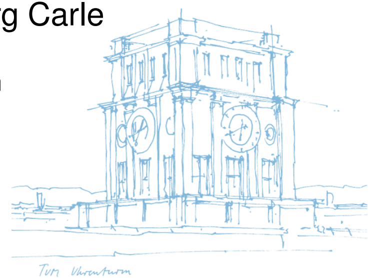
Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

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

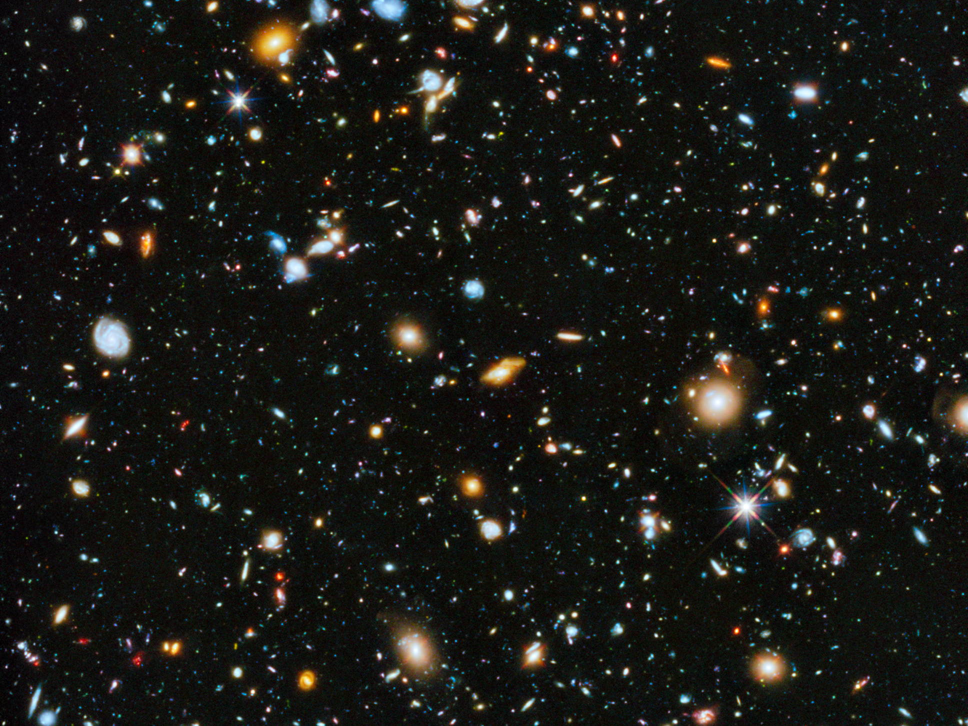


Joint work








THE
EXPANSE

THE

EXPANSE

of the IPv6 Address Space

Previous work

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)

Open questions

1. How balanced are different hitlist sources?

Open questions

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

Open questions

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?

Open questions

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?

Open questions

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?

Hitlist sources

Where can we learn potential IPv6 addresses?

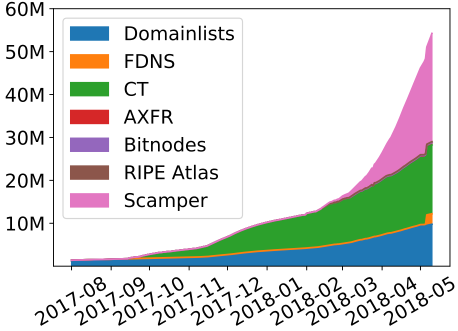


Figure 1: Cumulative runup of IPv6 addresses.

Hitlist sources

Where can we learn potential IPv6 addresses?

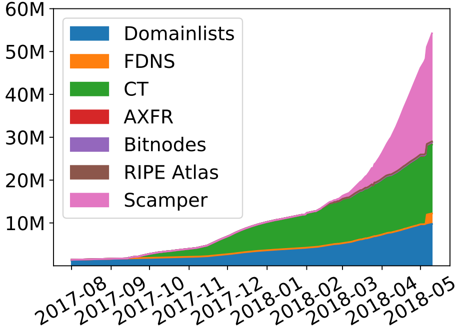


Figure 1: Cumulative runup of IPv6 addresses.

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?

