DDoS, Peering, Automation and more

Martin J. Levy

AfPIF 2015 – Maputo, Mozambique

24th August 2015

Agenda

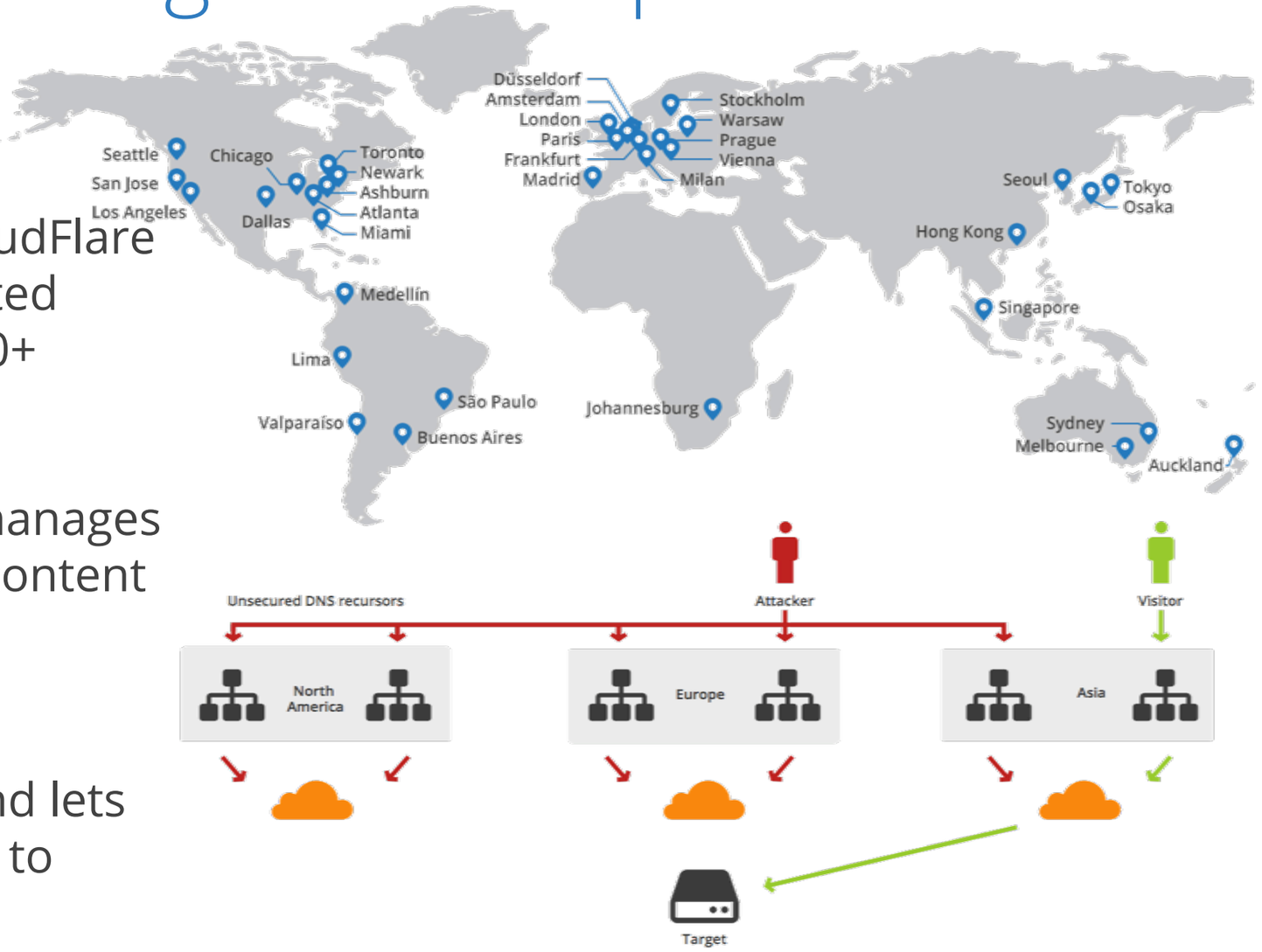
- Introduction to the CloudFlare network
  - How and where we deploy, peer, interconnect
  - Why distribute a DDoS mitigation and CDN service?
- Deploying 1,000's of servers, deploying replicated networking
  - Description of tools and more
- Peering and Interconnections at scale
  - A review of SANOG region and surrounding regions
- Fun things we do with massive servers and network gear
- Summary

