The invisible impact of network handovers within content delivery

with CDNs and compute moving deeper into the edge,
a few challenges occur which need to be addressed jointly

2022

Trend to the edge: will shrink backbones

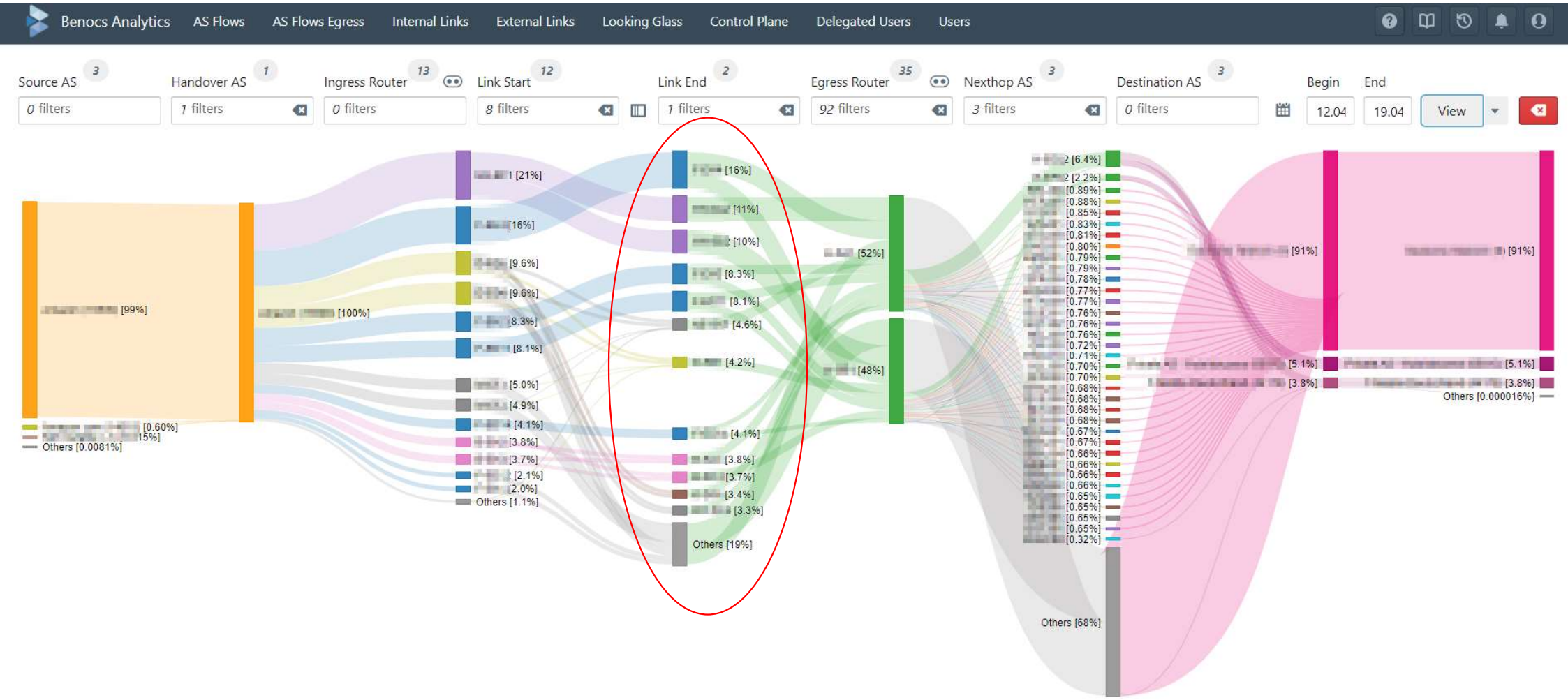
- Good for customer experience
- Good for traffic distribution cost (network)

- More headroom per node needed for traffic spikes
- Failover-concepts are currently poorly aligned

