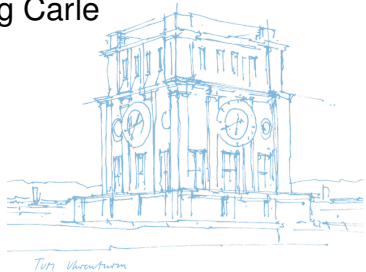


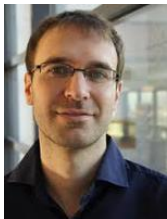
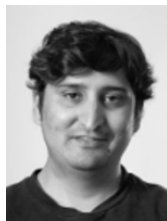
Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists

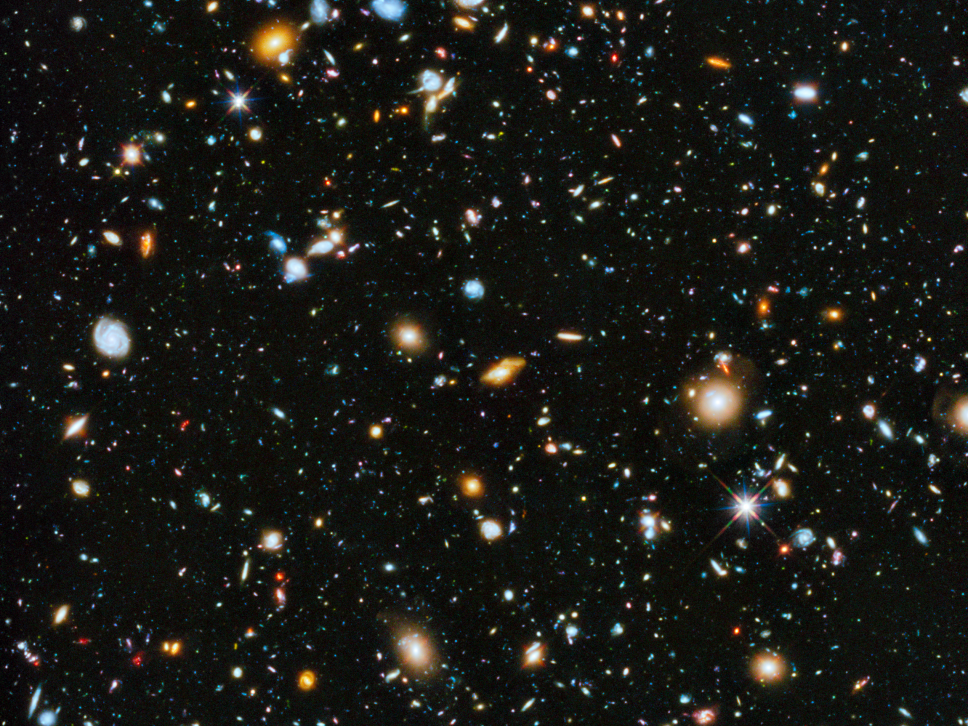
Oliver Gasser, Quirin Scheitle, Paweł Foremski,
Qasim Lone, Maciej Korczyński, Stephen D. Strowes,
Luuk Hendriks, Georg Carle

IMC 2018, Boston



Joint work







THE
EXPANSE



THE
EXPANSE

of the IPv6 Address Space

Previous work on IPv6 address space analysis

- Dhamdhere et al. (2012)
- Czyz et al. (2014)
- Plonka and Berger (2015, 2017)
- Ullrich et al. (2015)
- Gasser et al. (2016)
- Rohrer et al. (2016)
- Foremski et al. (2016)
- Murdock et al. (2017)
- Fiebig et al. (2017, 2018)
- Borgolte et al. (2018)

1. How balanced are different hitlist sources?

1. How balanced are different hitlist sources?
2. Can we identify addressing schemes to find new addresses?

1. How balanced are different hitlist sources?
2. Can we identify addressing schemes to find new addresses?
3. What is the influence of aliased prefixes on IPv6 hitlists?

1. How balanced are different hitlist sources?
2. Can we identify addressing schemes to find new addresses?
3. What is the influence of aliased prefixes on IPv6 hitlists?
4. How does cross-protocol responsiveness in IPv6 differ from IPv4?

1. How balanced are different hitlist sources?
2. Can we identify addressing schemes to find new addresses?
3. What is the influence of aliased prefixes on IPv6 hitlists?
4. How does cross-protocol responsiveness in IPv6 differ from IPv4?
5. Is there a benefit of using more than one address learning tool?

1. How balanced are different hitlist sources?

Hitlist sources

Where can we learn potential IPv6 addresses?

Hitlist sources

Where can we learn potential IPv6 addresses?

- Domainlists
- FDNS
- CT
- AXFR
- Bitnodes
- RIPE Atlas
- Scamper

Hitlist sources

Where can we learn potential IPv6 addresses?

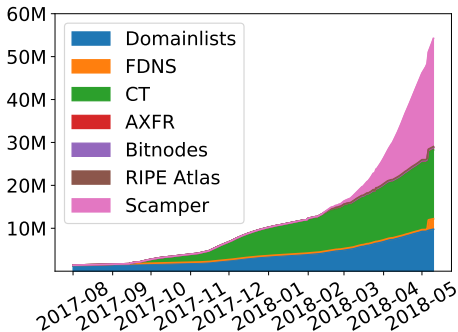


Figure 1: Cumulative runup of IPv6 addresses.

Where can we learn potential IPv6 addresses?

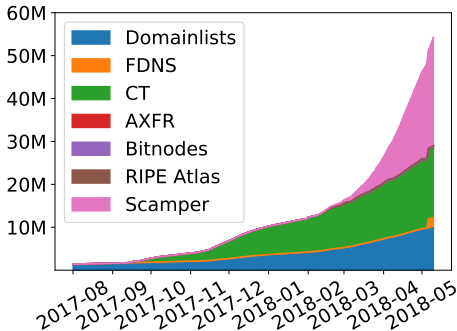


Figure 1: Cumulative runup of IPv6 addresses.

Address distribution

- Many addresses from domainlists, CT, and scamper
- Rapid increase of scamper addresses due to CPE routers

Hitlist sources



How balanced are the addresses from different sources?

Hitlist sources

How balanced are the addresses from different sources?

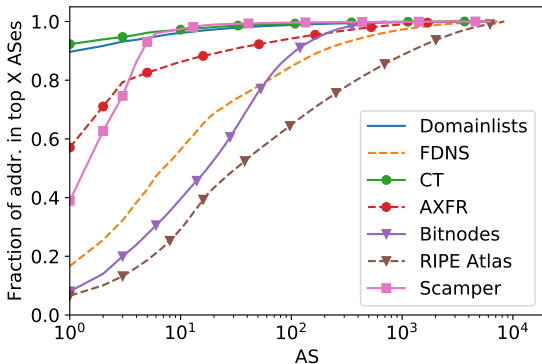


Figure 2: AS distribution for hitlist sources.

