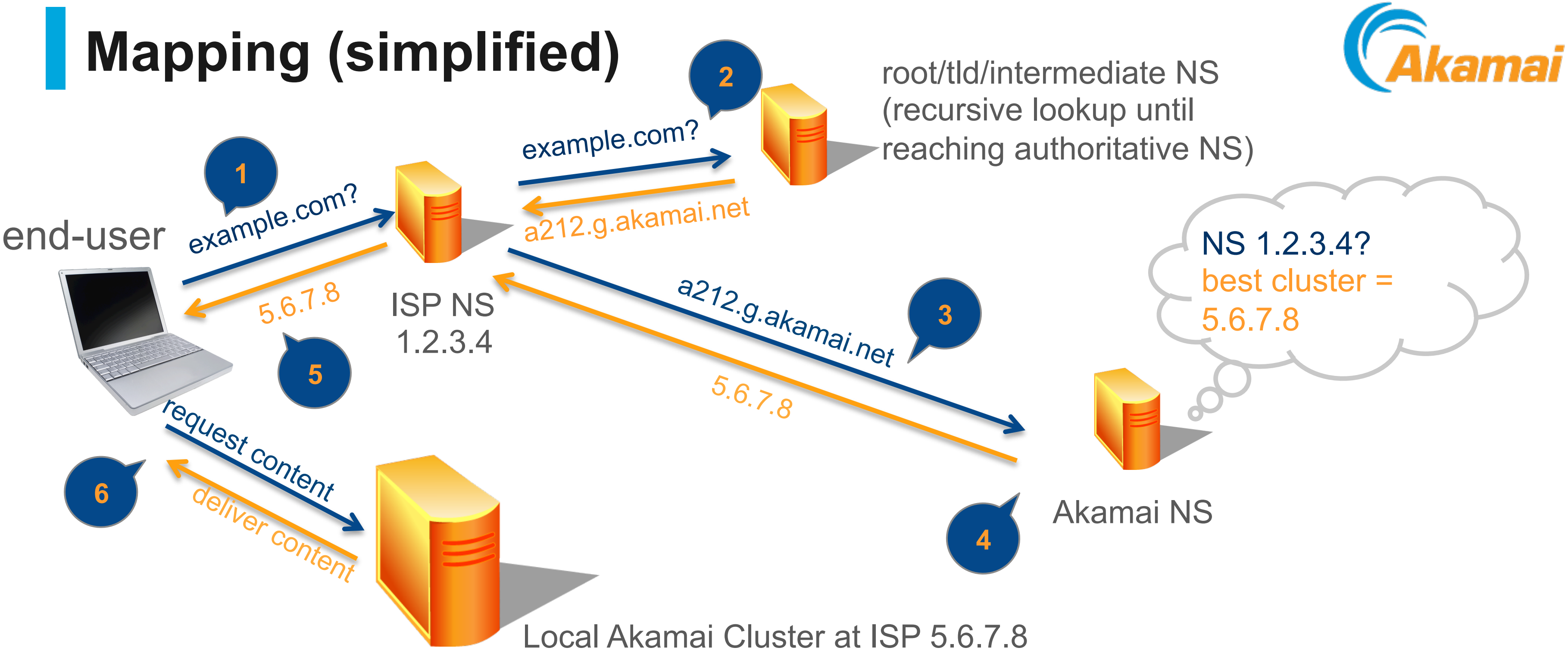
```
[NYC]% host www.symantec.com
www.symantec.com   CNAME  e5211.b.akamaiedge.net.
e5211.b.akamaiedge.net.  A      207.40.194.46
e5211.b.akamaiedge.net.  A      207.40.194.49

[Boston]% host www.symantec.com
www.symantec.com   CNAME  e5211.b.akamaiedge.net.
e5211.b.akamaiedge.net.  A      81.23.243.152
e5211.b.akamaiedge.net.  A      81.23.243.145
```

How Akamai's CDN works

Akamai uses multiple criteria to choose the optimal server

- These include standard network metrics:
  - Latency
  - Throughput
  - Packet loss
- as well as internal ones such as:
  - CPU load on the server
  - HD space
  - network utilization

Mapping (simplified)



1) end-user requests www.example.com from ISP NS
2) ISP NS recursively (multiple iterations) looks up www.example.com being referred to authoritative Akamai NS (by cname)
3) ISP NS asks authoritative Akamai NS
4) Akamai NS looks up IP of requestor (ISP NS) and replies with IP of optimal cluster to serve content (local cluster in that ISP)
5) ISP NS replies to end-user who
6) requests content from local Cluster

The Problem: 3rd Party DNS servers

All of this works very well if the end-user used their provider's DNS servers.