=> Backbones will – at one point - not be able to handle overflow

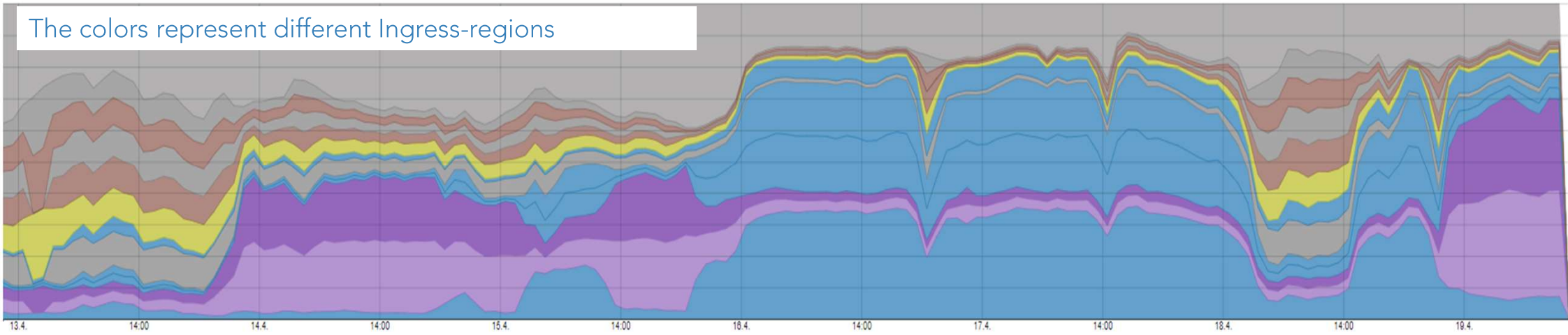
=> Localization and stability will become key requirements

Capacity planning – backbone links into Region A

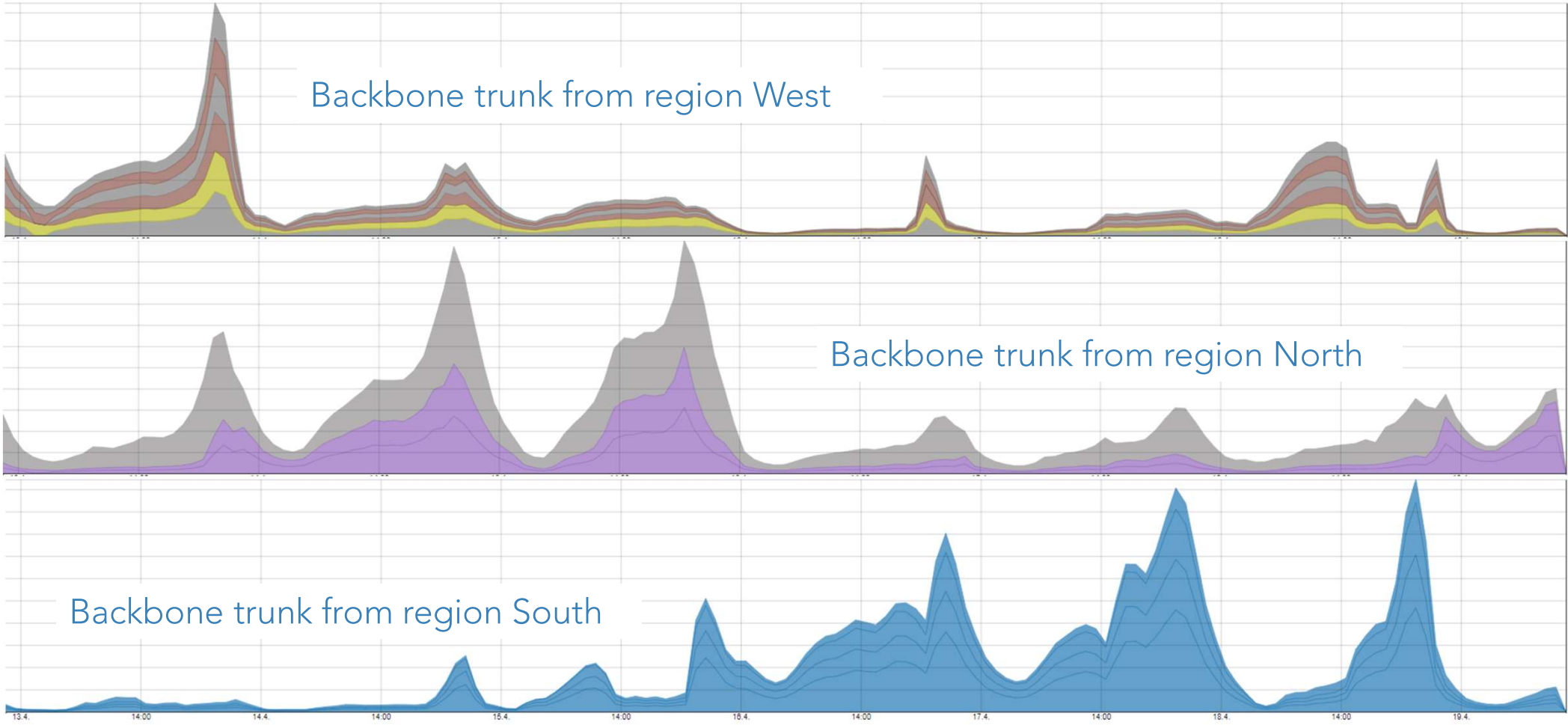


Capacity planning – fail

- 7-day relative traffic profile
- Ingress: 5 CDN ingress locations
- Egress: Region A in an ISP network.



