Akamai & ISPs

Patrick W. Gilmore, Chief Network Architect

UKNOF25

April 18, 2013

Agenda

1. Disclaimer
2. Rules
3. What is a CDN, types of CDNs
4. Akamai's topology
5. Peering, from both sides
6. Mapping

Disclaimer

I was asked to give a talk about Akamai's CDN, not CDNs in general

While some of the information may be general enough to apply to all CDNs, I made no effort to generalize the information

In Other Words: Your Mileage May Vary

A Few Simple Rules

1. There are exceptions to every rule

   When I say “X == Y”, please hear “except for these few corner cases” even if I do not say it

2. This is very high level

   We only have 30 minutes, I need to gloss over some details

3. Questions are welcome & encouraged

   This is for you, be sure you get the most out of it

What is a Content Distribution Network?

The RFCs and Internet Drafts define a Content Distribution Network, “CDN”, as:

> **Content Delivery Network or Content Distribution Network. A type of CONTENT NETWORK in which the CONTENT NETWORK ELEMENTS are arranged for more effective delivery of CONTENT to CLIENTS.**

What is a CDN - In English? (Or at least American?)

A CDN is an overlay network, designed to deliver content from the optimal location

Very Generally: *Users* in Tokyo go to a server in Tokyo, users in Frankfurt go to a server in Frankfurt

This obviously over-simplifies things, as “optimal” is frequently not equivalent to “geographically close” – topology matters

- Obviously serving users in Beijing from Johannesburg is unlikely to be optimal for a global CDN no matter the topology

To Network Or Not To Network

Some CDNs have a network (i.e. backbone)

- E.g. Level 3, Limelight
- Typically CDNs owned by a network will have a network (shocker)

Some CDNs do not

- E.g. Akamai, EdgeCast, CloudFlare

Network-based CDNs have most of their servers in their own CDN

Non-Network CDNs can place servers directly in other ASNs

- Which means you cannot find their traffic with NetFlow

Akamai's CDN (Requisite Marketing Slide)

Akamai is the largest CDN in the world

- 3rd party estimates show Akamai's traffic equal to or greater than all other CDNs combined

The Akamai EdgePlatform:

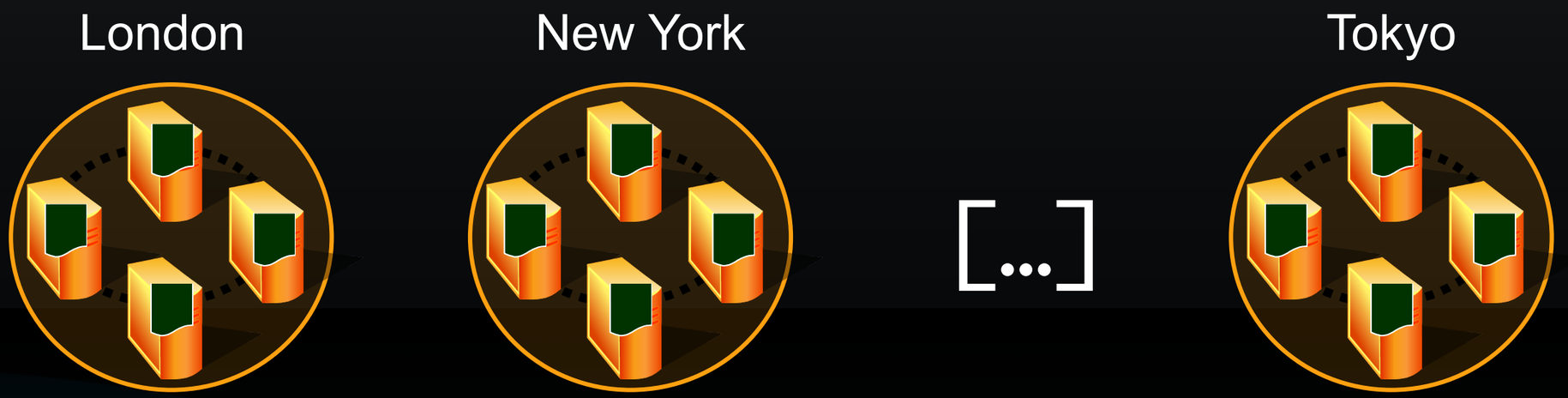
130,000+ Servers	~2,200 POPs	~1,200 Networks	800+ Cities	81 Countries