Introduction to the CloudFlare network

CloudFlare global peering for DDoS protection

CloudFlare works at the network level

- Once a website is part of the CloudFlare community, its web traffic is routed through our global network of 30+ datacenters
- At each edge node, CloudFlare manages DNS, caching, bot-filtering, web content optimisation and third party app installations
- DDoS attack traffic is localized and lets other geographic areas continue to operate

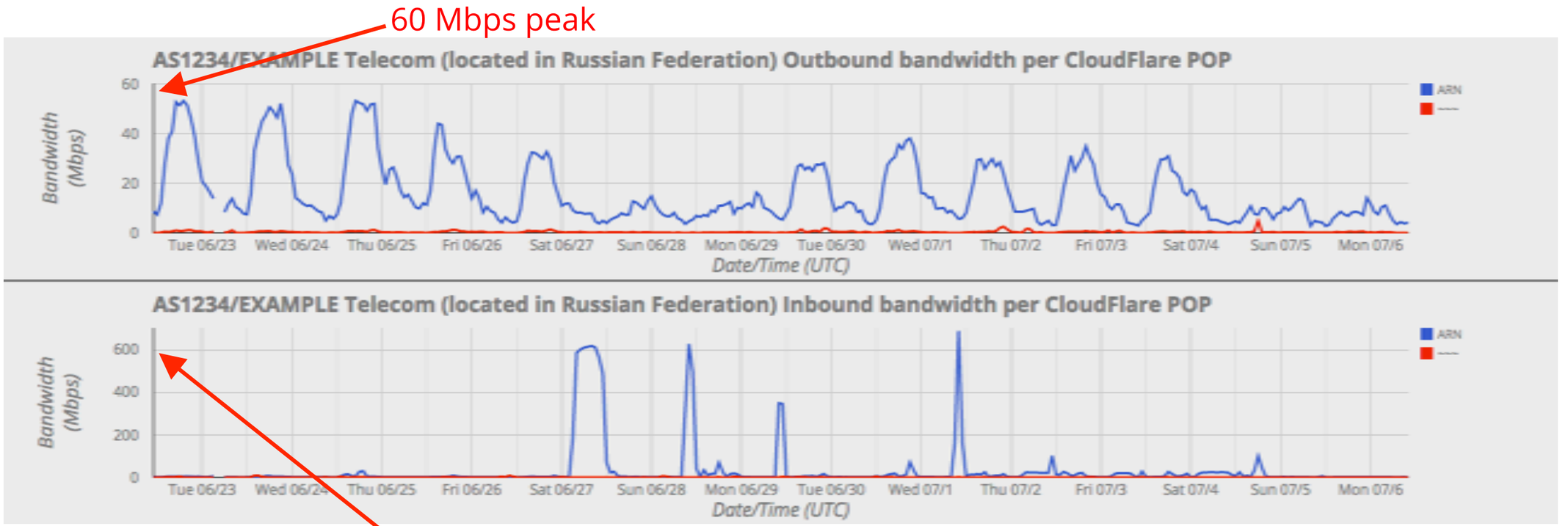


What does a DDoS attack look like?