How balanced are the addresses from different sources?

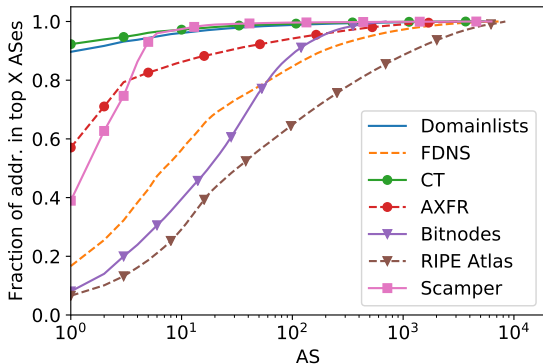


Figure 2: AS distribution for hitlist sources.

Autonomous System distribution

- Unbalanced (CT, domainlists) vs. balanced (RIPE Atlas)

How much of the announced address space do we cover?

Excursion: Visualizing prefixes

Visualizing prefixes using Hilbert space-filling curves

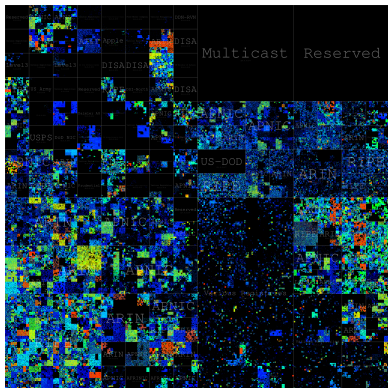


Figure 3: IPv4

Excursion: Visualizing prefixes

Visualizing prefixes using Hilbert space-filling curves

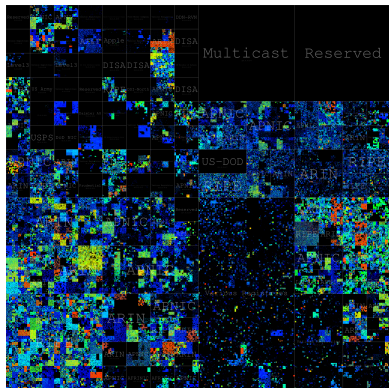


Figure 3: IPv4

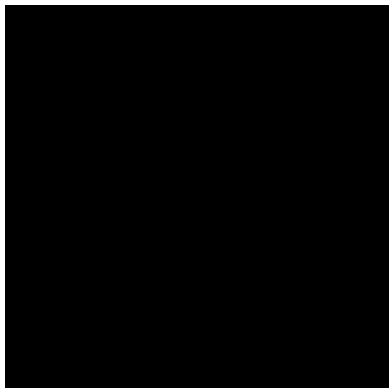


Figure 4: IPv6

Figures by Ben Cartwright-Cox <https://blog.benjojo.co.uk/post/scan-ping-the-internet-hilbert-curve>

How much of the announced address space do we cover?

How much of the announced address space do we cover?

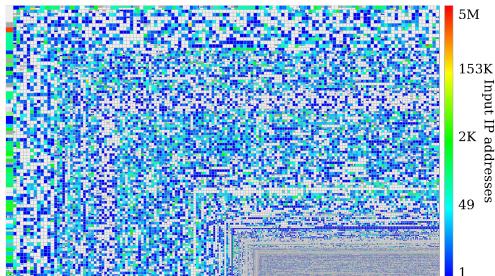


Figure 5: Number of addresses per prefix.

How much of the announced address space do we cover?

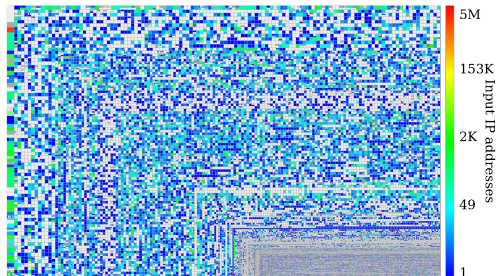


zesplot

- IPv6 prefix visualization tool
- Input: set of IPv6 prefixes
- Each plotted as rectangle
- Prefixes of same AS and size are plotted adjacently
- Color based on metric (e.g. number of addrs. in prefix)

Figure 5: Number of addresses per prefix.

How much of the announced address space do we cover?



zesplot

- IPv6 prefix visualization tool
- Input: set of IPv6 prefixes
- Each plotted as rectangle
- Prefixes of same AS and size are plotted adjacently
- Color based on metric (e.g. number of addrs. in prefix)

Figure 5: Number of addresses per prefix.

BGP prefix distribution

- Good coverage of BGP prefixes: 25.5 k of 51.2 k
- Some prefixes with many addresses

2. Can we identify common addressing schemes in our hitlist?

Entropy clustering



Understand addressing patterns in IPv6 hitlists

Entropy clustering

Understand addressing patterns in IPv6 hitlists

```
2001:0db8:4001:0806:0000:0000:0000:201b
2001:0db8:4003:0c00:0000:0000:0000:00c2
2001:0db8:4004:080f:0000:0000:0000:2014
2001:0db8:4001:0c08:0000:0000:0000:001c
2001:0db8:4002:0803:0000:0000:0000:2009
2001:0db8:4002:0c09:0000:0000:0000:007d
2001:0db8:4009:080d:0000:0000:0000:101b
2001:0db8:400a:0807:0000:0000:0000:2011
2001:0db8:400c:0c04:0000:0000:0000:0056
2001:0db8:400c:0c05:0000:0000:0000:009b
2001:0db8:400e:0c03:0000:0000:0000:00a7
2001:0db8:4012:0806:0000:0000:0000:1003
```

(ignore) **fingerprint!**



```
2001:0db9:0011:00d1:fda4:faa0:0370:7321
2001:0db9:402f:7d00:fdce:da4c:aa23:5ea5
2001:0db9:4134:9700:645c:b3c2:b5bd:ae87
2001:0db9:4134:9700:f47d:cc3b:5956:845f
2001:0db9:4306:9d00:eca1:e02e:13e0:4ca3
2001:0db9:4333:5400:fa32:e4ff:fea0:86dc
2001:0db9:43da:9600:98b2:c969:b41c:ddcb
2001:0db9:43e6:9200:402c:87a9:c25b:76a6
2001:0db9:43e6:9200:455b:da2b:2482:ef42
2001:0db9:43e6:9200:d921:6beb:16f8:41d6
2001:0db9:4400:aa00:24e1:56a6:3253:52d0
2001:0db9:4400:aa00:2cb5:98e4:9b40:61a2
```

ignore **fingerprint!**



Networks have different entropy fingerprints

Understand addressing patterns in IPv6 hitlists

```
2001:0db8:4001:0806:0000:0000:0000:201b
2001:0db8:4003:0c00:0000:0000:0000:00c2
2001:0db8:4004:080f:0000:0000:0000:2014
2001:0db8:4001:0c08:0000:0000:0000:001c
2001:0db8:4002:0803:0000:0000:0000:2009
2001:0db8:4002:0c09:0000:0000:0000:007d
2001:0db8:4009:080d:0000:0000:0000:101b
2001:0db8:400a:0807:0000:0000:0000:2011
2001:0db8:400c:0c04:0000:0000:0000:0056
2001:0db8:400c:0c05:0000:0000:0000:009b
2001:0db8:400e:0c03:0000:0000:0000:00a7
2001:0db8:4012:0806:0000:0000:0000:1003
```