Delivering:
30+ million hits / second
1.7+ trillion hits / day
Double-digit Tbps

Akamai's CDN

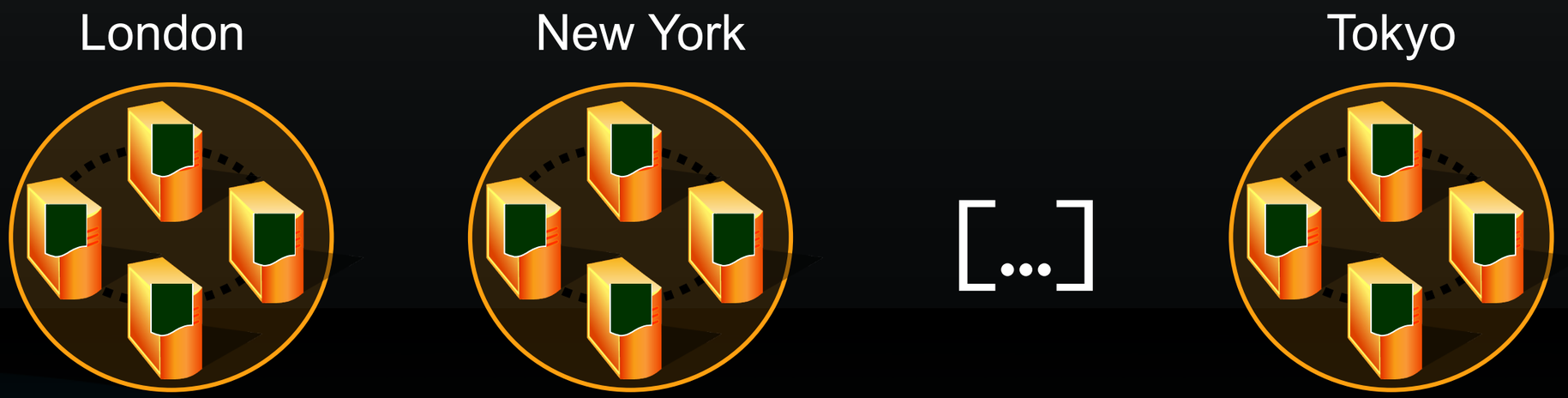
Akamai's CDN is comprised of distinct, geographically & topologically disparate nodes

We believe having lots of nodes in lots of places gives us better performance than a few large sites



No Backbone

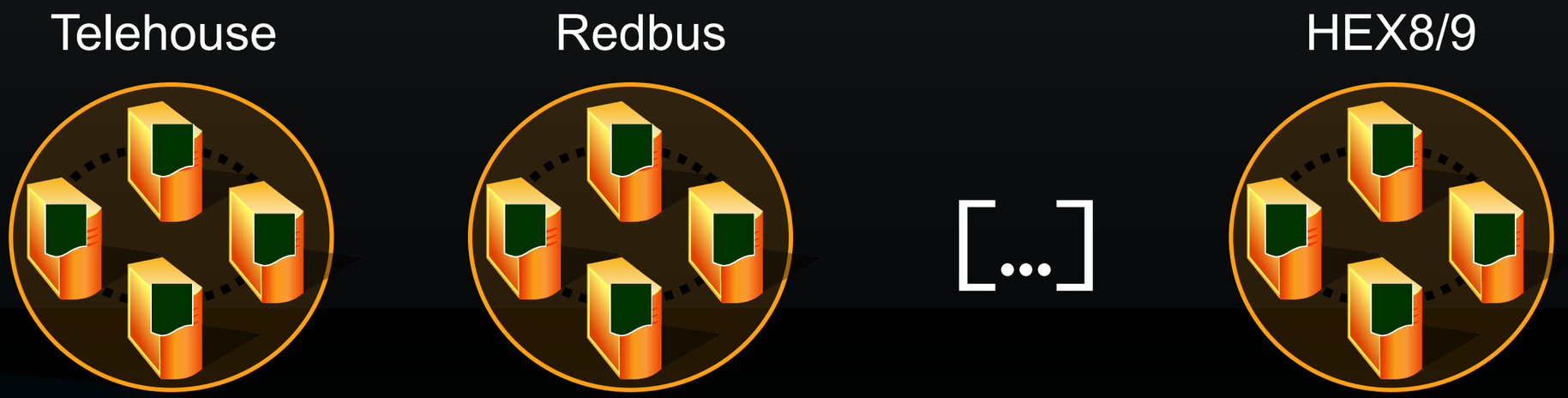
It is important to realize there is no network between Akamai nodes