However if the end-user is making use of a 3rd party DNS service like

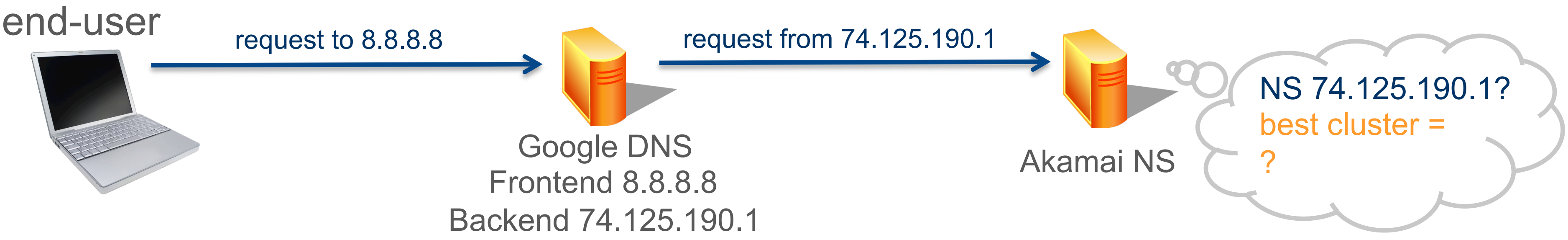
- Google DNS (28 locations worldwide)

  https://developers.google.com/speed/public-dns/faq#locations
- OpenDNS (20 locations worldwide)

  http://www.opendns.com/network-map/

a DNS-based CDN does not know which network the request originated from, and can therefore in the best case serve it in the rough geographic area

How 3rd party (open) resolvers typically work



global ‘frontend’ anycast address, local unique ‘backend’ address for recursive queries

- CDN can tell which NS location it came from (by backend-ip)
- but not which end-user location or network

-> have to serve from a large infrastructure cluster (typically located at the big IXs) to ensure we can reach any end-user

Use of 3rd party DNS servers

relatively small numbers in most countries with a mature internet ecosystem:

USA, Germany, Netherlands, Singapore: less than 1%

but very high percentage of users in developing countries and/or countries performing some form of DNS-based web-filtering:

Brazil: 12%,Turkey: 22%, Indonesia: 22%

Sri Lanka 7%, India: 10%, Nepal 11%, Bangladesh: 25%

Use of 3rd party DNS servers in India

	ISP DNS	Google	OpenDNS	Others
ISP A	85.9%	6.4%	1.5%	6.2%
ISP B	81.3%	9.4%	2.7%	6.6%
ISP C	80.1%	5.8%	0.8%	13.2%
ISP D	76.0%	4.6%	0.4%	18.9%
ISP E	71.2%	1.9%	0.0%	26.9%
ISP F	64.9%	4.4%	0.2%	30.5%

End User Mapping

Use end-user IP instead of NS IP for mapping

Problem: at the time of authoritative DNS answer end-user IP is not known yet

- HTTP redirect
  - Map based on DNS
  - Measure RTT of initial request from end-user received (and therefore IP known), if over threshold:
  - Redirect to better positioned server to reach end-user IP

Problem: slow, not suitable for small objects

The Solution: EDNS0 client-subnet

EDNS0 client-subnet

https://tools.ietf.org/html/draft-vandergaast-edns-client-subnet-02

The recursive resolver includes the end-user’s prefix in the request to the authoritative nameserver