Ingress instability – resulting in 3x capacity needs

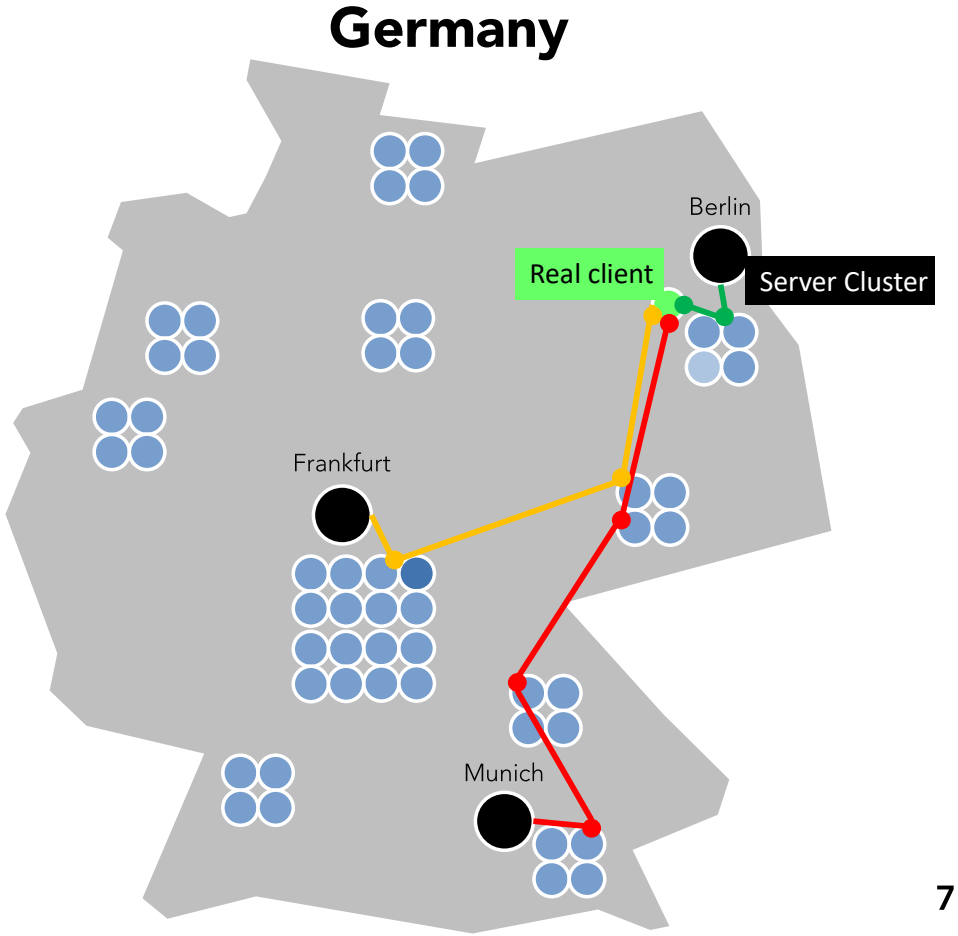


Localization challenges

When you invest in regionalizing servers,
make sure your traffic-regionalization keeps up

Why is localization important?

Ingress (Server-Cluster)	Metric (hopcount, km)
Berlin	1,25
Frankfurt	3,75
Munich	6,89



How CDNs Currently Resolve This:

Workaround	Mechanism	Downside
Anycast	Egress = Ingress	No failover control, requires all content everywhere, no load control, ignores outbound policies
Virtual/regional DNSs	Group all subnets of region in one vDNS	Complex configuration, does not work properly in daily life, failover issues, ignores DoH, SmartTV-DNS
Roundtrip measurement	Send via lowest RTT path	Challenges to break down aggregate prefixes Issues with asymmetrical paths
External Geo-IP Data	Acquire Geo-IP from 3rd parties	Accuracy issues for neighboring locations typically outdated, not topology-aware
Real-time data exchange (ISP-CDN)	Read full topology, status and traffic, export to CDN in real-time	Requires in-network installation But: mapping answers based on the ground truth.