No Backbone

It is important to realize there is no network between Akamai

Even if they are in the same city

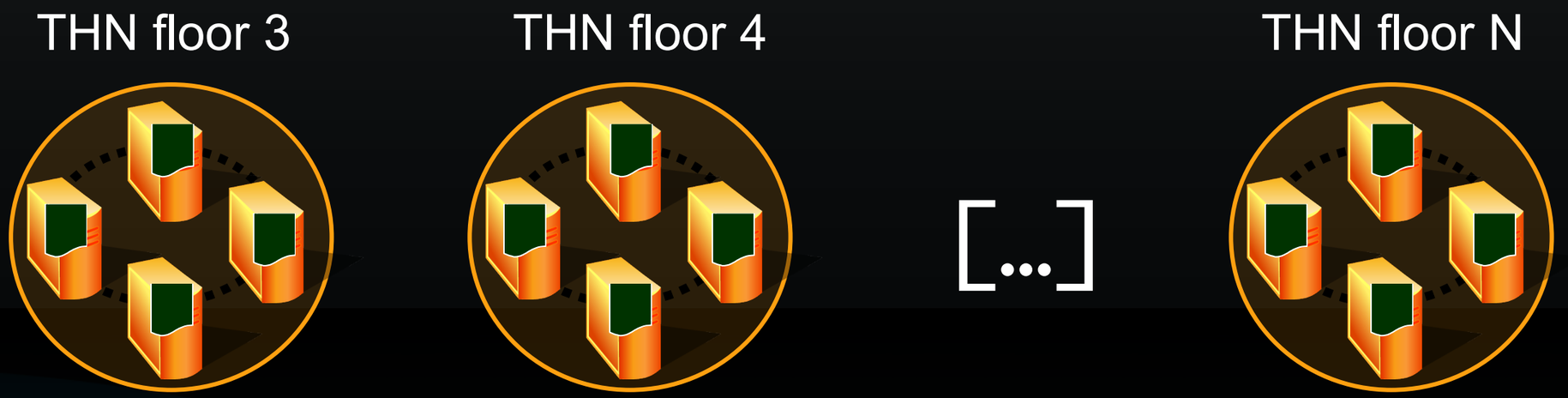


No Backbone

It is important to realize there is no network between Akamai

Even if they are in the same city

In fact, Akamai nodes in the same ***building*** do not share traffic



Origin GETs

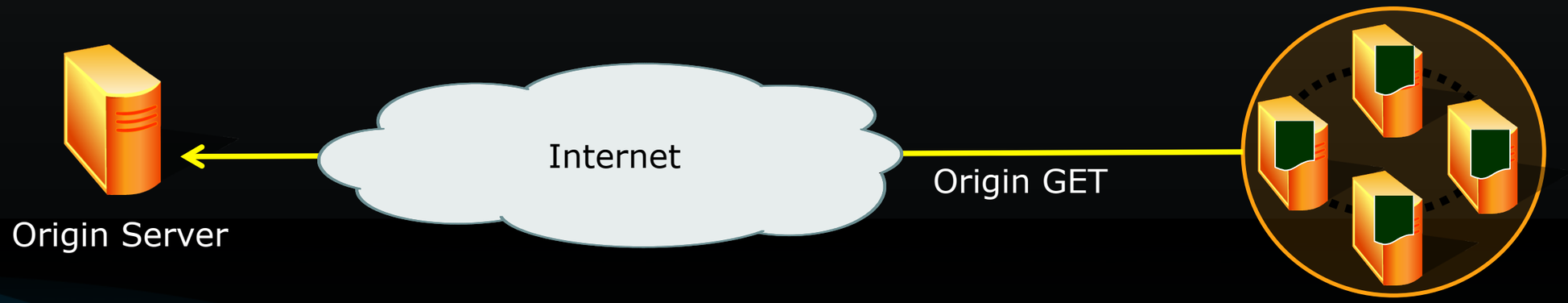
Since each node is an island, there is no way for Akamai to deliver content to a node ourselves

Origin GETs

Since each node is an island, there is no way for Akamai to deliver content to a node ourselves

Akamai's "backbone" is the Internet

- Also used for log delivery, SSH'ing to servers, etc.



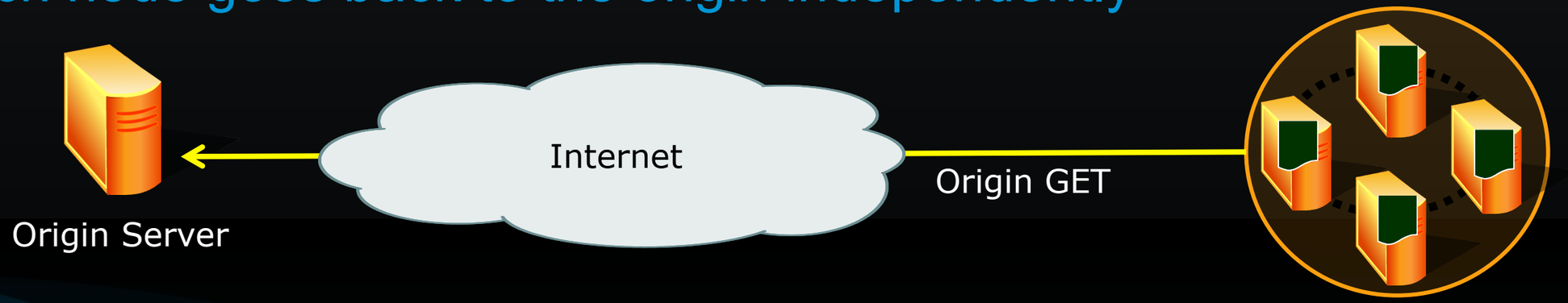
Origin GETs

Since each node is an island, there is no way for Akamai to deliver content to a node ourselves

Akamai's "backbone" is the Internet

- Also used for log delivery, SSH'ing to servers, etc.

Each node goes back to the origin independently



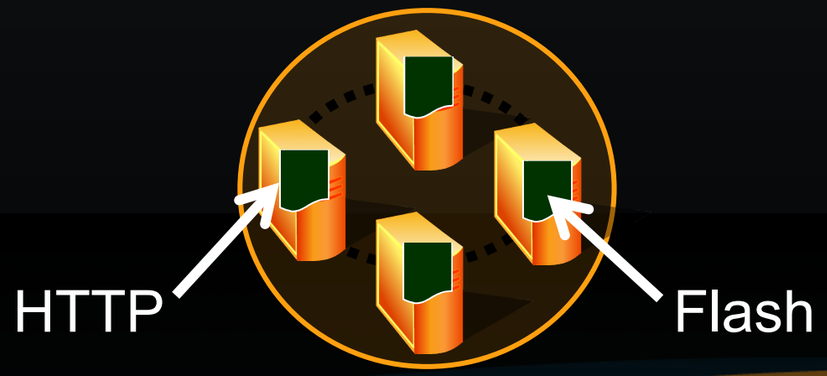
Large Akamai Nodes, up close & personal

Large nodes have a mix of content types

Some servers will serve Flash streaming, others serve HTTP, etc.