(ignore) **fingerprint!**



```
2001:0db9:0011:00d1:fda4:faa0:0370:7321
2001:0db9:402f:7d00:fdce:da4c:aa23:5ea5
2001:0db9:4134:9700:645c:b3c2:b5bd:ae87
2001:0db9:4134:9700:f47d:cc3b:5956:845f
2001:0db9:4306:9d00:eca1:e02e:13e0:4ca3
2001:0db9:4333:5400:fa32:e4ff:fea0:86dc
2001:0db9:43da:9600:98b2:c969:b41c:ddcb
2001:0db9:43e6:9200:402c:87a9:c25b:76a6
2001:0db9:43e6:9200:455b:da2b:2482:ef42
2001:0db9:43e6:9200:d921:6beb:16f8:41d6
2001:0db9:4400:aa00:24e1:56a6:3253:52d0
2001:0db9:4400:aa00:2cb5:98e4:9b40:61a2
```

ignore **fingerprint!**



Networks have different entropy fingerprints

1. Fingerprint each network

Understand addressing patterns in IPv6 hitlists

```
2001:0db8:4001:0806:0000:0000:0000:201b
2001:0db8:4003:0c00:0000:0000:0000:00c2
2001:0db8:4004:080f:0000:0000:0000:2014
2001:0db8:4001:0c08:0000:0000:0000:001c
2001:0db8:4002:0803:0000:0000:0000:2009
2001:0db8:4002:0c09:0000:0000:0000:007d
2001:0db8:4009:080d:0000:0000:0000:101b
2001:0db8:400a:0807:0000:0000:0000:2011
2001:0db8:400c:0c04:0000:0000:0000:0056
2001:0db8:400c:0c05:0000:0000:0000:009b
2001:0db8:400e:0c03:0000:0000:0000:00a7
2001:0db8:4012:0806:0000:0000:0000:1003
```

(ignore) **fingerprint!**



```
2001:0db9:0011:00d1:fda4:faa0:0370:7321
2001:0db9:402f:7d00:fdce:da4c:aa23:5ea5
2001:0db9:4134:9700:645c:b3c2:b5bd:ae87
2001:0db9:4134:9700:f47d:cc3b:5956:845f
2001:0db9:4306:9d00:eca1:e02e:13e0:4ca3
2001:0db9:4333:5400:fa32:e4ff:fea0:86dc
2001:0db9:43da:9600:98b2:c969:b41c:ddcb
2001:0db9:43e6:9200:402c:87a9:c25b:76a6
2001:0db9:43e6:9200:455b:da2b:2482:ef42
2001:0db9:43e6:9200:d921:6beb:16f8:41d6
2001:0db9:4400:aa00:24e1:56a6:3253:52d0
2001:0db9:4400:aa00:2cb5:98e4:9b40:61a2
```

ignore **fingerprint!**



Networks have different entropy fingerprints

1. Fingerprint each network
2. Feed to k-means clustering

Understand addressing patterns in IPv6 hitlists

```
2001:0db8:4001:0806:0000:0000:0000:201b
2001:0db8:4003:0c00:0000:0000:0000:00c2
2001:0db8:4004:080f:0000:0000:0000:2014
2001:0db8:4001:0c08:0000:0000:0000:001c
2001:0db8:4002:0803:0000:0000:0000:2009
2001:0db8:4002:0c09:0000:0000:0000:007d
2001:0db8:4009:080d:0000:0000:0000:101b
2001:0db8:400a:0807:0000:0000:0000:2011
2001:0db8:400c:0c04:0000:0000:0000:0056
2001:0db8:400c:0c05:0000:0000:0000:009b
2001:0db8:400e:0c03:0000:0000:0000:00a7
2001:0db8:4012:0806:0000:0000:0000:1003
```

(ignore) **fingerprint!**



```
2001:0db9:0011:00d1:fda4:faa0:0370:7321
2001:0db9:402f:7d00:fdce:da4c:aa23:5ea5
2001:0db9:4134:9700:645c:b3c2:b5bd:ae87
2001:0db9:4134:9700:f47d:cc3b:5956:845f
2001:0db9:4306:9d00:eca1:e02e:13e0:4ca3
2001:0db9:4333:5400:fa32:e4ff:fea0:86dc
2001:0db9:43da:9600:98b2:c969:b41c:ddcb
2001:0db9:43e6:9200:402c:87a9:c25b:76a6
2001:0db9:43e6:9200:455b:da2b:2482:ef42
2001:0db9:43e6:9200:d921:6beb:16f8:41d6
2001:0db9:4400:aa00:24e1:56a6:3253:52d0
2001:0db9:4400:aa00:2cb5:98e4:9b40:61a2
```

ignore **fingerprint!**



Networks have different entropy fingerprints

1. Fingerprint each network
2. Feed to k-means clustering
3. Plot median fingerprints and cluster popularity

IPv6 interface identifiers (IIDs)

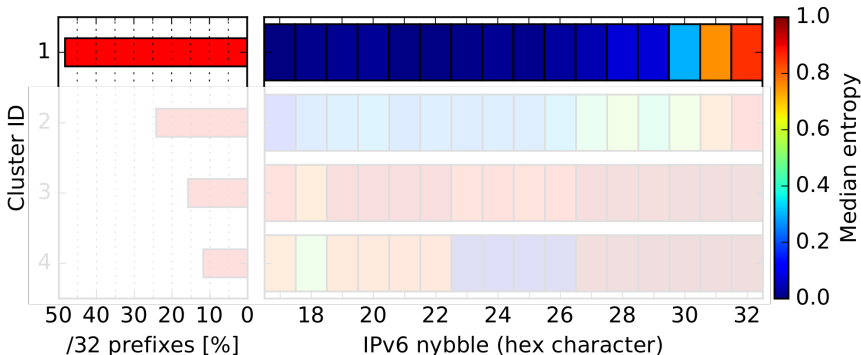


Figure 6: Hitlist addressing schemes for IIDs.