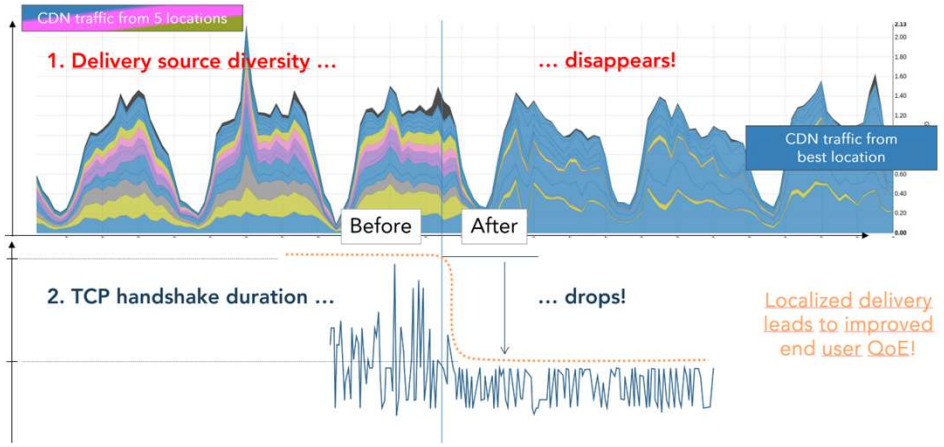
Identifying DNS-resolvers to locate users

- Only few addresses to monitor
- DNS locations fairly stable and easy to communicate
- No ECS/EDNS0 required

- DNS-resolvers get load-balanced and “mis”configured
- Fall-back resolver locations can be far-off
- Solution fails for DoH (8.8.8.8, 1.1.1.1,…) and SmartTV / IoT

Identifying DNS-resolvers – fail 1

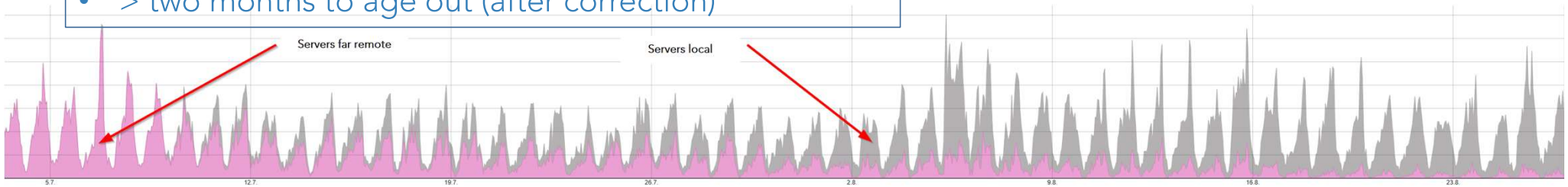
①



- resolvers in round-robin load-balance
- unaware of impacts on CDN's mapping
- once corrected, traffic stabilized

②

- ISP misconfigured remote DNS-resolvers as primary
- CDN was thus using remote servers for delivery.
- > two months to age out (after correction)



Geo-locating users (with internal or external database)

- Databases are broadly available
- Straightforward and seemingly working
- No CDN-ISP engagement needed