Different content types typically require separate physical machines

- Flash works best on Windows (ugh), HTTP runs on unix (yay), etc.
- But also because different software requires different hardware, does not play well with others, etc.

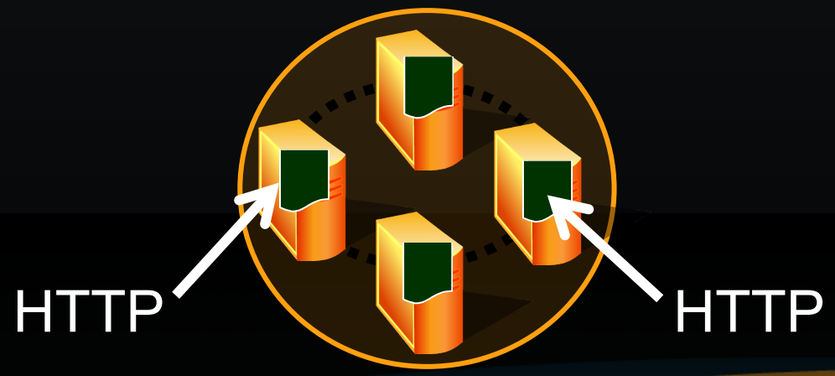


Small Akamai Nodes, up close & personal

Small nodes have a single server type (HTTP)

Each content type requires a minimum number of machines (usually 3-5), ruling out multiple content types on some nodes

Even nodes with enough machines may only serve HTTP



Small Akamai Nodes, up close & personal (2)

In addition to only serving HTTP, small nodes cannot serve all HTTP content

The fact there are only a few servers limits the amount of storage, meaning not all content can be cached

Because very few networks have enough traffic to require nodes large enough to carry all content types and all customers, it is important to peer with Akamai even if you have an on-net node

Why Akamai peers with ISPs - performance

The first and foremost reason to peer is improved performance

- Since Akamai's entire reason for existence is to improve performance, peering directly with networks (over non-congested links) obviously helps

Traffic served over peering typically performs better than over transit

- Sometimes there is no difference or (rarely) peering is worse, but no one here is like that, right?

Removing intermediate AS hops allows higher peak traffic for the majority of end user Ases

- I cannot prove why this is true, but we have hard data showing it is
- Anyone have ideas? (I have a few, but would like to hear yours)

Why Akamai peers with ISPs (2)

Lots of other reasons:

- Redundancy
- Burstability
- Network Intelligence
- Backup for on-net servers
- Serving additional content types

While all those are important, they really all are related to performance

So the real second reason to peer is cost (duh)

Why you should peer with Akamai

Why not?

- CDNs and ISPs are in the same business, just on different sides - we both want to serve end users as quickly and reliably as possible