IPv6 interface identifiers (IIDs)

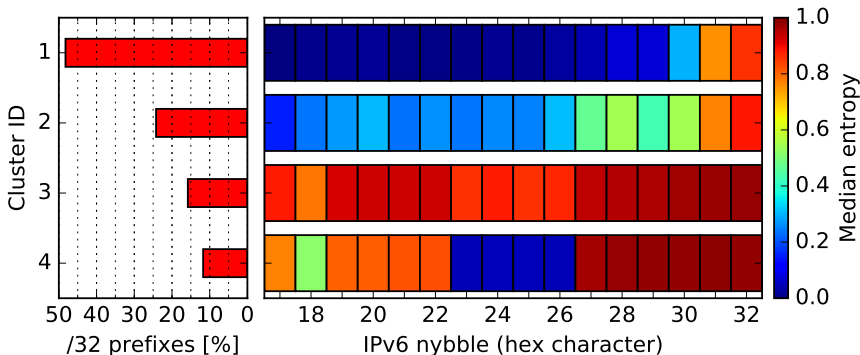


Figure 6: Hitlist addressing schemes for IIDs.

- The IPv6 networks we cover employ predictable IIDs
- Also visible: privacy extensions, modified EUI-64 (ff:fe)

Entropy clustering

Full IPv6 fingerprints

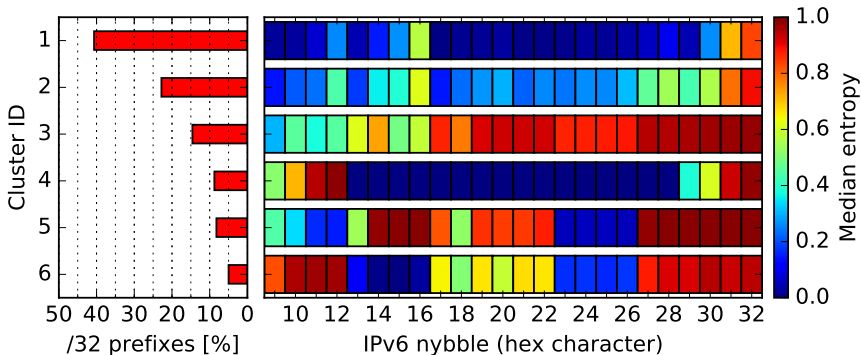


Figure 7: Hitlist addressing schemes for full addresses.

Full IPv6 fingerprints

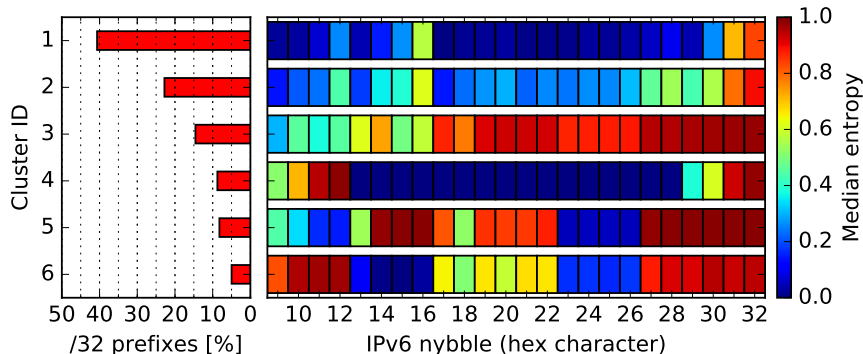


Figure 7: Hitlist addressing schemes for full addresses.

- Just a handful of schemes on the Internet
- Addresses largely predictable

3. What is the influence of aliased prefixes on IPv6 hitlists?

Taxonomy:

- Alias: another address of the same host
- Aliased prefix: whole prefix bound to the same host
- Bias: some hosts overrepresented due to aliased prefixes

Taxonomy:

- Alias: another address of the same host
- Aliased prefix: whole prefix bound to the same host
- Bias: some hosts overrepresented due to aliased prefixes

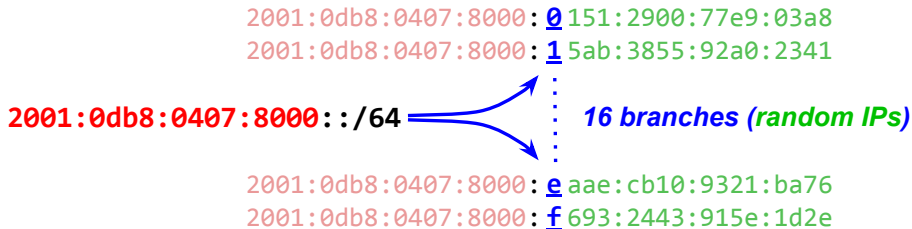


Figure 8: Multi-level aliased prefix detection using pseudo-random probing.

Results

Detecting aliased prefixes

Results

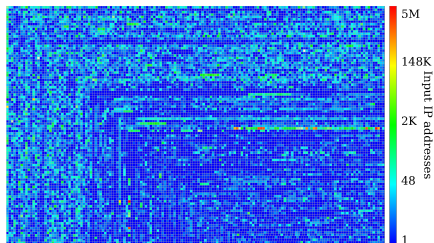


Figure 9: All prefixes covered by hitlist.

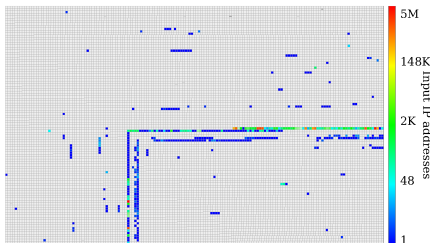


Figure 10: Aliased prefixes.

Results

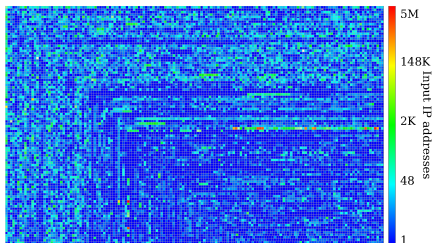


Figure 9: All prefixes covered by hitlist.

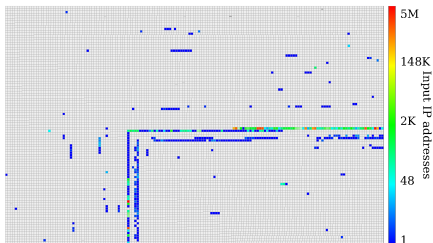


Figure 10: Aliased prefixes.

- Only 3.2 % of prefixes are aliased
- But 46.6 % of addresses are in aliased prefixes
- Validated using fingerprinting (iTTL, TCP opts, timestamps)

4. How does cross-protocol responsiveness in IPv6 differ from IPv4?

- If address responds on protocol X, how likely is it to respond on protocol Y?
- Goal: Identify relevant addresses for specific measurements

Cross-protocol responsiveness

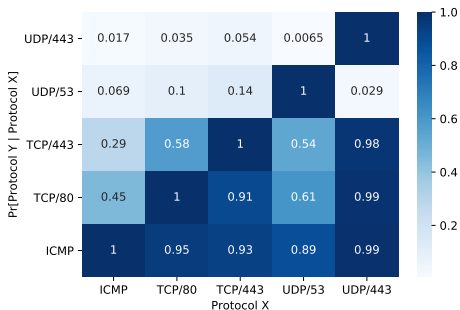


Figure 11: Cross-protocol responsiveness between services.