DDoS look-and-feel

Our usual traffic ratio to eyeball ISPs is around 1:20 inbound:outbound

- However the ratio from the graph is 10:1 inbound:outbound
- The attacks shown on the graph are likely part of a much bigger global DDoS



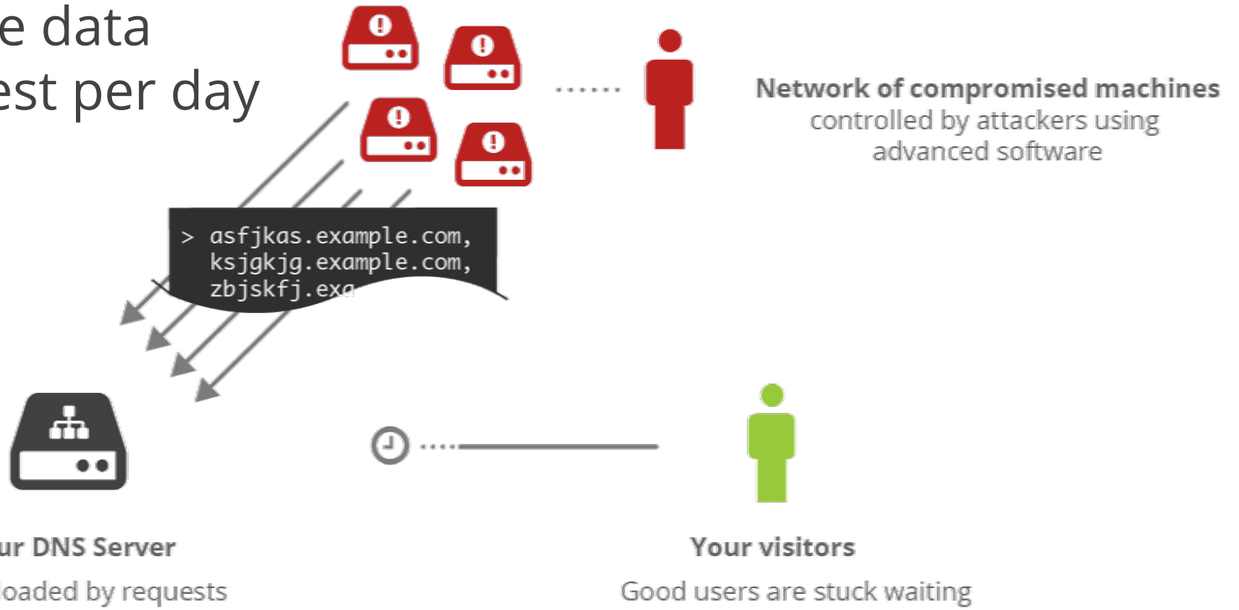
DDoS look-and-feel

DNS Attacks look different

- Layer-7 attacks (hitting the application layer)
- Purpose: exhaust the CPU (vs. bandwidth)

Malicious Payload

- Request sent to exploit vulnerability on server
- Purpose: gain control or release sensitive data
- CloudFlare WAF blocks ~1.2 billion request per day



Deploying 1,000’s of servers, deploying replicated networking

Why run 1,000's and 1,000's of servers?

Geography

- As already stated; spread the load for both content delivery and DDoS processing
- Hence allow us to distribute the attack more effectively
- Allow specific attack sources to be isolated

In-POP load balancing

- Allows us to ensure no one server bears the entire brunt of an attack

Externally presented IP addresses

- One IP can map to 100's (or 1,000's) of servers

This is not just one box!

```
$ host bob.ns.cloudflare.com
bob.ns.cloudflare.com has address 173.245.59.104
bob.ns.cloudflare.com has IPv6 address 2400:cb00:2049:1::adf5:3b68
$
```

DNS - BPF tools + lots and lots of DNS IPs

DNS attacks have a number of unique solutions;

- CloudFlare have many many thousands of DNS servers

```
$ host bob.ns.cloudflare.com
bob.ns.cloudflare.com has address 173.245.59.104
bob.ns.cloudflare.com has IPv6 address 2400:cb00:2049:1::adf5:3b68
$
```

  - Allows us to distribute the attack more effectively
  - Can null route specific DNS server IPs with minimal impact
- BPF (Berkeley Packet Filter) tools
  - High performance pattern matching driven filtering
  - Allows us to filter out DNS attack traffic using far less CPU resource
  - *http://blog.cloudflare.com/introducing-the-bpf-tools/*
  - *https://github.com/cloudflare/bpftools*

ECMP to distribute traffic between servers

Allows us to ensure no one server bears the entire brunt (for traffic coming into a given site) of the attack load aimed at a single IP. (16 servers can more easily mitigate an attack than 1).