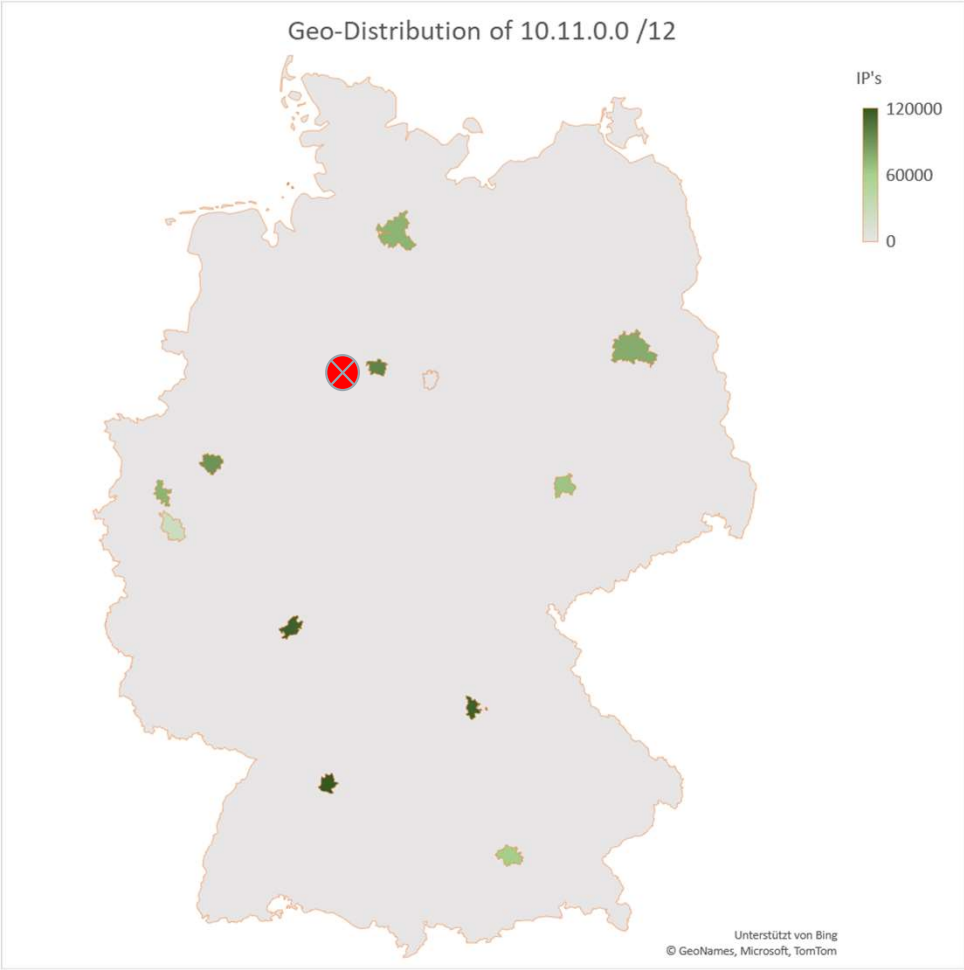
- eBGP aggregates ≠ iBGP subnets (1)
- Update-delay for refarmed subnets (2)
- Geo-distance ≠ Network-distance (3)
- Ignores roundtrip-reality (i.e. outbound path) (3)
- Often inaccurate, no reliable quality check

Geo-locating fail 1: iBGP vs eBGP / aggregated vs specific

eBGP:

10.11.0.0 /12
⇒ 1.048.574 hosts

IP-Geo: ⊗
Braunschweig



iBGP:

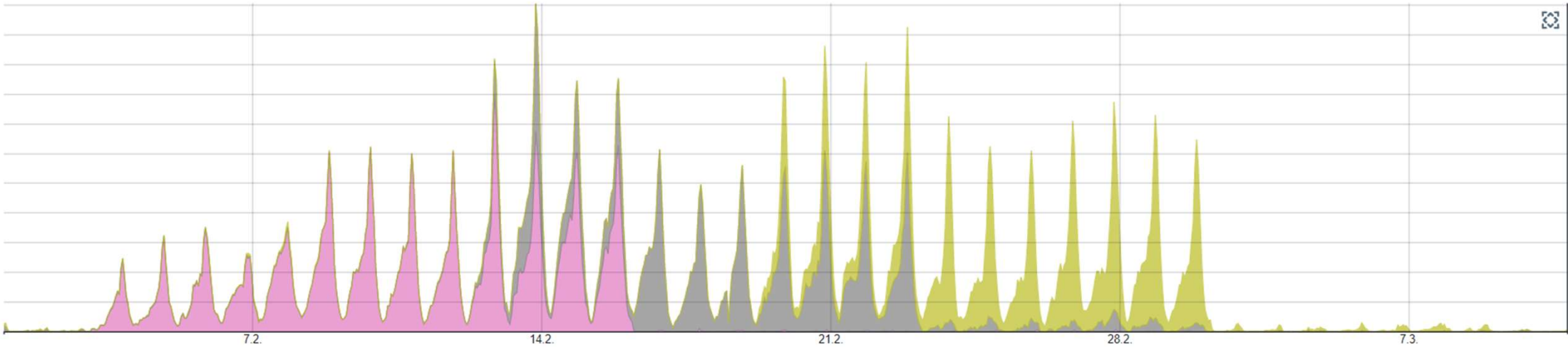
10.11.0.0 /12
⇒ 1.048.574 hosts
⇒ 220 (main) prefixes

Allocation (regions):

Munich
Stuttgart
Nuremberg
Frankfurt
Cologne
Düsseldorf
Dortmund
Leipzig
Hannover
Berlin
Hamburg