Cross-protocol responsiveness

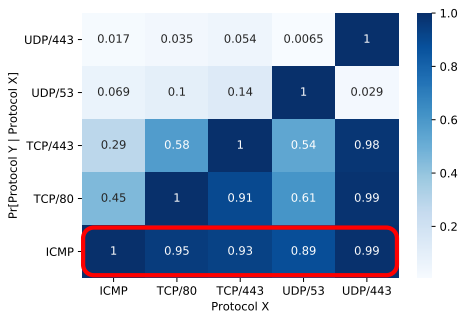


Figure 11: Cross-protocol responsiveness between services.

- If responsive to any of the probes → at least 89% probability it will answer to ICMPv6

Cross-protocol responsiveness

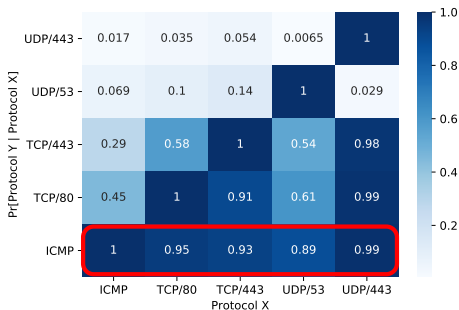


Figure 11: Cross-protocol responsiveness between services.

- If responsive to any of the probes → at least 89% probability it will answer to ICMPv6 vs. 73% in IPv4

Cross-protocol responsiveness

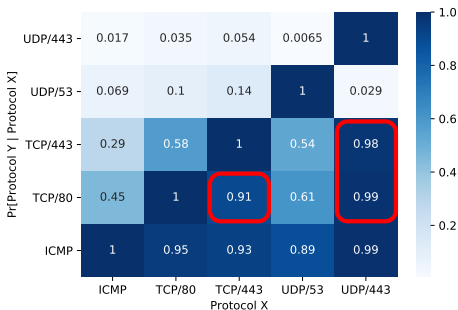


Figure 11: Cross-protocol responsiveness between services.

- If responsive to any of the probes → at least 89% probability it will answer to ICMPv6 vs. 73% in IPv4
- Web protocols: QUIC → HTTPS and HTTP, HTTPS → HTTP; but not the other way around

5. Is there a benefit of using more than one address learning tool?

Techniques to learn new addresses

- Entropy/IP: Generate new addresses by leveraging entropy of seed addresses
 - Similar approach to grouping addresses based on their structure as shown earlier

Techniques to learn new addresses

- Entropy/IP: Generate new addresses by leveraging entropy of seed addresses
 - Similar approach to grouping addresses based on their structure as shown earlier
- 6Gen: Generate new addresses in dense address regions
 - If we see addresses
 - 2001:0db8:0407:8000::4
 - 2001:0db8:0407:8000::5
 - 2001:0db8:0407:8000::8
 - Likely other valid addresses
 - 2001:0db8:0407:8000::6
 - 2001:0db8:0407:8000::7

Learning new addresses

How well do Entropy/IP and 6Gen perform?

- Input: All previously found IPv6 addresses
- Responsiveness: 278 k (of 118 M) and 489 k (of 129 M)

Learning new addresses

How well do Entropy/IP and 6Gen perform?

- Input: All previously found IPv6 addresses
- Responsiveness: 278 k (of 118 M) and 489 k (of 129 M)
- Overlap of only 675 k generated addresses
- 10x higher response rate for overlapping addresses

Learning new addresses

How well do Entropy/IP and 6Gen perform?

- Input: All previously found IPv6 addresses
- Responsiveness: 278 k (of 118 M) and 489 k (of 129 M)
- Overlap of only 675 k generated addresses
- 10x higher response rate for overlapping addresses

Table 1: Top 5 responsive protocol combinations for Entropy/IP and 6Gen.

ICMPv6	TCP/80	TCP/443	UDP/53	UDP/443	Entropy/IP	6Gen
✓	✗	✗	✗	✗	41.1 %	66.8 %
✓	✓	✓	✗	✗	12.3 %	9.2 %
✗	✗	✗	✓	✗	23.1 %	7.3 %
✓	✓	✗	✗	✗	3.4 %	4.9 %
✓	✓	✓	✗	✓	6.1 %	3.2 %

- Different host populations

Reproducibility

- We publish data, code, and analysis scripts
- DOI: 10.14459/2018mp1452739

Software and tools published on GitHub

- ZMapv6
- zesplot
- Entropy clustering
- New Entropy/IP generator
- Entropy/IP open-sourced (thanks to Akamai)

A one-off analysis is all well and good, but what if I need an up-to-date IPv6 hitlist for my research starting next month?

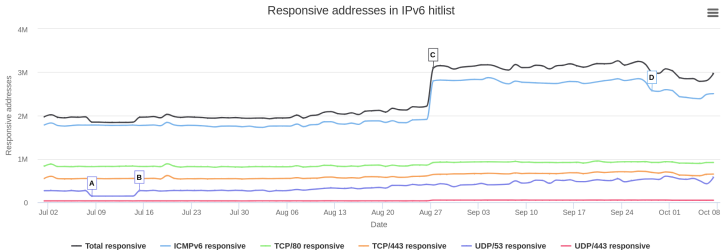
A one-off analysis is all well and good, but what if I need an up-to-date IPv6 hitlist for my research starting next month?