All our servers speak BGP to our routing infrastructure, so this is not particularly difficult to implement.

By default, ECMP hashes will be re-calculated every time there is a next-hop change.

- Causes flows to shift between servers
  - TCP sessions reset
- Can solve this with consistent ECMP hashing
  - Available in Junos from 13.3R3 for any trio based chipset
  - Only works for up to 1k unicast prefixes, so struggles to scale

Solarflare cards and OpenOnload

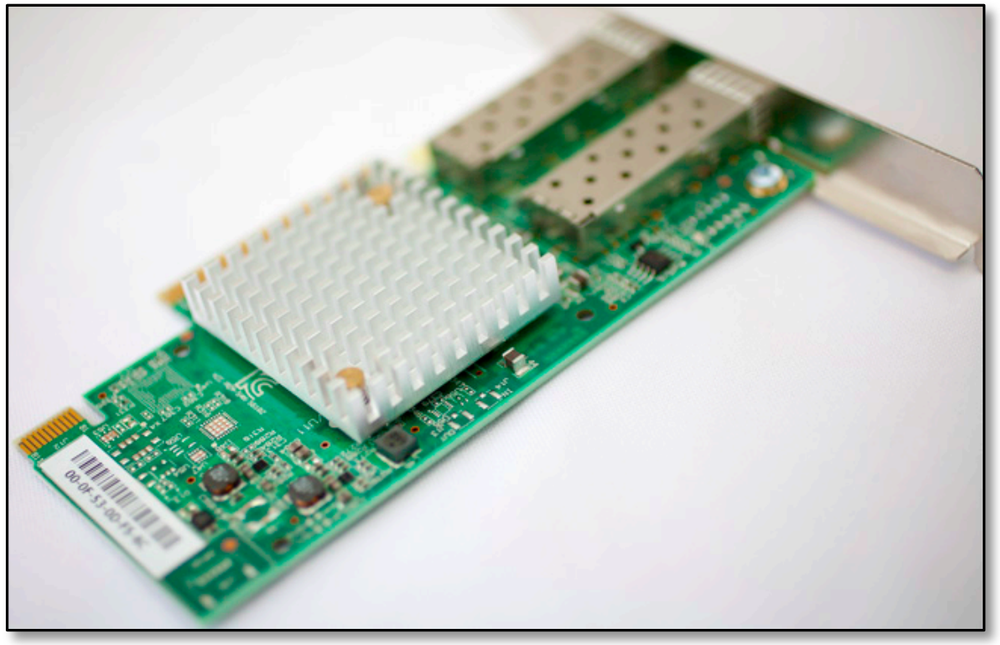
In our latest generation of server hardware we;

- Made the move to 2x10Gbit per server (from 6x1Gbit LAGs)
- Did this with NICs from Solarflare.

SolarFlare NICs have very cool abilities to pre-process traffic on-board before handing to the CPU (OpenOnload).

Can identify certain types of traffic and assign it to cores based on rules pushed in the cards.

Can handle certain requests in userspace without creating CPU interrupts



Cloudflare have been helping the SolarFlare develop this functionality for their cards.

http://blog.cloudflare.com/a-tour-inside-cloudflares-latest-generation-servers/

Hashlimits & “I’m under attack” mode

Enforce “no more than X connection attempts per minute for this hash”, otherwise blacklist

Hash is made up from whatever criterion you want, but for our purposes combo of src + dest IPs

Fairly effective method of easily detecting “ddos-like” traffic.