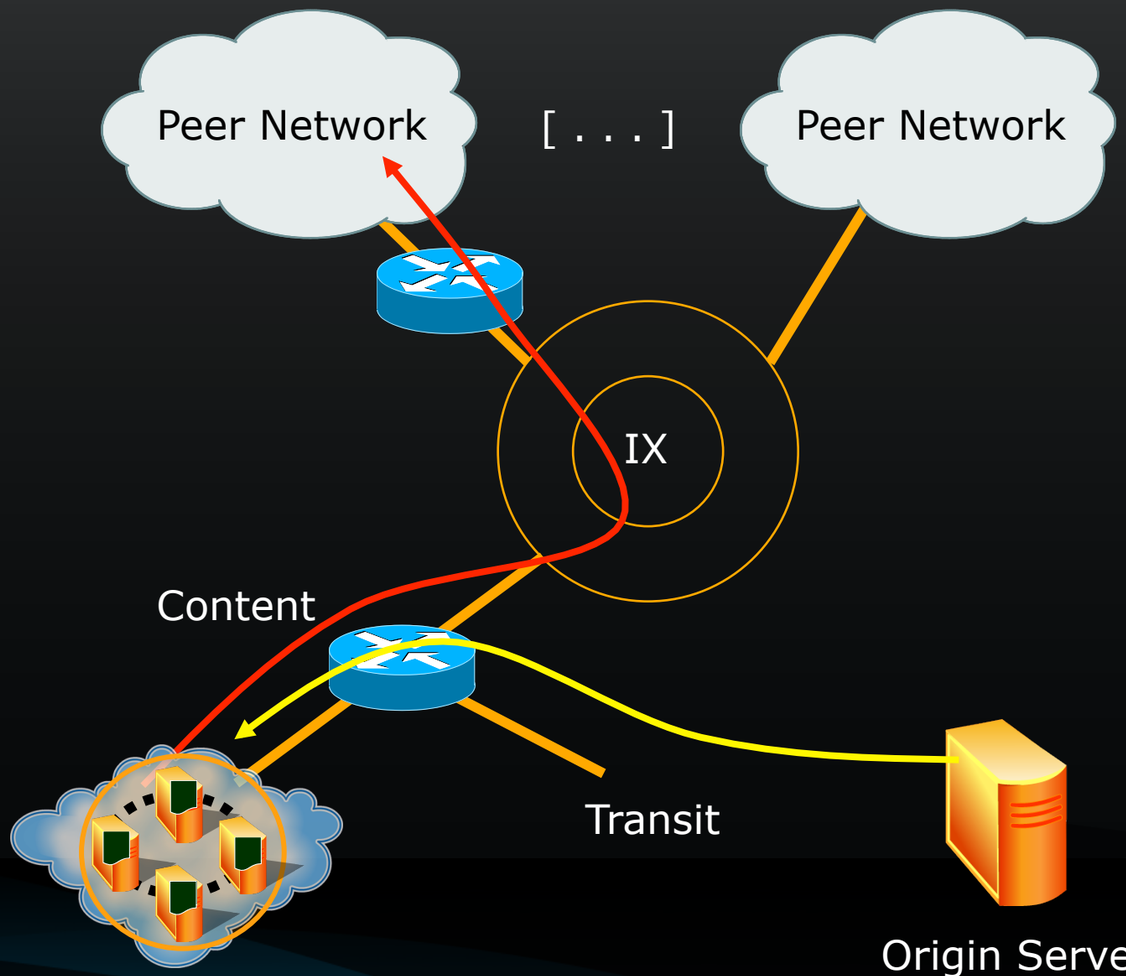
Cost Reduction

- Transit savings
- Possible backbone / backhaul savings

Because you are nice

- Doesn’t everyone want to be nice?