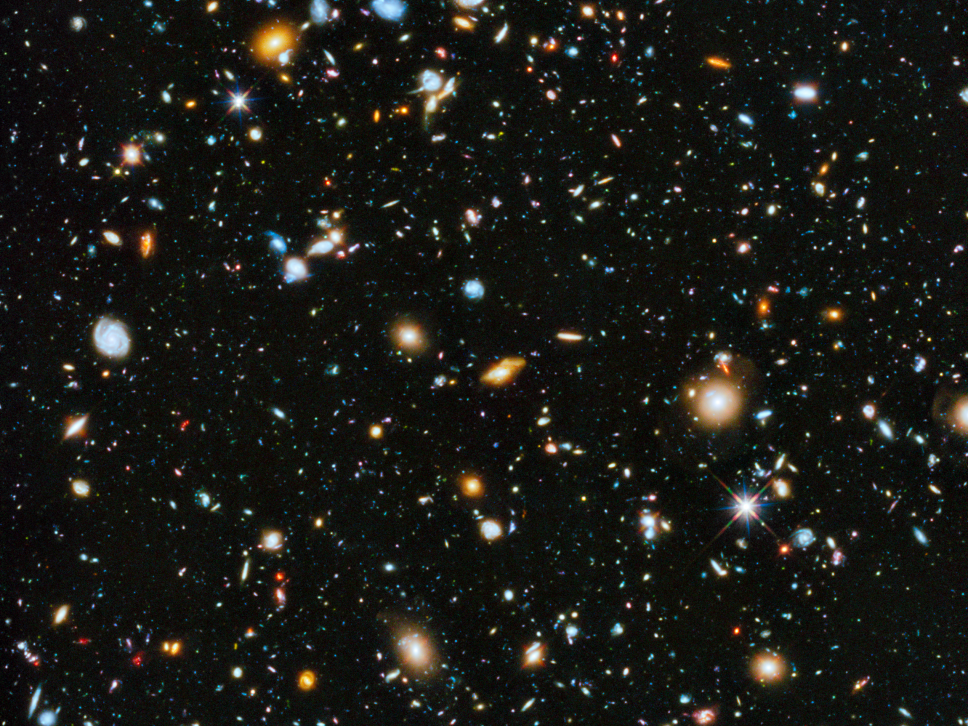
ipv6hitlist.github.io

A one-off analysis is all well and good, but what if I need an up-to-date IPv6 hitlist for my research starting next month?

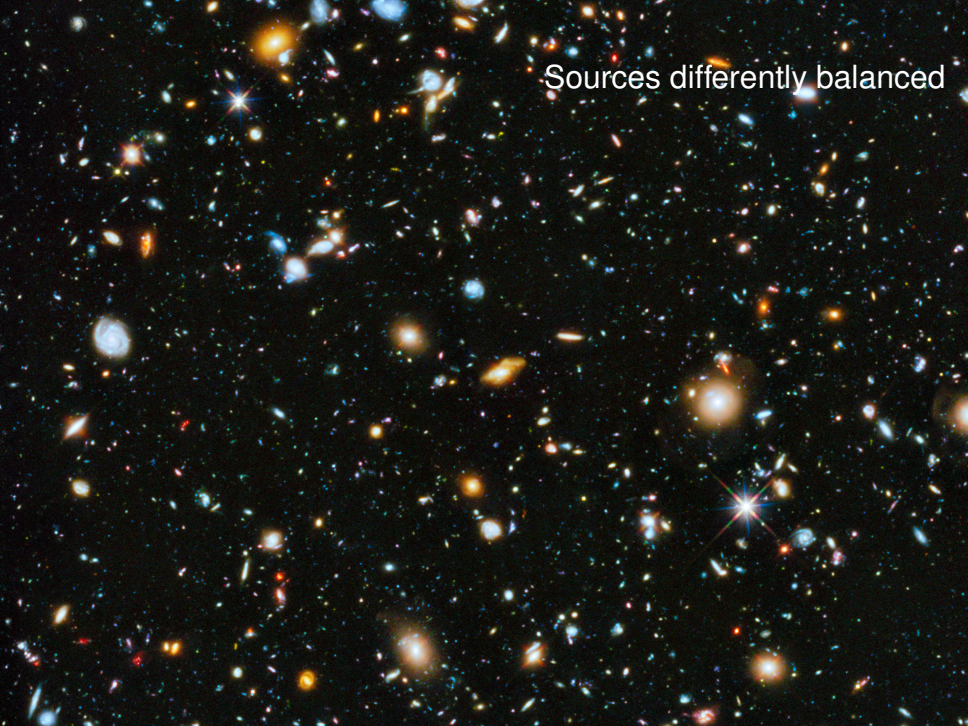
ipv6hitlist.github.io

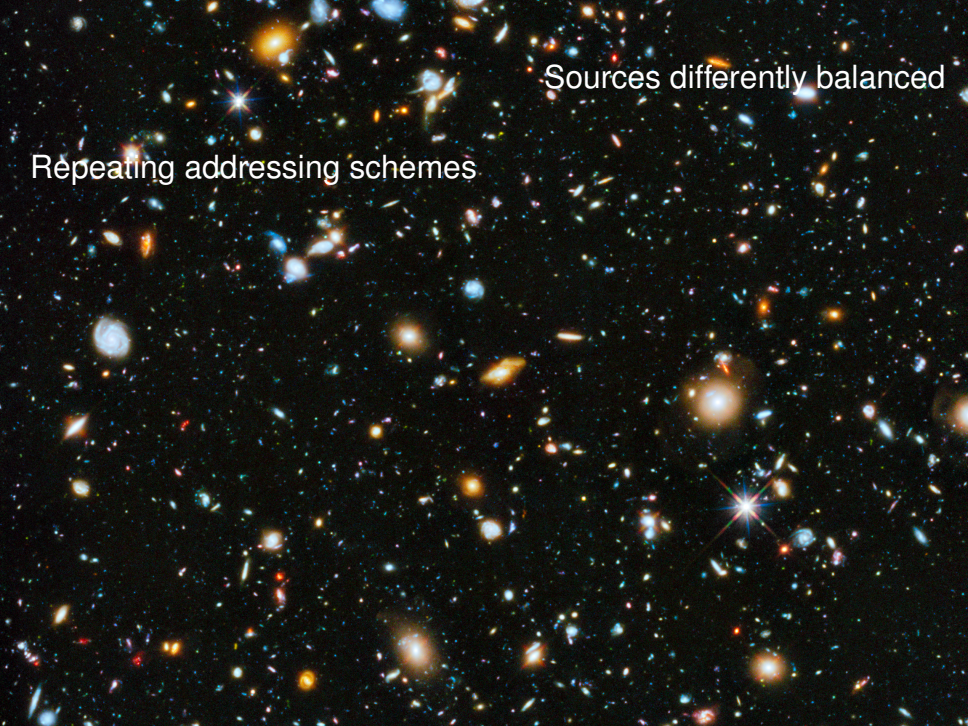
- Daily IPv6 hitlists and aliased prefixes available for download
- Interactive zesplots
- Continuously updated graphs