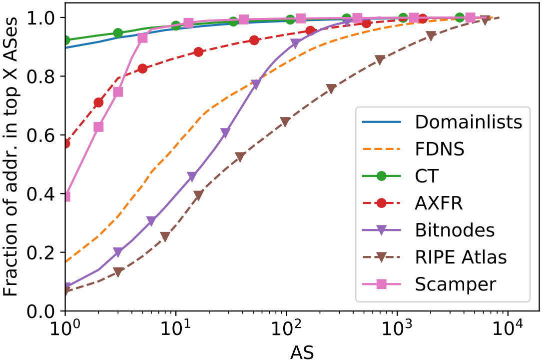


Figure 2: AS distribution for hitlist sources.

Hitlist sources

How balanced are the addresses from different sources?

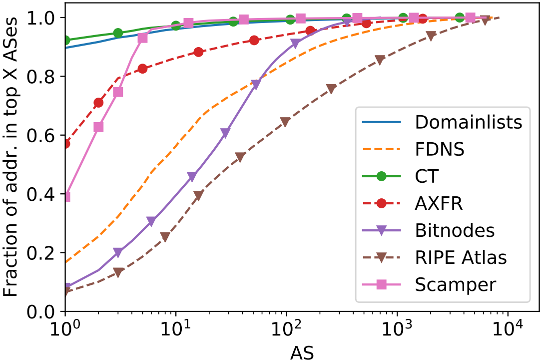


Figure 2: AS distribution for hitlist sources.

Autonomous System distribution

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

Hitlist sources

How much of the announced address space do we cover?

Excursion: Visualizing prefixes

Visualizing prefixes using Hilbert space-filling curves

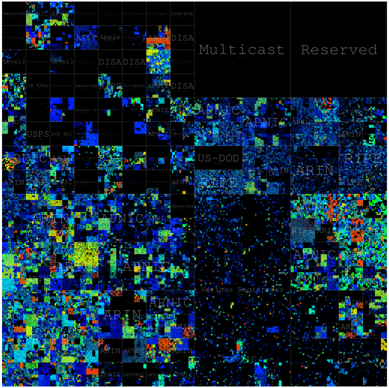


Figure 3: IPv4

Excursion: Visualizing prefixes

Visualizing prefixes using Hilbert space-filling curves

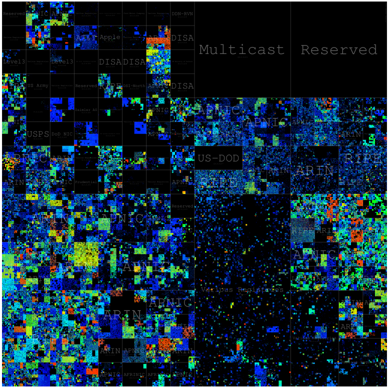


Figure 3: IPv4

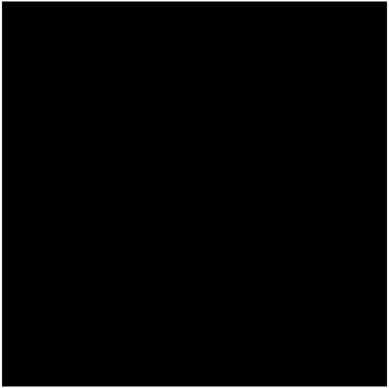


Figure 4: IPv6

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

Hitlist sources

How much of the announced address space do we cover?

Hitlist sources

How much of the announced address space do we cover?

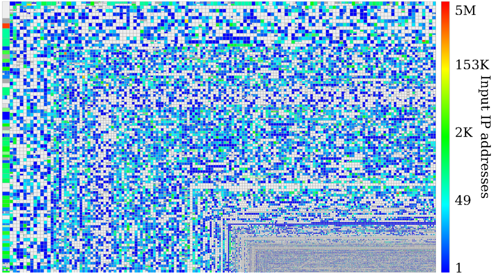


Figure 5: Number of addresses per prefix.

Hitlist sources

How much of the announced address space do we cover?

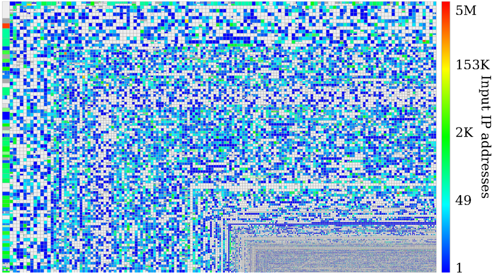


Figure 5: Number of addresses per prefix.

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)

Hitlist sources

How much of the announced address space do we cover?

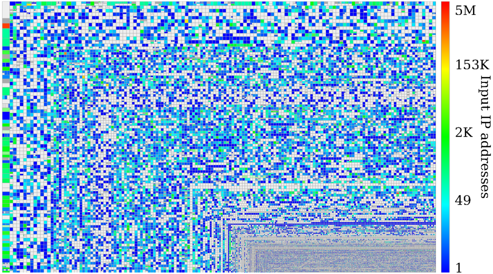


Figure 5: Number of addresses per prefix.