Akamai at IXPs

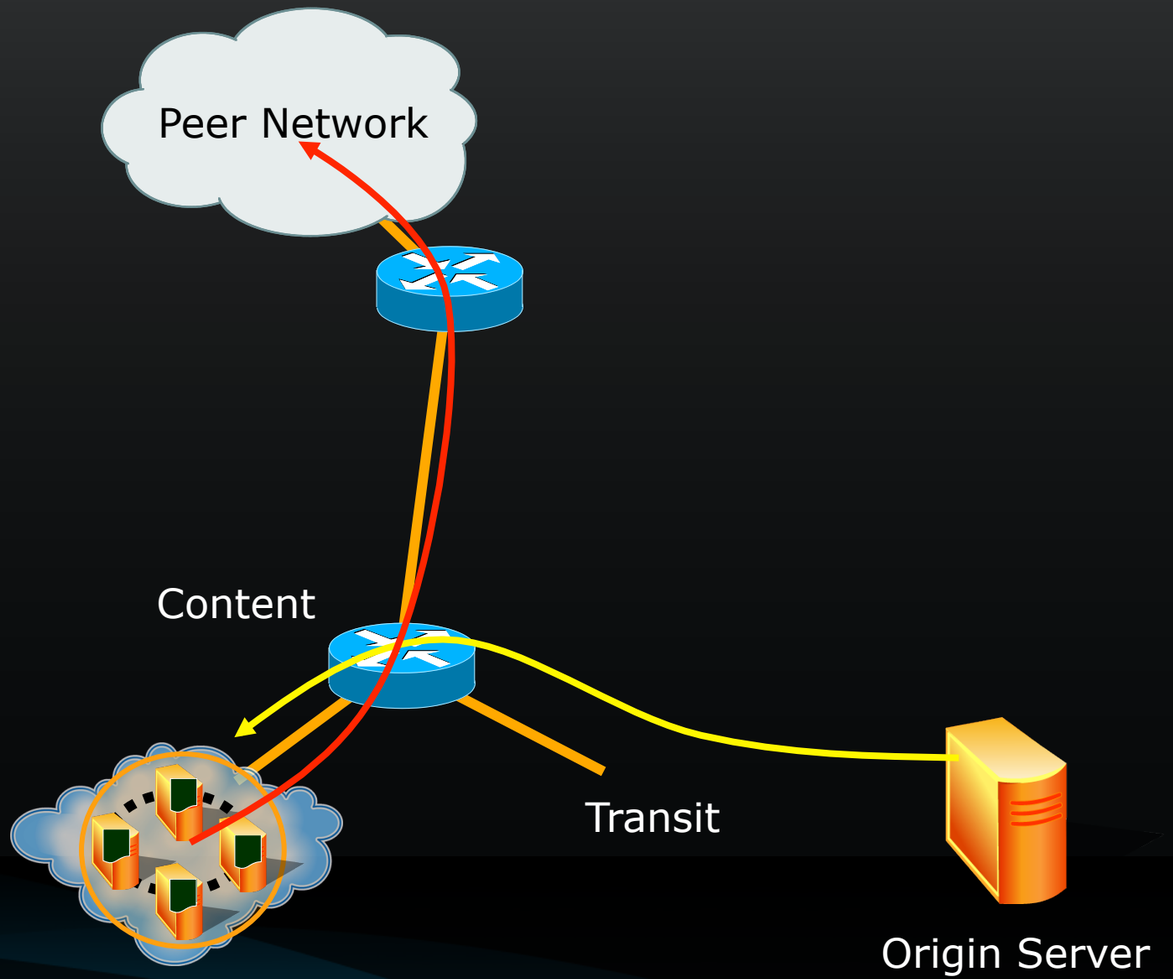


Remember each node is an island

1. The CDN uses transit to pull content into the servers
2. Content is then served to peers over the IX

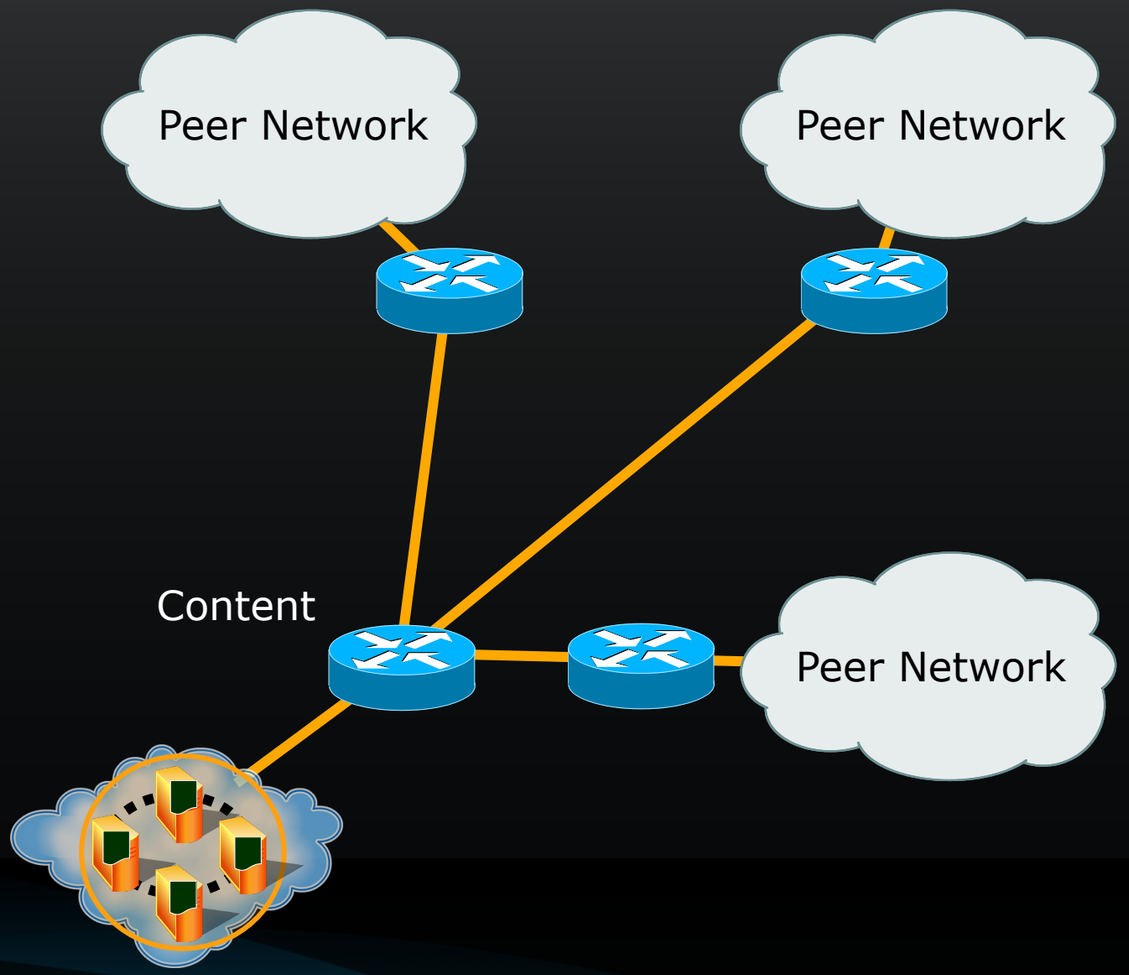
This is designed specifically to appear exactly like any other peer

Akamai & Private Peering



Private peering is just the degenerate case of an IXP – i.e. an IX with one peer

Akamai & Private Peering



Private peering is just the degenerate case of an IXP – i.e. an IX with one peer

Of course, Akamai usually sets up more than one peer per node to get economies of scale

Akamai's traffic control

Akamai also has (IMHO) amazing control over our traffic

This is an actual graph of an Akamai IX port over a week

- Time for an upgrade maybe?