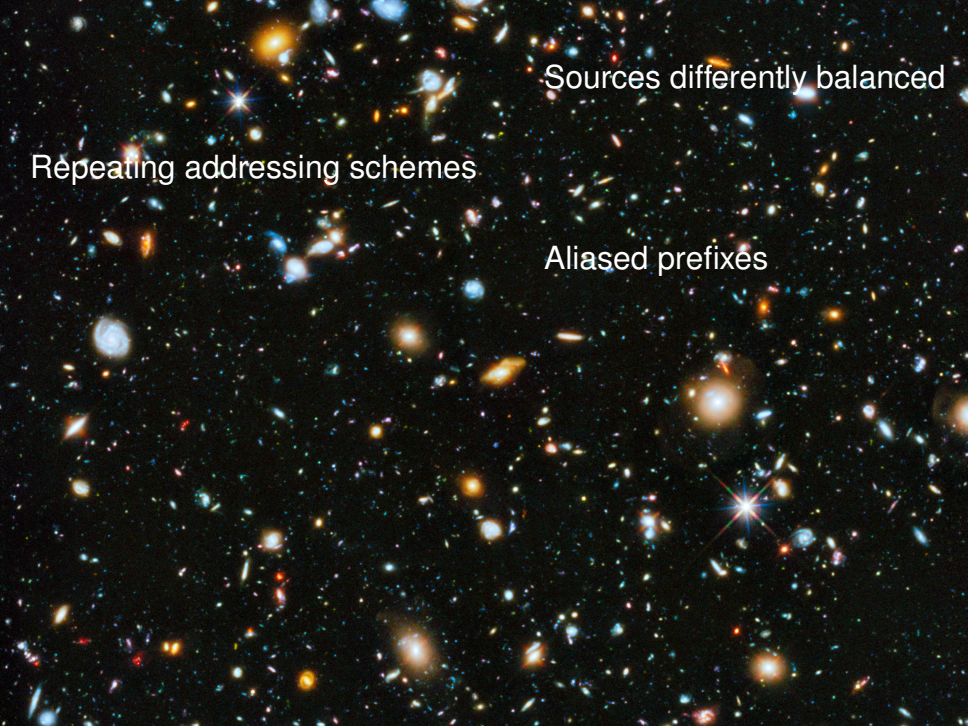
Sources differently balanced





Sources differently balanced

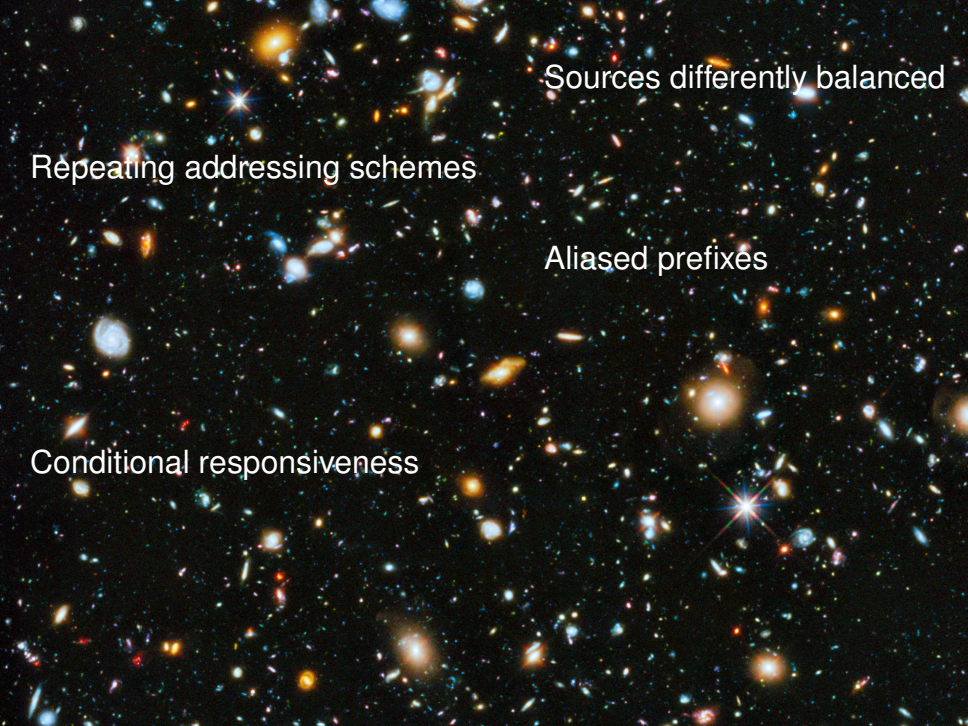
Repeating addressing schemes



Sources differently balanced

Repeating addressing schemes

Aliased prefixes



Sources differently balanced

Repeating addressing schemes

Aliased prefixes

Conditional responsiveness



Repeating addressing schemes

Sources differently balanced

Aliased prefixes

Conditional responsiveness

Learning unknowns

Sources differently balanced

Repeating addressing schemes

Aliased prefixes

ipv6hitlist.github.io

Conditional responsiveness

Learning unknowns

Table 2: Comparison to four previous works.

Previous work	#publ.	#pfx.	#ASes	#priv.	Cts	Prob.	APD
Gasser et al.	2.7 M	5.8 k	8.6 k	149 M	✓	✓	✗
Foremski et al.	620 k	<100 ¹	<100 ¹	3.5 G	✓	✓	✗
Fiebig et al.	2.8 M	n/a ²	n/a	0	✓	✗	✗
Murdock et al.	1.0 M	2.8 k	2.4 k	0	✓	✓	○
This work	55.1 M	25.5 k	10.9 k	0	✓	✓	✓

1: 15 networks, with few prefixes and ASes. 2: 582 k /64s. 3: Responsive addresses.

Table 3: Overview of hitlist sources, as of 2018-05-11.

Name	Public	Nature	IPs	new IPs	#ASes	#PFXes	Top AS1	Top AS2	Top AS3
DL: Domain Lists ¹	Yes	Servers	9.8 M	9.8 M	6.1 k	10.3 k	89.7%★	2.0%●	1.5%■
FDNS: Rapid7 FDNS	Yes	Servers	3.3 M	2.5 M	7.7 k	13.6 k	16.7%★	8.9%▲	6.7%⚡
CT: Domains from CT logs ²	Yes	Servers	18.5 M	16.2 M	5.3 k	8.7 k	92.3%★	1.6%⚡	0.8%★
AXFR: AXFR&TLDR	Yes	Mixed	0.7 M	0.5 M	3.2 k	4.7 k	57.0%★	14.0%●	8.3%■
BIT: Bitnodes	Yes	Mixed	31 k	27 k	695	1.4 k	8.0%★	6.0%■	6.0%▲
RA: RIPE Atlas ³	Yes	Routers	0.2 M	0.2 M	8.4 k	19.1 k	6.6%⚡	3.5%★	3.1%⚡
Scamper	–	Routers	26.0 M	25.9 M	6.3 k	9.8 k	38.9%★	23.8%●	12.0%■
Total			58.5 M	55.1 M	10.9 k	25.5 k	45.4%★	18.4%★	11.5%●

1: Zone Files, Toplists, Blacklists (partially with NDA); 2: Excluding DNS names already included in Domain Lists; 3: Traceroute and ipmap data

★Amazon, ●Host Europe, ■Cloudflare, ▲Linode, ⚡DTAG, ★Proxad, ●Hetzner, ■Comcast, ▲Swisscom, ⚡Google, ★Antel, ●Versatel, ■BIHNET

rDNS as a data source

rDNS data provided by Fiebig et al.

- Is it a useful addition to the hitlist?