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)

BGP prefix distribution

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

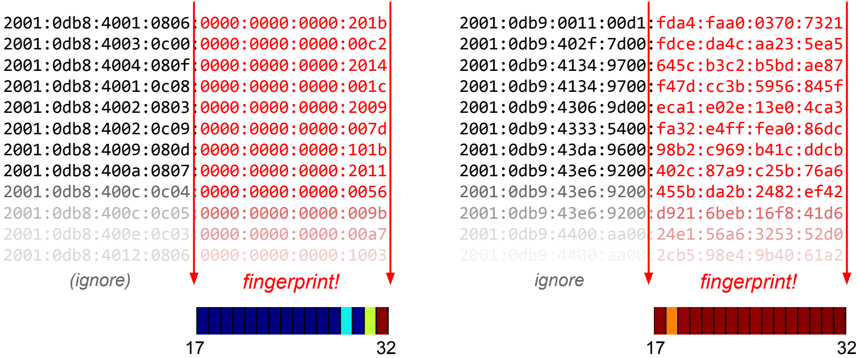
2. Can we identify common addressing schemes in our hitlist?

Entropy clustering

Understand addressing patterns in IPv6 hitlists

Entropy clustering

Understand addressing patterns in IPv6 hitlists



Networks have different entropy fingerprints

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

1. Fingerprint each network

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
```

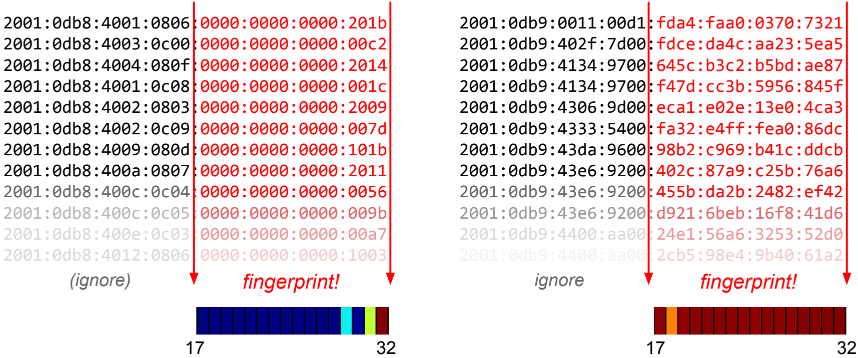
ignore *fingerprint!*

Networks have different entropy fingerprints

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

Entropy clustering

Understand addressing patterns in IPv6 hitlists



Networks have different entropy fingerprints

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