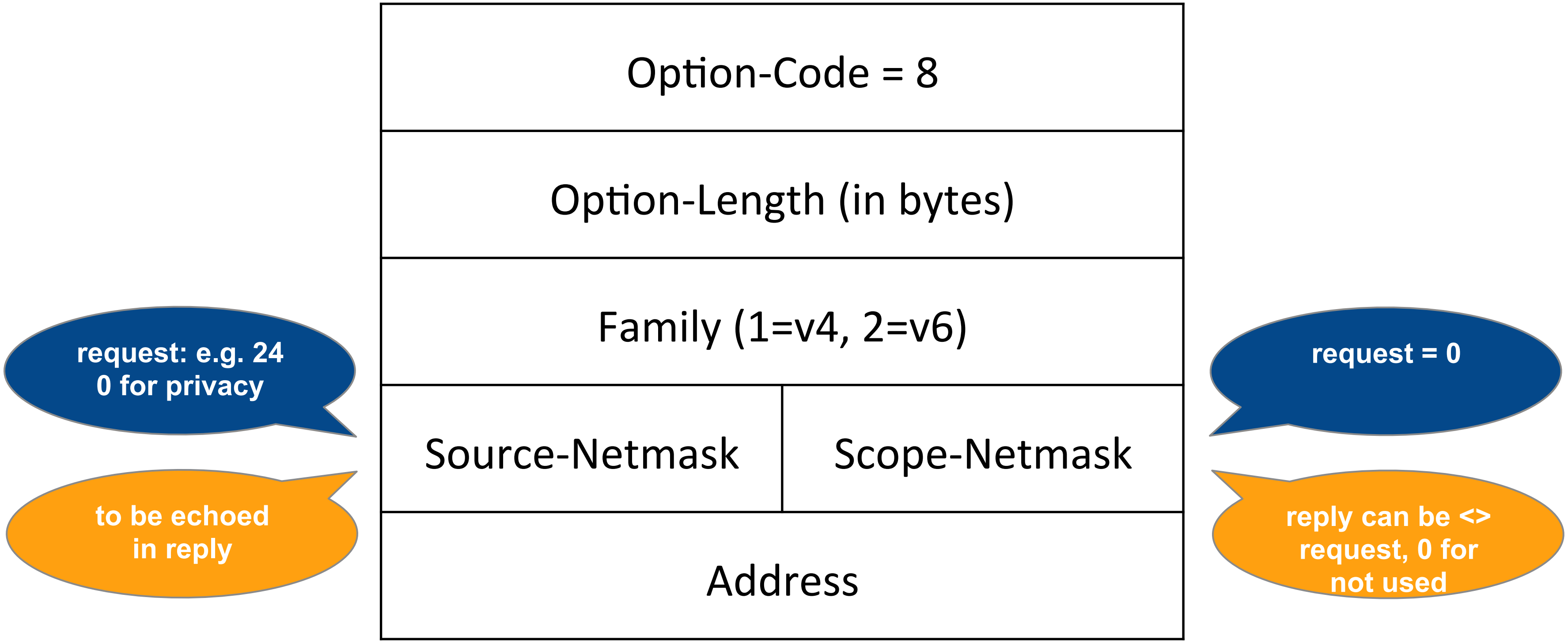
This allows the authoritative nameserver (the CDN) to process this information and optimize the reply not based on the requesting nameserver but the end-user’s prefix

The Solution: EDNS0 client-subnet

- Open standard (draft)
- Has to be supported by recursive resolver (3rd Party DNS)
- and by Authoritative NS (CDN)

- Privacy: only prefix, not full address transmitted

EDNS0 client-subnet implementation



Option-Code = 8

Option-Length (in bytes)

Family (1=v4, 2=v6)