Trick is preventing false detections:

- Customer with many millions of users released an application update causing the application to regularly perform JSON queries against their application.
- Users behind a CG-NAT appeared as if they were coming from a single IP.
- Triggered enforcement on non-malicious traffic.

“I’m under attack” mode … customer enabled mode that forces users to a challenge page.

Significantly less CPU required to process requests than going through the full process of serving their request.

Mitigation - in the network

Null route and move on

When an attacker targets a website or a service, while they may want to take this website/service down, they target the IP address in order to do this.

First order of business can be to update the DNS A/AAAA record and move on.

If the attacker follows, keep doing this.

Easy to automate, requires an attacker to continually change the attack to follow.

Depends on rDNS service operators honouring our TTLs

FlowSpec (RFC 5575)

Important to understand from the outset that ALL flowspec does is automate the provisioning of a backplane-wide firewall filter on multiple devices. Having said that, **it does this really well**.

Can use most “from” and “then” actions available in Juniper firewall filters in FlowSpec. While Juniper have been an early adopter, other vendors have struggled to get this into their code. Even Juniper has only recently implemented IPv6 support for FlowSpec.

Being able to match “TCP packets from this /24, to this /32, with SYN but no ACK and a packet length of 63 bytes” and “rate-limit to 5Mbit” per edge router is incredibly useful.

Being able to configure this in one place and have it push to the entire network is awesome!

Other scaling methods

Regional enforcement

- Under certain circumstances, it makes sense to enforce regionally
- Regional null routing can also be worthwhile at times

Dealing with attacks on infrastructure IPs

- Multiple hundred gig attack on an anycast IP
- Distribute!

Attacks on Infrastructure - obfuscation of IPs

- Take all your linknet IPs from a /24 that is not advertised on the internet

Peering exchanges should not be reachable on the internet anyway

Scaling the network – it's about capacity

Ultimately, this is all a capacity game.

As you scale up your routers, you may discover that PPS bottlenecks simply move to your transit providers.

Peering and Interconnections at scale

CloudFlare global peering for DDoS protection

AKL-IX	Auckland)
AMS-IX	(Amsterdam)
APE	(Auckland)
BBIX	(Tokyo, Osaka, Singapore)
CABASE-BUE	(Buenos Aires)
DE-CIX	(Frankfurt, New York)
ECIX	(Düsseldorf, Frankfurt)
ESPANIX	(Madrid)
Equinix	(Ashburn, Atlanta, Chicago, Dallas, Hong Kong, Los Angeles, New York, Osaka, Paris, San Jose, Seattle, Singapore, Sydney, Tokyo)
FL-IX	(Miami)
France-IX	(Paris, Marseille)
HKIX	(Hong Kong)
Interlan	(Bucharest)
IX Australia	(Melbourne, Sydney)
JPIX	(Tokyo, Osaka)
JPNAP	(Tokyo, Osaka)
LINX	(London)
LONAP	(London)
MIX-IT	(Milan)
Megaport	(Auckland, Singapore, Sydney)
MyIX	(Kuala Lumpur)
Nap Do Brasil	(São Paulo)
NIX CZ	(Prague)
NL-IX	(Amsterdam)
NOTA	(Miami)
Netnod	(Stockholm)
PIPE	(Melbourne, Sydney)
PLIX	(Warsaw)
PTT-SP	(São Paulo)
Peering.cz	(Prague)
SH-IX	(Fujairah)
SIX	(Seattle)
STHIX	(Stockholm)
Telx	(Atlanta)
TorIX	(Toronto)
VIX	(Vienna)
…	

CloudFlare global peering for DDoS protection

Why do we peer?