Entropy clustering

IPv6 interface identifiers (IIDs)

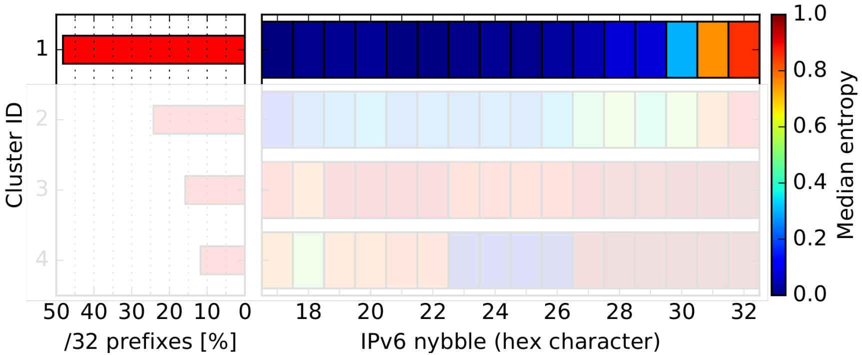


Figure 6: Hitlist addressing schemes for IIDs.

Entropy clustering

IPv6 interface identifiers (IIDs)

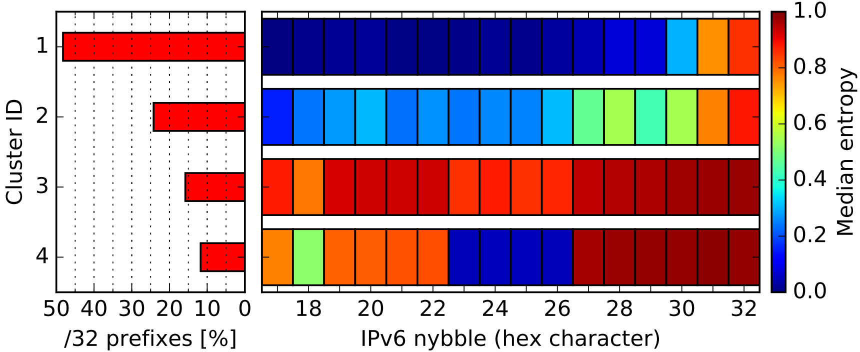


Figure 6: Hitlist addressing schemes for IIDs.

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

Entropy clustering

Full IPv6 fingerprints

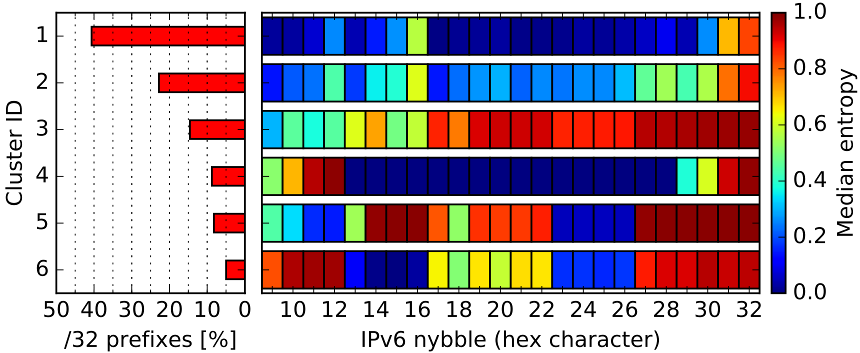


Figure 7: Hitlist addressing schemes for full addresses.

Entropy clustering

Full IPv6 fingerprints

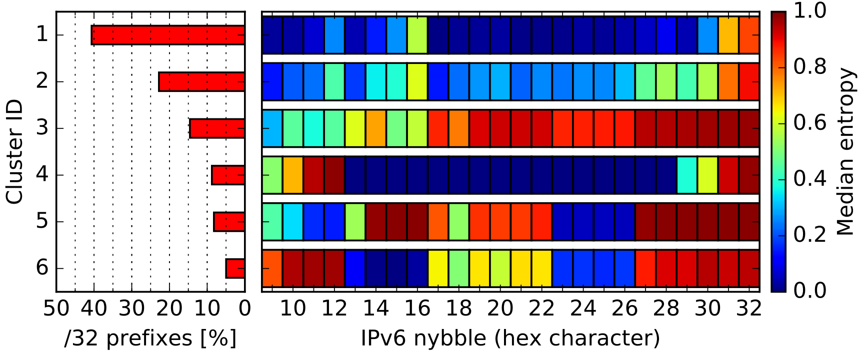


Figure 7: Hitlist addressing schemes for full addresses.

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

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

Detecting 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

Detecting 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