Mapping

Most people think of Akamai as a caching company

Caching objects is something Akamai does, but the Akamai's core business is Mapping the Internet

Akamai makes these decisions in near-real time, adjusting to performance changes in 10-30 seconds

Deciding which end users should go to which web (streaming, whatever) servers is hard

Mapping & DNS

Akamai maps end users through DNS

Specifically, querying the same hostname from different locations will return different A records

Traffic is still routed to the end user directly

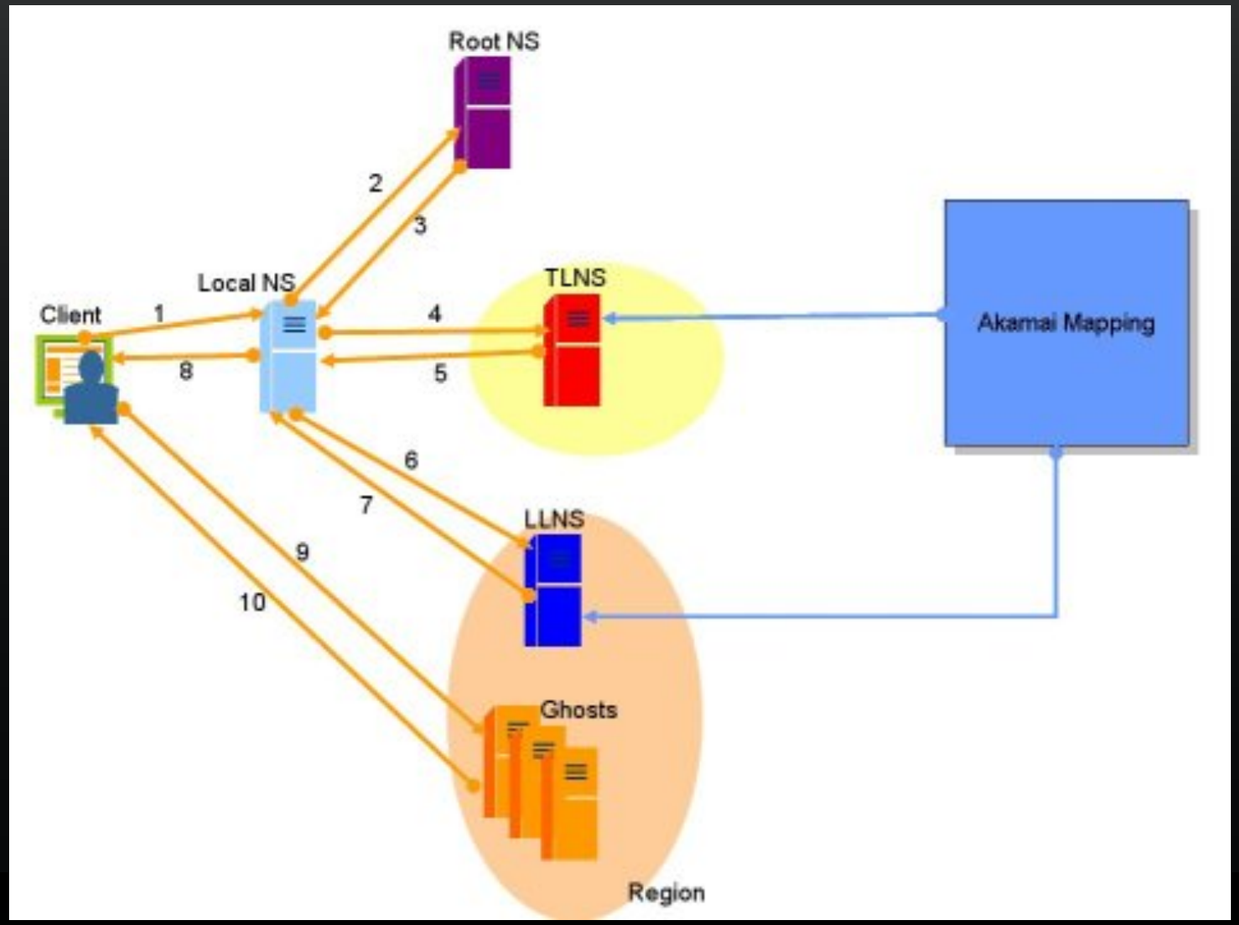
- No transparent caching
- No WCCP
- No HTTP redirects
- Etc.

Mapping Decision Steps

Akamai's Mapping has a few steps:

1. User types www.customer.com into browser
2. User's machine goes to ISP's recursive name server
3. ISP's RNS goes to [roots, GTLDs] customer's NS and asks for A record
4. Akamai responds with CNAM to a1.g.akamai.net
5. ISP's RNS goes to [roots, GTLDs] Akamai's NS and asks for a1.g.akamai.net
6. Akamai responds with a delegation for g.akamai.net
7. ISP's RNS asks Akamai's second level NS for a1.g.akamai.net
8. Akamai responds with at least two IP addresses
9. ISP's RNS answers user's machine

Akamai's DNS mapping visual aid



TLNS == Top Level NS

- I.e. IP address handed out by the GTLDs

LLNS == Low Level NS

- I.e. IP address delegated by the TLNS

"Ghosts" (should be "GHost") is Akamai's code name for our web servers: "Global Host"

Users vs. Name Servers

Note at no time during the Mapping decision, did the Akamai name server ever speak directly to the user

Even if we wanted to map by user IP address, we could not as we do not have it when we make the Mapping decision

Moreover, even if we did, the ISP's RNS caches the answer and hands the same IP address to the next end user without asking Akamai again

Users vs. Name Servers (2)

This means if a user configures an off-net name server Akamai will map the user where the name server is, not where the user is

If you dig an Akamai hostname against a name server in Tokyo or San Jose from a machine, you will get an Akamai server in Tokyo or San Jose

- Ignoring the fact there shouldn't be any open recursive name servers…

The most obvious examples of this are OpenDNS and Google DNS

- Yes, we are working on fixes, but they will not work for all ISPs
- Easier just to assure your users are configured with a local name server