Source-Netmask

Scope-Netmask

Address

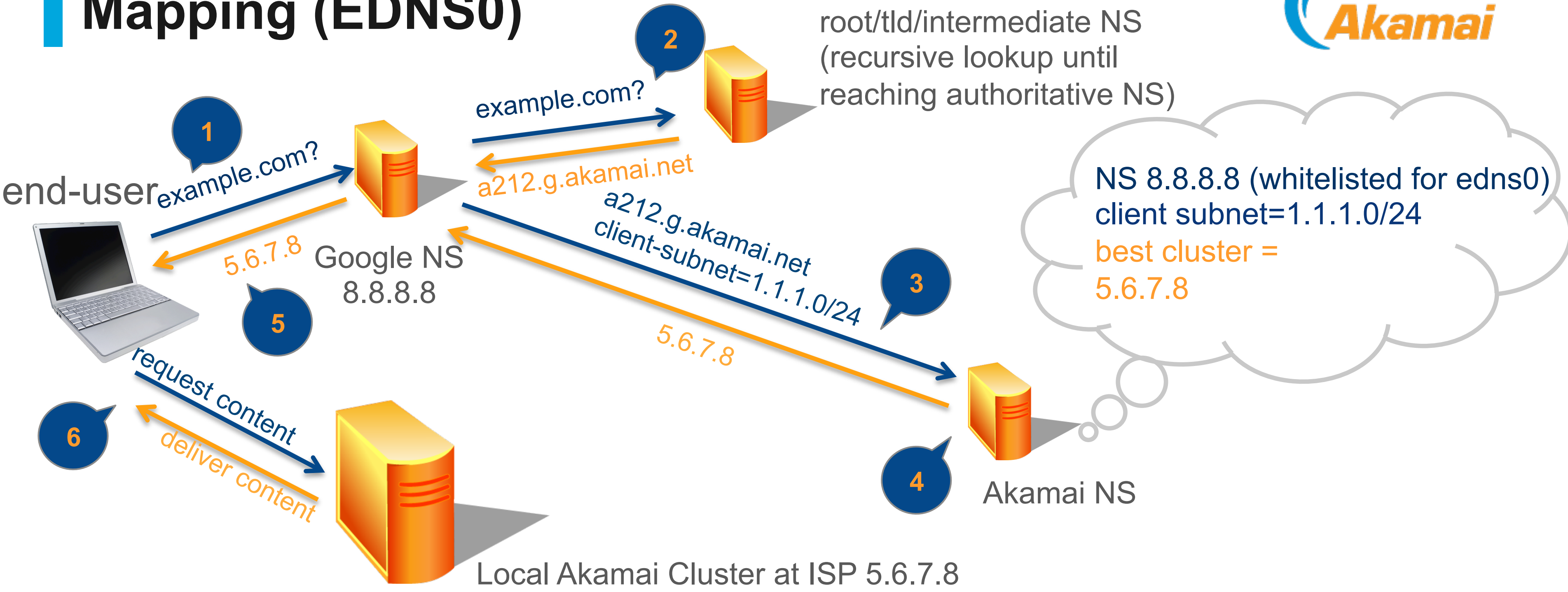
request: e.g. 24
0 for privacy

to be echoed
in reply

request = 0

reply can be <>
request, 0 for
not used

Mapping (EDNS0)



1) end-user requests www.example.com from Google NS
2) Google NS recursively looks up www.example.com being referred to authoritative Akamai NS (by cname)
3) Google NS asks Akamai NS including client-subnet
4) Akamai NS looks up client-subnet and replies with IP of optimal cluster to serve content (local cluster in that ISP)
5) ISP NS replies to end-user who
6) requests content from local Cluster

Privacy concerns

Only prefix, not full IP transmitted

CDN already gets your full IP anyways (in the subsequent HTTP request)

Set source-netmask/address to 0.0.0.0/0

- Google DNS honors forwards request with 0.0.0.0/0
- OpenDNS ignores at time of writing

Do not use client-subnet capable resolver if intention is to hide client origin

Security concerns

Scanning/walking the mapping algorithm

- double whitelist (at recursive resolver & auth NS)
- enforced replacement of client-tagged edns0 option by Google & OpenDNS before being send to Akamai

Amplification

- double whitelist
- echoing request in reply
- standard rate limiting methods work

