Addressing IPv6
A CDN perspective

@jta joão taveira araújo

RIPE 74
Addressing IPv6
A CDN perspective

@jta joão taveira araújo

RIPE 74
Addressing IPv6
A CDN perspective

your mileage may vary

@jta joão taveira araújo

RIPE 74
SFO
LHR
Unicast model

Points of Presence

SJC

LAX

clients
Unicast model

Prefix announcements

SJC

LAX

2012
Unicast model

DNS

SJC offset

IP address

SJC

LAX

2012
Unicast model

2012

SJC

LAX

SJC

LAX

DNS

SJC offset

LAX
IP allocation

/24

(most specific prefix)
IP allocation

SJC

/24
(most specific prefix)

SJC

2012
IP allocation

Hosts

/24
(most specific prefix)

SJC

2012
IP allocation

2012

SJC

/24 (most specific prefix)

Hosts

VIPs
Fate-sharing

**Hosts**

**VIPS**

/24 (most specific prefix)

SJC

2012
Everything is fine

📅 2013

🔥 fate-sharing between address types
🔥 unicast has poor fallback properties
💣 /24 per POP will run out at some point

💸 anycast to support apex domains and in-house DNS
Anycast model

Anycast IP address

Same prefix

SJC

LAX

DNS

offset

2014
Anycast model
Anycast model
Anycast model
Everything is fine

- fate-sharing between address types
- unicast has poor fallback properties
- /24 per POP will run out at some point
- anycast is hard to get right
- inbound path control is terrible
- overhead of running concurrent models
IP addressing

/24 (most specific prefix)
IP addressing

/24 (most specific prefix)

SJC
IP addressing

Hosts

VIPs

/24

(most specific prefix)
Everything is terrible

📅 2015

- fate-sharing between address types
- unicast has poor fallback properties
- IPv6 grew POPs allocation will run out at some point
- anycast is hard to get right
- inbound path control is terrible
- overhead of running concurrent models
- you should probably do IPv6
The good news

no first-mover advantage
  at least two competitors already offered IPv6

limited demand for IPv6
  more valued features: caching, purging, logging, stats, VCL
  already lost the very few customers who cared about IPv6
  wasn’t affecting our retention rate or growth

no need to rush, so clean slate
Everything is terrible

- fate-sharing between address types
- unicast has poor fallback properties
- outgrown IPv4 allocation scheme
- anycast is hard to get right
- inbound path control is terrible
- overhead of running concurrent models
Decouple address types
Decouple address types

VIPs
one-to-many mapping, service abstraction

Infrastructure
one-to-one mapping to a physical endpoint
Everything is terrible

- fate-sharing between address types
- unicast has poor fallback properties
- outgrown IPv4 allocation scheme
- anycast is hard to get right
- inbound path control is terrible
- overhead of running concurrent models
Everything is terrible

- fate-sharing between address types
- unicast has poor fallback properties
- outgrown IPv4 allocation scheme
- anycast is hard to get right
- inbound path control is terrible
- overhead of running concurrent models
Back ing anycast

~2015

anycast prefix

“unicast” prefix
(subnetted from anycast prefix)
Backing anycast

DNS offset

SYD

SYD ~2015
Backing anycast

~2015
Backing anycast ~2015
Everything is terrible

- fate-sharing between address types
- unicast has poor fallback properties
- outgrown IPv4 allocation scheme
- anycast is hard to get right
- inbound path control is terrible
- overhead of running concurrent models
Everything is terrible

2015

- fate-sharing between address types
- unicast has poor fallback properties
- outgrown IPv4 allocation scheme
- anycast is hard to get right
- inbound path control is terrible
- overhead of running concurrent models
Everything is terrible

2015

- Outgrown IPv4 allocation scheme
- Unicast has poor fallback properties
- Anycast is hard to get right
- Inbound path control is terrible
- Overhead of running concurrent models
VIPs
one-to-many mapping, service abstraction

Infrastructure
one-to-one mapping to a physical endpoint
VIP allocation

2016
Per provider planes

2016

132

136
VIP allocation

FRA SYD LAX ASIA

/48

ASIA

ASIA

ASIA

ASIA

2016
Impact on global table  

2016

number of VIP groups

number of provider planes
Impact on global table

- provider independent
- set no-export (more specific routes)

number of VIP groups

number of provider planes

2016
Impact on global table 2016

- Number of VIP groups
- Number of provider planes
- Provider independent
- Set no-export (more specific routes)
Locator / Identifier

www.example.com

64 bit service identifier
VIPs
one-to-many mapping, service abstraction

Infrastructure
one-to-one mapping to a physical endpoint
Infrastructure allocation

2016
Infrastructure allocation

2016
number of announcements ~ \[ \sum_{i \in \text{POPs}} |\text{providers}_i| \]
Impact on global table

\[ \sum_{i \in \text{POP}s} |\text{providers}_i| \]

- Number of announcements: \( \sum_{i \in \text{POP}s} |\text{providers}_i| \)
- Each infrastructure prefix in a POP is a /40
Impact on global table

\[ \text{number of announcements} \sim \sum_{i \in \text{POPs}} |\text{providers}_i| \]

up to 16 /44s

each infrastructure prefix in a POP is a /40
Everything is terrible

2015

- fate-sharing between address types
- unicast has poor fallback properties
- outgrown IPv4 allocation scheme
- anycast is hard to get right
- inbound path control is terrible
- overhead of running concurrent models
Everything is terrible

2016

- fate-sharing between address types
- unicast has poor fallback properties
- outgrown IPv4 allocation scheme
- anycast is hard to get right
- inbound path control is terrible
- overhead of running concurrent models
Everything is terrible

- fate-sharing between address types
- unicast has poor fallback properties
- outgrown IPv4 allocation scheme
- anycast is hard to get right
- inbound path control is terrible
- overhead of running concurrent models
Locator names

- vip.
- ntt.vip.
- sjc.global.vip.
- fra.inf.
- peering.fra.inf.

2016
Locator names

vip.
nntt.vip.
sjc.global.vip.

fra.inf.
peering.fra.inf.
Locator names

vip.
ntt.vip.
sjc.global.vip.
fra.inf.
peering.fra.inf.
Anchoring IPv4

→ bird-export.example
if locator ~ "*cogent.vip" && provider != "cogent" then reject;
if locator ~ "*.cogent.vip" && provider == "cogent" then set_no_export()
Summary

- decoupled address types
- graceful fallback
- VIP prefix mobility
- fine-grained inbound path control
- unified model based on locator names

- takes a long time
Intellectual heritage

ILNP  mobility, multi-homing, inbound TE

MP-TCP  resource pooling

re-ECN  information asymmetry in connectivity markets
ILNP  locators expose path diversity

MP-TCP  pool path diversity at transport and above

re-ECN  e2e metrics drive path selection
WIP

ILNP

MP-TCP

re-ECN

EU research from ~ 10 years ago
Either the questions don’t matter

**ILNP** mobility, multi-homing, inbound TE

**MP-TCP** resource pooling

**re-ECN** information asymmetry in connectivity markets
Either the ideas don’t work

- **ILNP**: locators expose path diversity
- **MP-TCP**: paths exposed to transport/app
- **re-ECN**: e2e metrics drive path selection
Questions

@jta joão taveira araújo

RIPE 74