```
                      2001:0db8:0407:8000:0151:2900:77e9:03a8
                      2001:0db8:0407:8000:15ab:3855:92a0:2341
                                          :
2001:0db8:0407:8000::/64                  :   16 branches (random IPs)
                                          :
                      2001:0db8:0407:8000:eaae:cb10:9321:ba76
                      2001:0db8:0407:8000:f693:2443:915e:1d2e
```

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

Detecting aliased prefixes

Results

Detecting aliased prefixes

Results

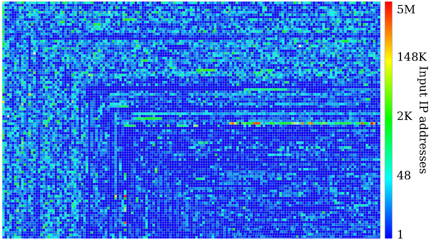


Figure 9: All prefixes covered by hitlist.

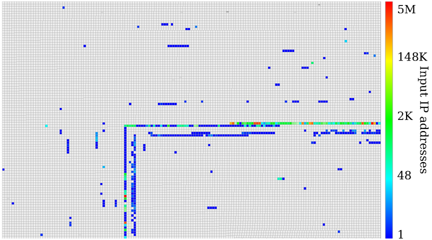


Figure 10: Aliased prefixes.

Detecting aliased prefixes

Results

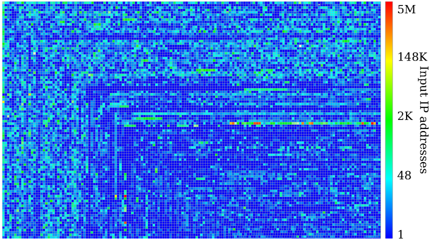


Figure 9: All prefixes covered by hitlist.