Geo-locating users – fail 2: update delay

- Here you see traffic delivered from a remote source to 3 local BNGs.
- After subnets were re-farmed from one region to another, it took the CDN 2 weeks to learn the new geo-location



Geo-locating – fail 3: geo-distance ≠ network distance

Step 1:
Prefixes identified for Heilbronn (HLB):
10.11.16.0/20
10.12.48.0/20
10.13.80.0/20
10.14.144.0/20
…)

Step 2:
Geo distance roundtrip:
MUC-HLB: 414km (+77% vs best choice)
FRA-HLB: 234km

Network distance roundtrip:
MUC-STU-HLB: 462km (+18% vs best choice)
FRA-STU-HLB: 392km

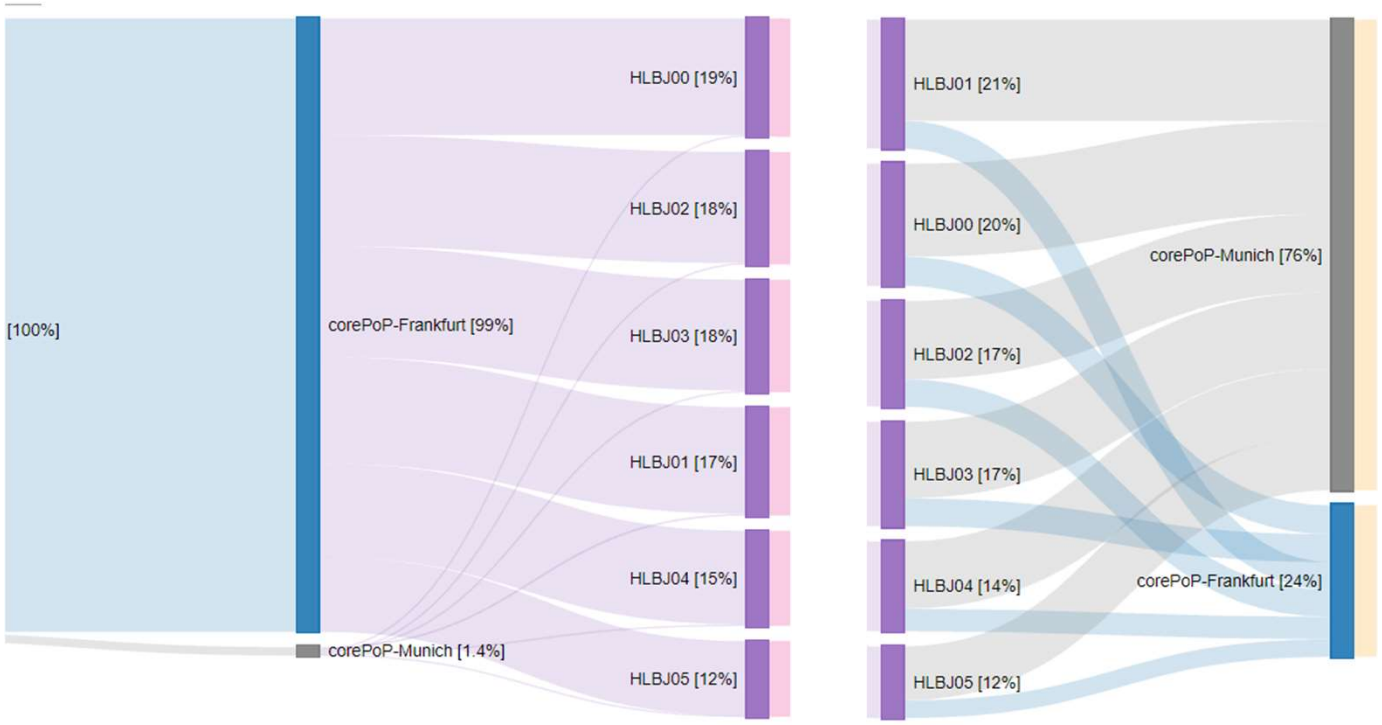
Step 3:
Reality check with actual routing (roundtrip) – (applies for 76% of measured traffic):

MUC-STU-HLB-STU-MUC: 462km
FRA-STU-HLB-STU-MUC-FRA: 735km (+60% vs best choice, 3x of best Geo-choice)

Geo-locating – fail 3: geo-distance ≠ network distance

In (CDN => ISP)

Out (ISP => CDN)



Be aware of asymmetrical paths!

Thanks!
Questions?

Stephan Schroeder
sschroeder@benocs.com