> *"In computer networking, peering is a voluntary interconnection of administratively separate Internet networks for the purpose of exchanging traffic between the users of each network."*

- To improve performance (reduce hop count, reduce latency etc.)
- To reduce costs
- To ensure anycast traffic lands locally
- To gain more control over routing
- To gain more control of DDoS traffic

Africa and the AfPIF region

The North/East/West issue …

Moving content into the region at scale

- BGP doesn't understand geography
- BGP doesn't understand latency (an AS-PATH adjacency doesn't show distance)
- BGP is actually complex (at a global scale)

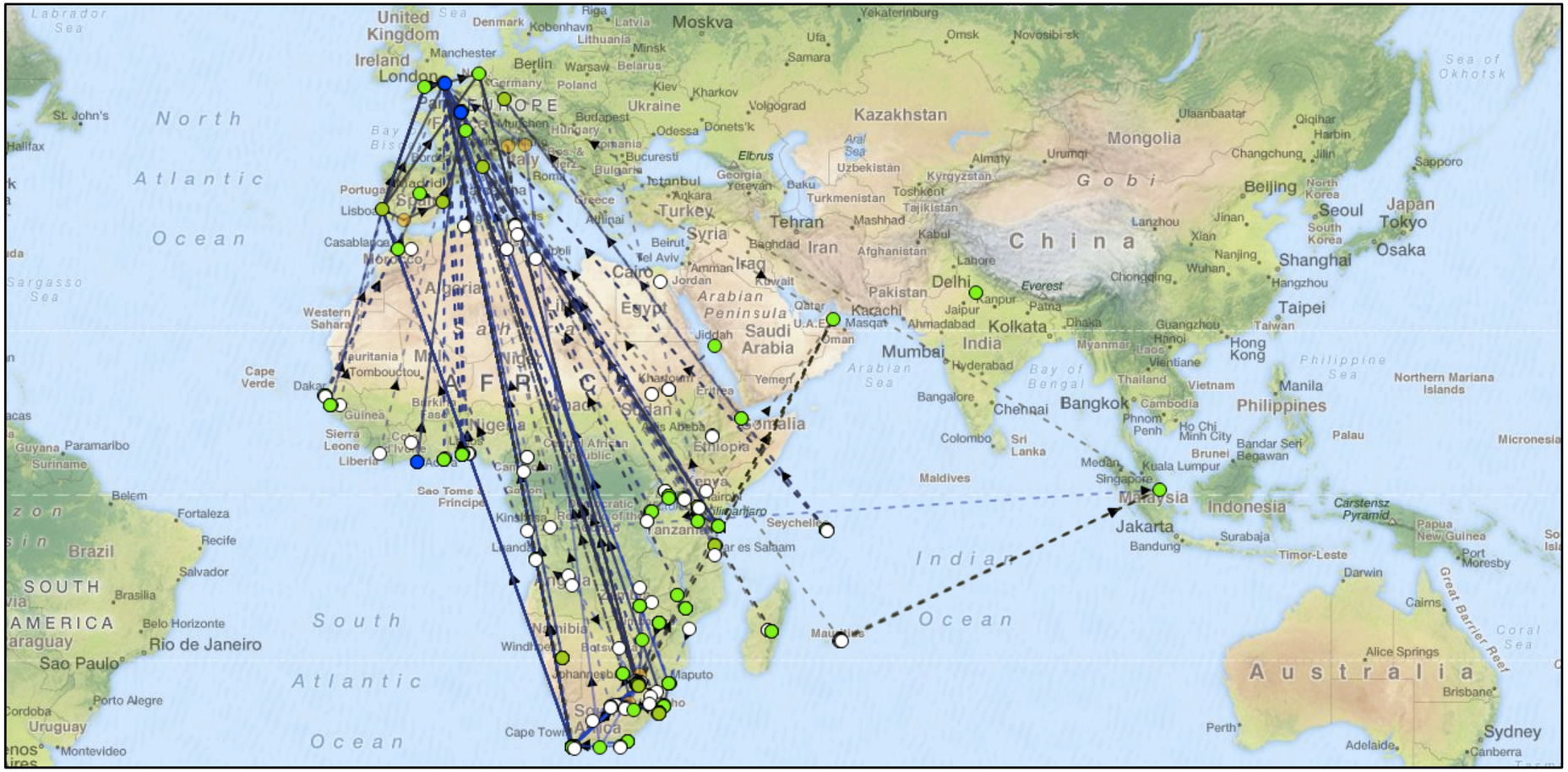
- Asia (Singapore & Hong Kong) or Europe (Marseille, etc) are far away