Cache pollution of recursive resolver can be a problem

- separate reply stored for each prefix

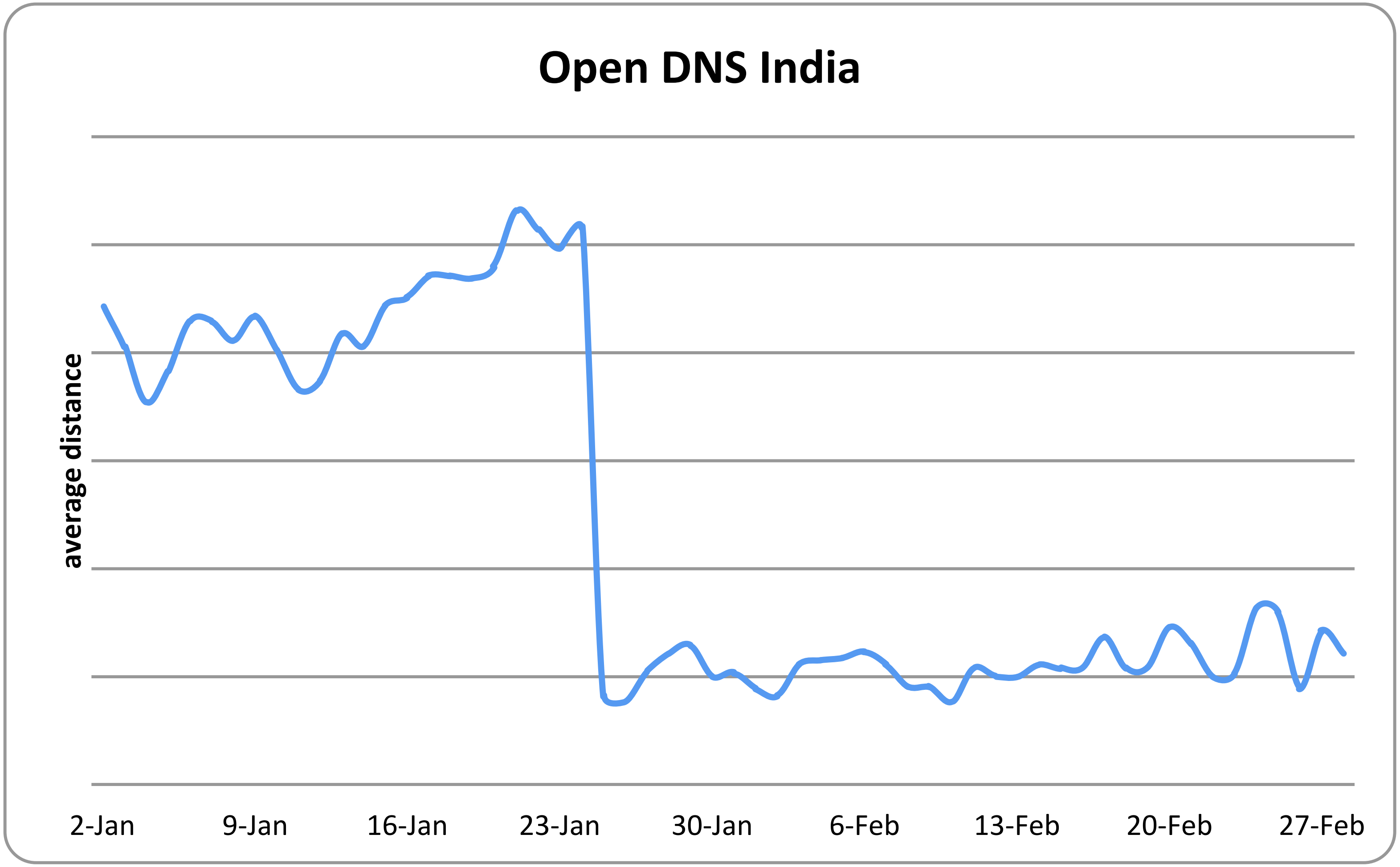
Prefix-Length

Google/OpenDNS currently always send client-subnet as /24 (for privacy/caching-efficiency reasons)

Mapping system has view of internet from it’s partners with differing prefix-lenghts

- client-subnet more specific than Akamai
  - e.g. Akamai has /20 from partner-> can be mapped
  - scope-netmask send to resolver for caching purposes
- client-subnet less specific than Akamai
  - e.g. Akamai has /26s from partner in different locations -> no clear choice to map -> will take first match
  - also send scope-netmask to resolver for information