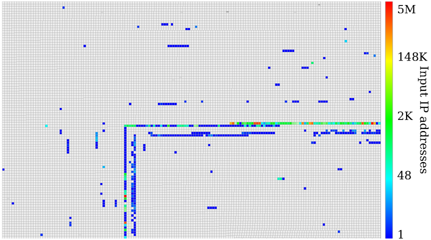


Figure 10: Aliased prefixes.

- 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?

Cross-protocol responsiveness

- 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

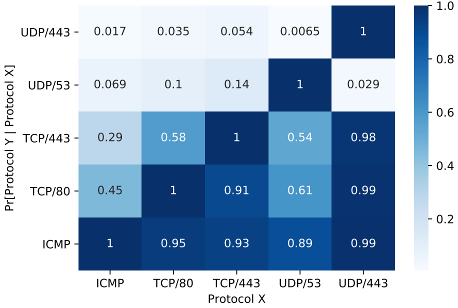


Figure 11: Cross-protocol responsiveness between services.

Cross-protocol responsiveness

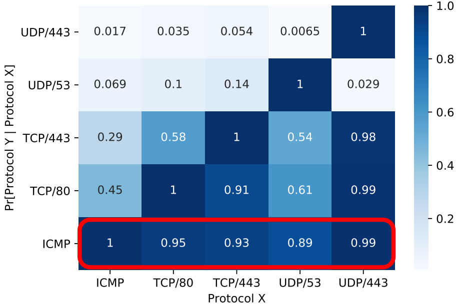


Figure 11: Cross-protocol responsiveness between services.

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

Cross-protocol responsiveness

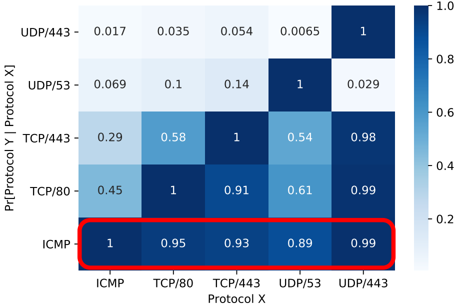


Figure 11: Cross-protocol responsiveness between services.

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

Cross-protocol responsiveness

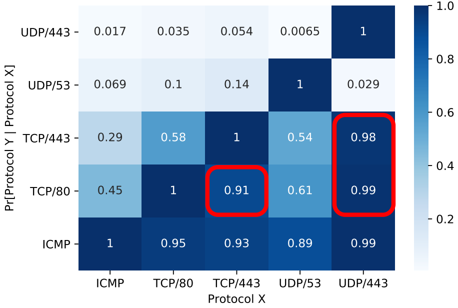


Figure 11: Cross-protocol responsiveness between services.