11.7 M addresses from IPv6 rDNS

- 11.1 M new addresses → small overlap
- Similar prefix distribution → no additional bias introduced

Active measurements

- Higher ICMPv6 response rate: 10 % vs. 6 %
- Lower HTTP(S) response rate: 2 % (1 %) vs. 3 % (2 %)
- Mostly servers: Few EUI-64 mapped addresses and privacy extensions

rDNS is a good addition to the IPv6 hitlist

- Do aliased prefixes indeed belong to the same host?
- Advanced fingerprinting (initial TTL, TCP options, TCP time-stamp linearity)

- Do aliased prefixes indeed belong to the same host?
- Advanced fingerprinting (initial TTL, TCP options, TCP time-stamp linearity)

Results for 20.7 k /64 prefixes detected as aliased

Test	Σ Incs.	Σ Cons.
iTTL	6	20,686
Optionstext	110	20,581
WScale	215	19,515
MSS	1175	19,513
WSize	1186	19,506
Timestamps	n/a	13,202

- Result confidence depends on the test
- Majority is strongly consistent, few inconsistencies
- Indicates that prefixes determined as aliased are indeed bound to same host

Address responsiveness

Longitudinal responsiveness

- How many responsive addresses in 14 day period?

Address responsiveness

Longitudinal responsiveness

- How many responsive addresses in 14 day period?

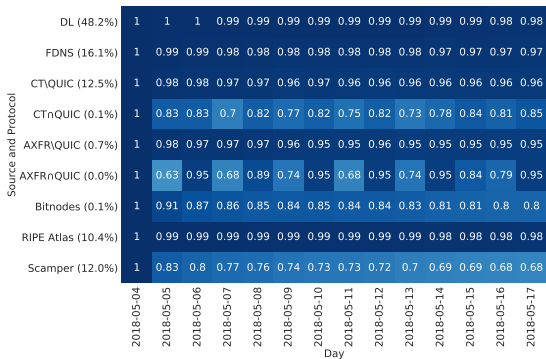


Figure 12: Responsiveness over time, by hitlist source and probed protocol.

Address responsiveness

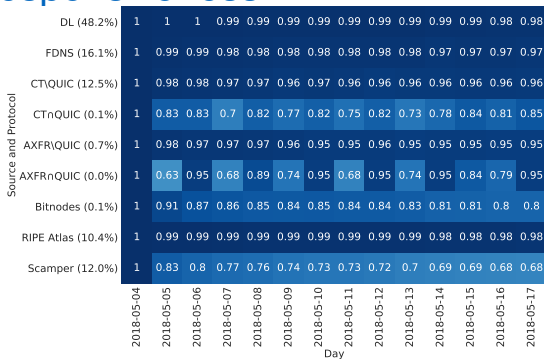


Figure 13: Responsiveness over time, by hitlist source and probed protocol.

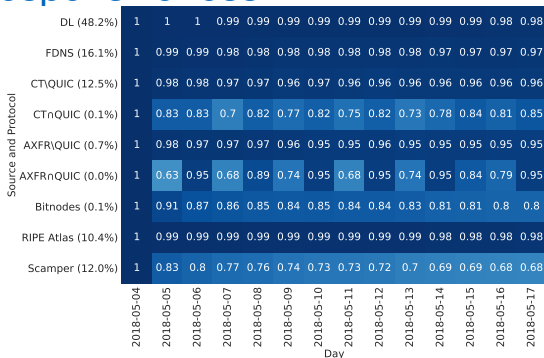


Figure 13: Responsiveness over time, by hitlist source and probed protocol.

- Domainlists, FDNS, RIPE Atlas answer consistently
- CT & AXFR have stable response rate; except for QUIC
- Client or CPE sources (Bitnodes, Scamper) lose 20 % and 32 % of responding hosts

Address responsiveness

Scanning results: UDP/53 responsive

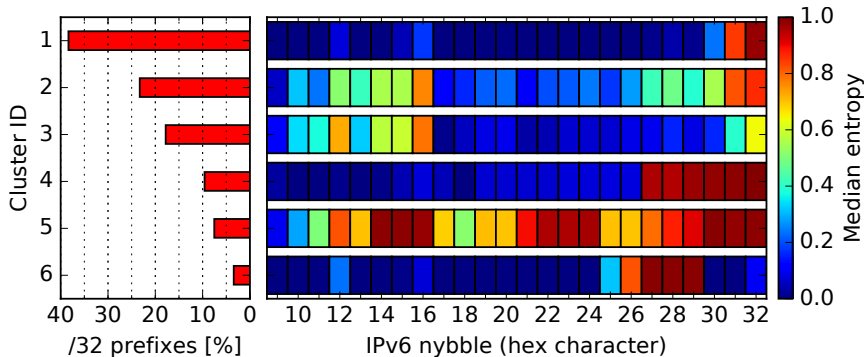


Figure 14: Addressing schemes for UDP/53 responsive addresses.

Address responsiveness

Scanning results: UDP/53 responsive

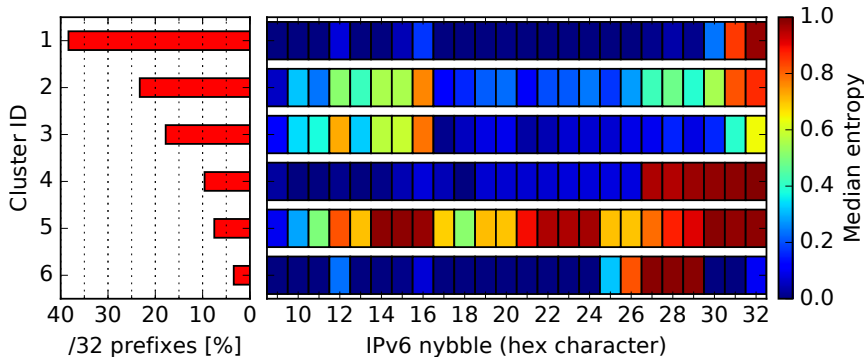


Figure 14: Addressing schemes for UDP/53 responsive addresses.

- Easy to predict
- Probabilistic scanning for DNS servers possible

Client addresses

- Difficult to come by in regular hitlist sources
- Approach: Use crowdsourcing to gather IPv6 client addresses
 - Amazon Mechanical Turk
 - Prolific Academic

Table 4: Client distribution in crowdsourcing study.

#	IPv4	IPv6	ASN ₄	ASN ₆	#CC ₄	#CC ₆
Mturk	5,707	1,787	842	73	93	22
ProA	1,176	245	272	48	33	21
Unique	6,862	2,032	983	92	98	29

Responsiveness

- 352 (17.3 %) respond to at least one probe → ICMPv6 filtering
 - 7 remain responsive over measurement period → lower stability than servers and routers
 - Measurements to clients need to be performed swiftly as clients disappear or change address