Improvements with edns0 client-subnet



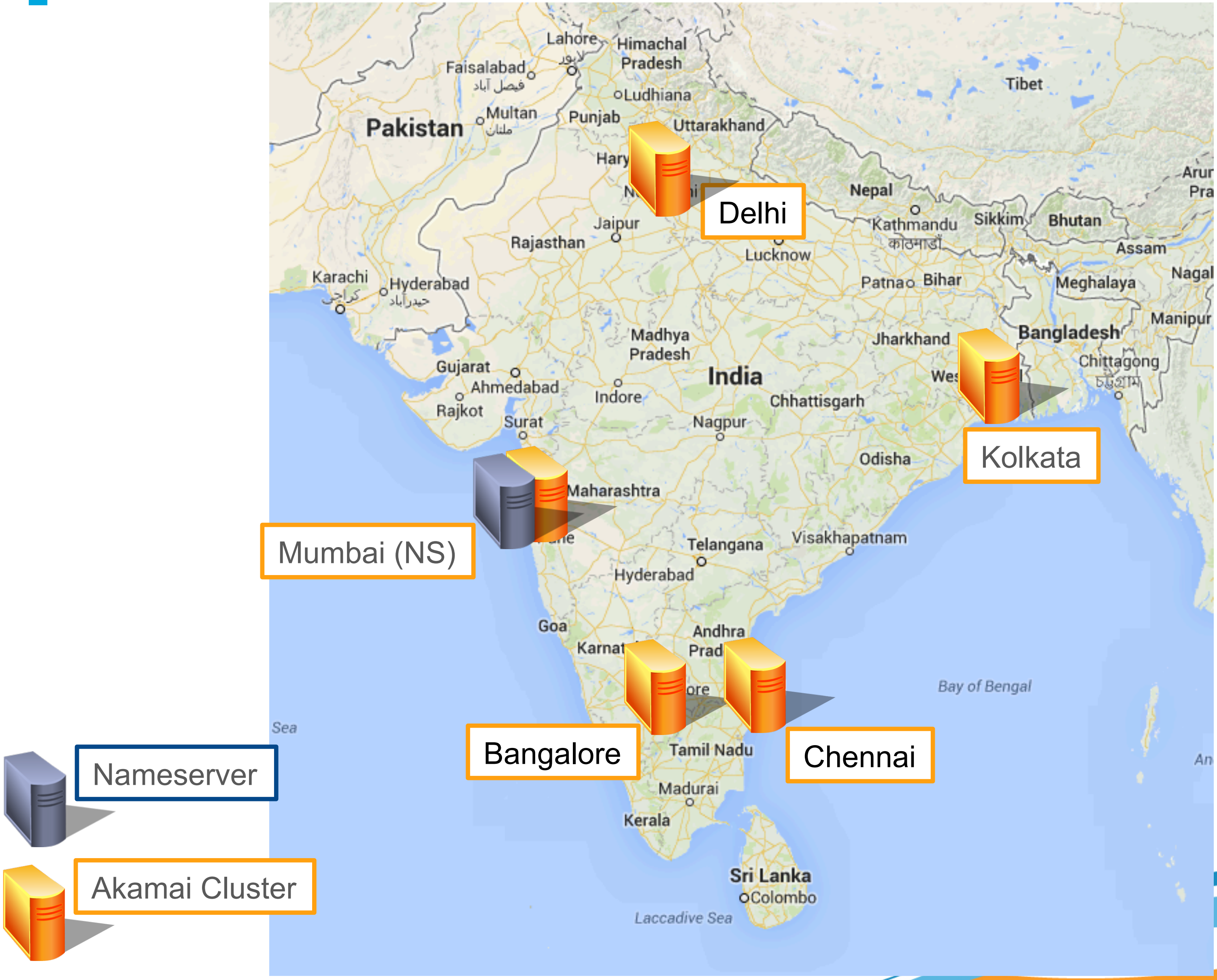
Additional Use-Case

can be used within a partner's network instead of distributed DNS architecture

A partner might have a widespread network (especially in countries spanning large geographical areas and/or different islands)

- Would like to deploy clusters around the network to localize traffic
- But central DNS infrastructure makes mapping traffic accurately difficult

Example for distributed architecture



Delhi

Kolkata

Mumbai (NS)

Bangalore

Chennai

Nameserver

Akamai Cluster

Solutions

Deploy additional NS in all locations

- Benefit: better DNS responses, can use anycast frontend IP to simplify administration/failover (announcing same frontend IP to all end-users)
- Drawback: additional CAPEX & support-costs

Virtual IPs on existing NS given to different geographic sets of end-users

- Benefit: no additional CAPEX, easy to implement
- Drawback: more difficult to administer, will require manual allocation of IPs to clusters on CDN side, no clear fallback

EDNS0 client-subnet within the providers network

- Benefit: no additional CAPEX, only software change on the NS, can dynamically adapt by changing announcements, can scale for very small clusters in remote places
- Drawback: needs compatible NS software

Questions?

Matt Jansen mj@akamai.com