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

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

Learning new addresses

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

Learning new addresses

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

Community contributions

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)

IPv6 Hitlist Service

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?

IPv6 Hitlist Service

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?

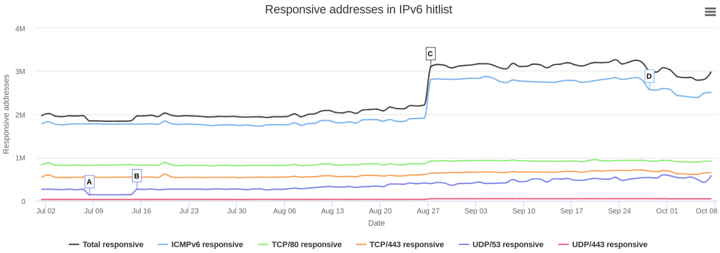
ipv6hitlist.github.io

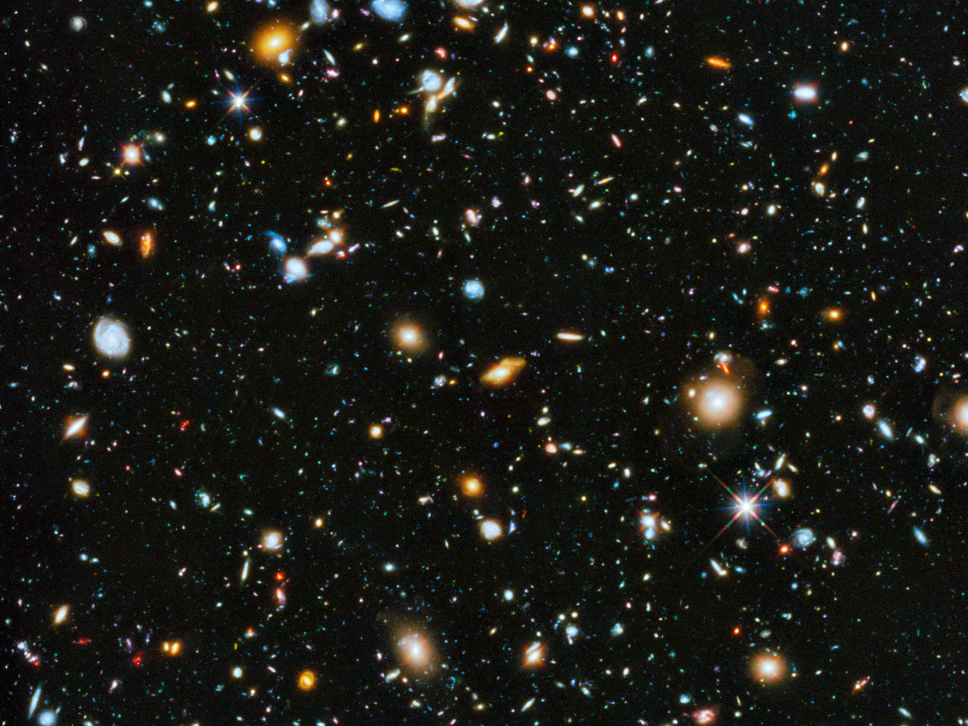
IPv6 Hitlist Service

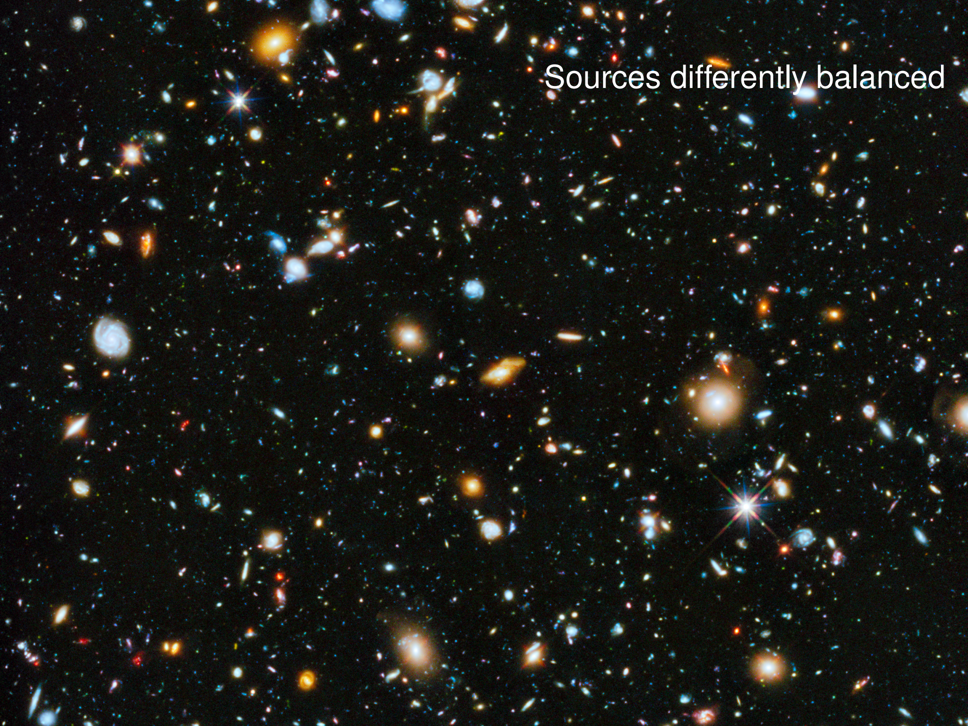
A one-off analysis is all well and good, but what if I need an up-to-date IPv6 hitlist for my research starting next month?

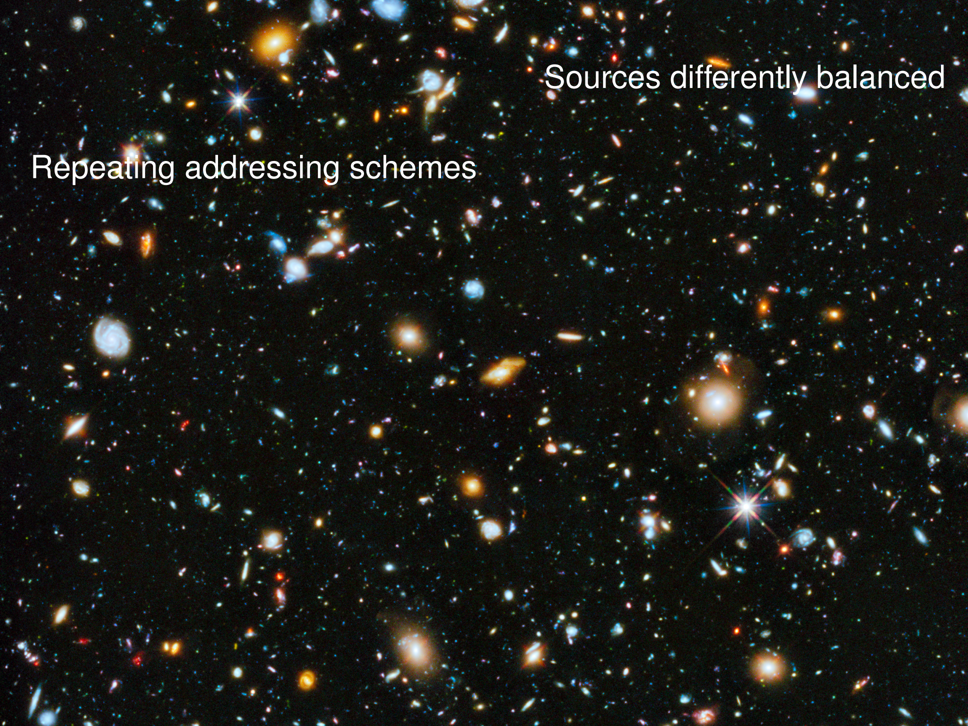