Example of Mapping

Example of CDN mapping

- Notice the different A records for different locations:

```
[London]% host www.symantec.com
www.symantec.com    CNAME   a568.d.akamai.net
a568.d.akamai.net   A       207.40.194.46
a568.d.akamai.net   A       207.40.194.49

[Boston]% host www.symantec.com
www.symantec.com    CNAME   a568.d.akamai.net
a568.d.akamai.net   A       81.23.243.152
a568.d.akamai.net   A       81.23.243.145
```

Mapping Selection Criteria

Akamai uses many metrics for Mapping, including the standard latency, packet loss, throughput

Akamai also includes things like CPU load, available storage, network utilization, where content is already cached, etc.

Geography still counts

- That whole speed-of-light thing
- 100G Ethernet solves that, right … ?

Example of bad Mapping

Tracing from IAD to www.facebook.com

```
pgilmore@prod-unix-shell01:~> nsh 72.246.30.12 mtr -d www.facebook.com
HOST: a72-246-30-12.deploy.akamai Loss%   Snt   Last   Avg  Best  Wrst StDev
  1. a72-246-30-1.deploy.akamaite  0.0%    10    0.4   0.4   0.3   0.5   0.1
  2. dc5.pr01.iad2.tfbnw.net       0.0%    10    0.3   0.3   0.2   0.5   0.1
  3. ae1.bb02.iad2.tfbnw.net       0.0%    10   16.9   3.0   0.5  16.9   5.5
  4. ae11.bb02.sjc1.tfbnw.net      0.0%    10   80.0  82.9  78.0  89.3   3.3
  5. ae2.dr01.snc5.tfbnw.net       0.0%    10   82.8  87.2  82.6 127.2  14.0
  6. po510.csw02a.snc5.tfbnw.net   0.0%    10   82.1  81.9  81.7  82.1   0.1
  7. www-11-02-snc5.facebook.com   0.0%    10   72.4  72.9  72.4  76.7   1.3
```

Example of bad Mapping

Tracing from SJC to www.facebook.com

```
pgilmore@prod-unix-shell01:~> nsh 173.223.232.110 mtr -d www.facebook.com
HOST: a173-223-232-110.deploy.aka Loss%   Snt   Last   Avg  Best  Wrst StDev
  1. a72-246-53-1.deploy.akamaite  0.0%    10    4.3   3.3   0.4   9.7   3.8
  2. paix.br01.pao1.tfbnw.net      0.0%    10    0.2   4.7   0.2  45.3  14.3
  3. ae8.bb02.pao1.tfbnw.net       0.0%    10    0.4   5.0   0.4  44.8  14.0
  4. ae8.bb02.iad1.tfbnw.net       0.0%    10   78.6  76.5  76.2  78.6   0.8
  5. ae1.dr02.ash4.tfbnw.net       0.0%    10   76.1  77.7  76.1  91.9   5.0
  6. po509.csw01a.ash4.tfbnw.net   0.0%    10   78.5  78.5  78.4  78.6   0.1
  7. www-11-01-ash4.facebook.com   0.0%    10   76.3  76.4  76.3  77.0   0.2
```

Example of good Mapping

Tracing from SJC to www.symantec.com

```
pgilmore@prod-unix-shell01:~> nsh 173.223.232.110 mtr -d www.symantec.com
HOST: a173-223-232-110.deploy.aka Loss%   Snt   Last   Avg  Best  Wrst StDev
  1. a173-223-232-105.deploy.akam  0.0%    10    0.1   0.2   0.1   1.6   0.5
```

OK ,that was cheating ☺

Example of good Mapping

Tracing from a server in ORD to www.symantec.com

```
inv2824# traceroute www.symantec.com
traceroute: Warning: www.symantec.com has multiple addresses; using 64.211.144.66
traceroute to a568.d.akamai.net (64.211.144.66), 64 hops max, 40 byte packets
 1  fe0-23.aggr3004.ord2.us.scnet.net (75.102.4.185)  0.629 ms  0.535 ms  0.661 ms
 2  v1503.ar1.ord1.us.scnet.net (216.246.94.209)  21.738 ms  1.778 ms  0.513 ms
 3  ae0-81.cr1.ord1.us.nlayer.net (69.31.111.1)  0.282 ms  0.389 ms  0.292 ms
 4  ae1-30g.ar1.ord1.us.nlayer.net (69.31.111.134)  12.214 ms
    ae1.ar2.ord1.us.nlayer.net (69.31.111.146)  0.517 ms
    ae1-30g.ar1.ord1.us.nlayer.net (69.31.111.134)  16.753 ms
 5  te9-3.ar3.CHI2.gblx.net (69.31.110.233)  0.699 ms
    te6-3.ar2.CHI2.gblx.net (69.31.111.201)  1.000 ms
    te9-3.ar3.CHI2.gblx.net (69.31.110.233)  1.203 ms
 6  ge9-1-10G.ar2.CHI2.gblx.net (67.17.109.117)  0.687 ms  0.710 ms
    64.211.144.66 (64.211.144.66)  0.514 ms
```

Questions?

patrick@akamai.com