- The Middle East has some routing from Africa; but it's not the norm.

- Choosing different transits for Asia & Europe causes suboptimal BGP routing

- Peering in Asia & Europe helps; if balanced

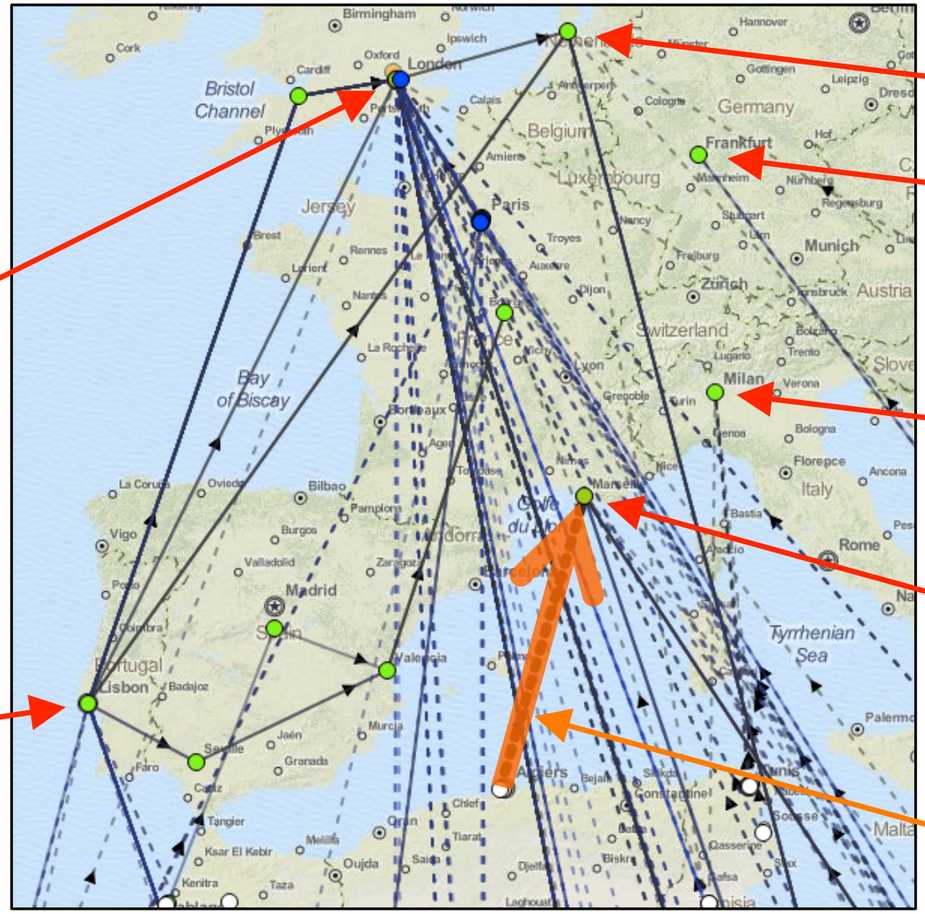
What does connectivity look like?



https://marmot.ripe.net/openipmap/tracemap?msm_ids=2347433&show_suggestions=1&max_probes=300

225 RIPE Atlas probes responding

What does connectivity look like? #2



Summary

Questions?

Thank you!

AS13335

Martin J. Levy, Network Strategy
@martin / @cloudflare
http://www.cloudflare.com/