ipv6hitlist.github.io

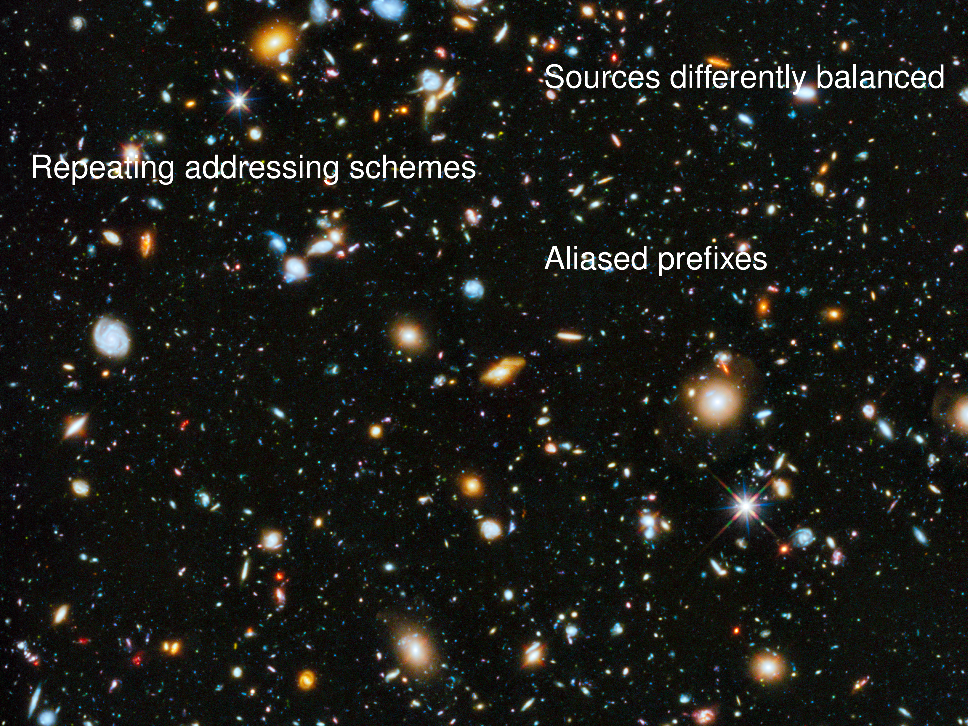
- Daily IPv6 hitlists and aliased prefixes available for download
- Interactive zesplots
- Continuously updated graphs

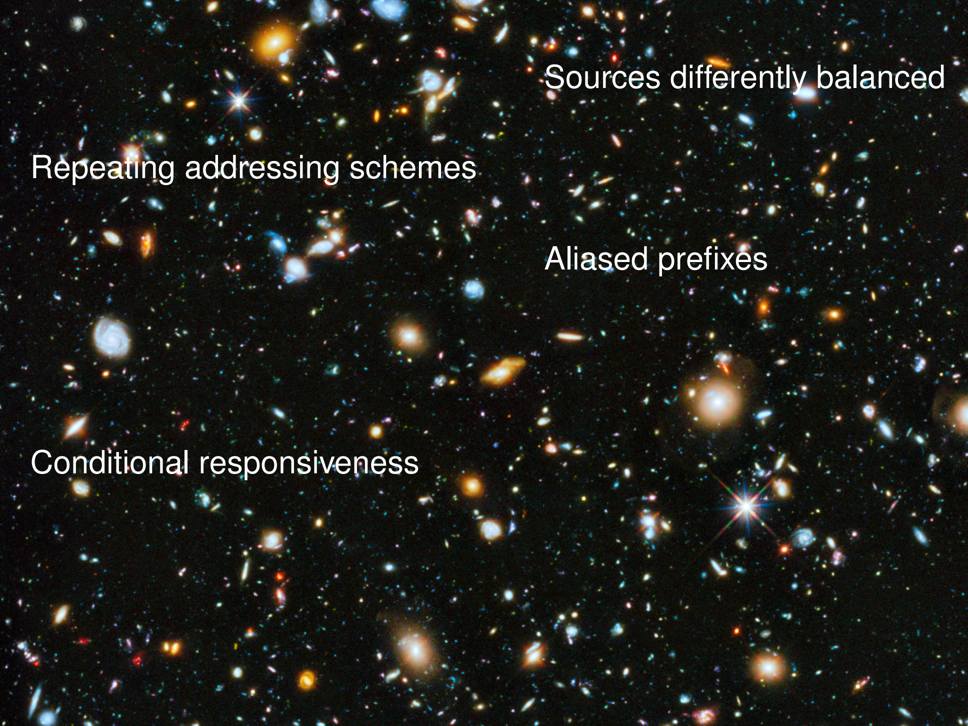


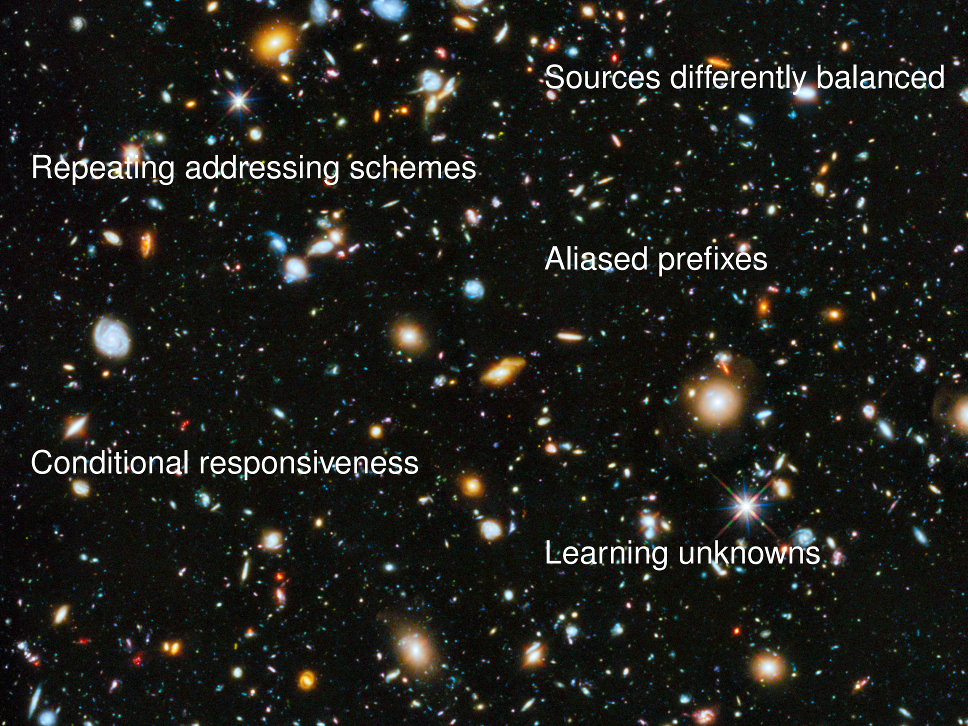



Sources differently balanced


Sources differently balanced
Repeating addressing schemes


Sources differently balanced
Repeating addressing schemes
Aliased prefixes


Sources differently balanced
Repeating addressing schemes
Aliased prefixes
Conditional responsiveness


Sources differently balanced
Repeating addressing schemes
Aliased prefixes
Conditional responsiveness
Learning unknowns

Sources differently balanced

Repeating addressing schemes

Aliased prefixes

ipv6hitlist.github.io

Conditional responsiveness

Learning unknowns

Oliver Gasser <gasser@net.in.tum.de>