Advanced Computer Networking (ACN)

IN2097 – WiSe 2023–2024

Prof. Dr.-Ing. Georg Carle

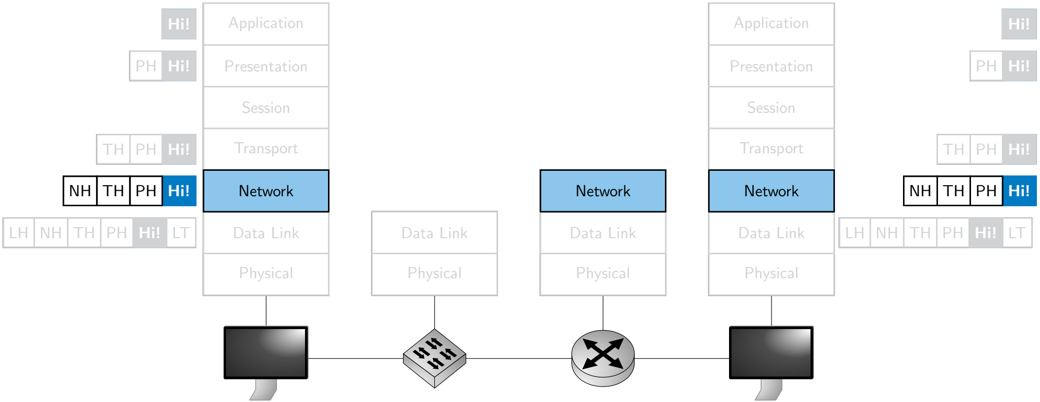
Sebastian Gallenmüller, Max Helm, Benedikt Jaeger,
Marcel Kempf, Patrick Sattler, Johannes Zirngibl

Chair of Network Architectures and Services
School of Computation, Information, and Technology
Technical University of Munich

Internet Protocol Version 6

Internet Protocol Version 6

Introduction
Network layer



Internet Protocol Version 6

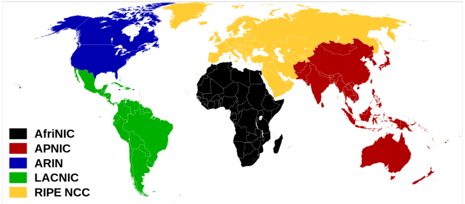
Motivation
IPv4 address space exhaustion

IPv4 address space is too small

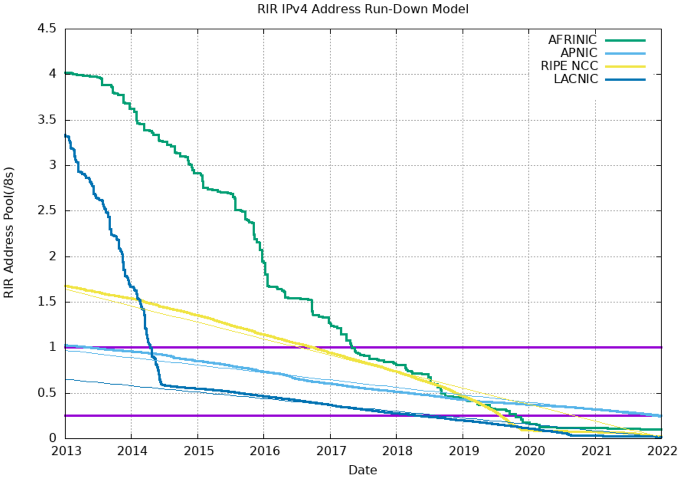
- Usage of NAT devices (home routers)
- Deployment of Carrier-Grade-NAT (CGN)
- IPv4 addresses are not end-to-end associative anymore

Timeline of Exhaustion

- 3 February 2011: IANA assigns last networks to RIRs
- 14 April 2011: APNIC depletion
- 14 September 2012: RIPE depletion
- 10 June 2014: LACNIC depletion
- 24 September 2015: ARIN depletion
- All RIRs have transition reserves
- AFRINIC is not yet depleted



Motivation
Overview of address space exhaustion[1]



1 Source: http://www.potaroo.net/tools/ipv4/

Motivation
Overview of address space exhaustion: RIPE NCC

RIPE NCC ran out of addresses on November 25th, 2019[2]

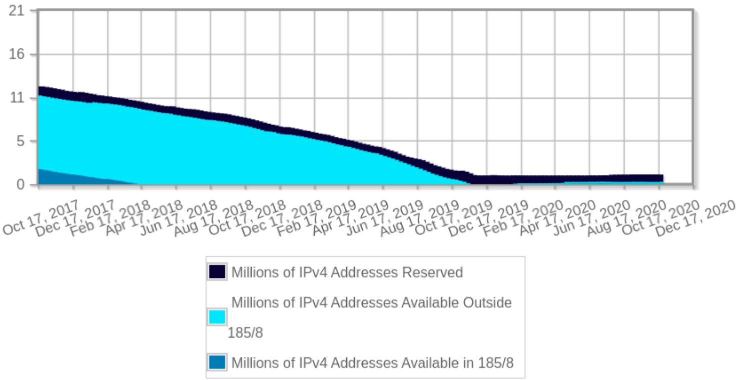


Figure 1: RIPE NCC IPv4 Pool — Last 36 Months (27.10.2020)[3]

2 https://www.ripe.net/manage-ips-and-asns/ipv4/getting-ready-for-ipv4-run-out
3 https://www.ripe.net/manage-ips-and-asns/ipv4/ipv4-available-pool

Overview of address space exhaustion: RIPE NCC

RIPE set up a wait list for LIRs to request IPv4 addresses

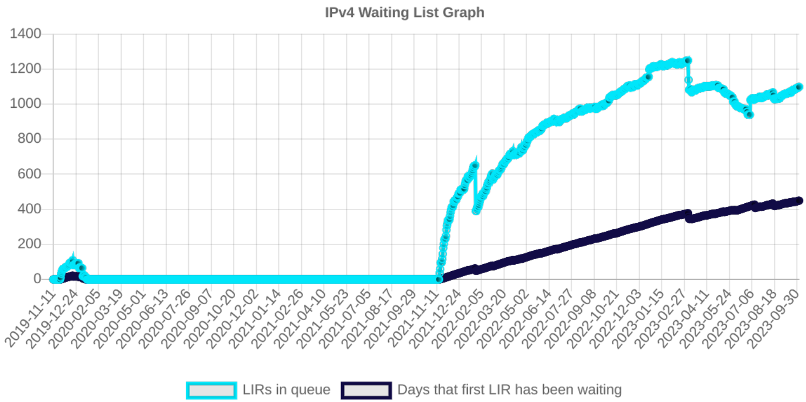


Figure 2: RIPE Waitlist (03.10.2023)[4]

4 https://www.ripe.net/manage-ips-and-asns/ipv4/ipv4-waiting-list

Motivation

- IPv6 designed as successor to IPv4
- IPv4 and IPv6 are not compatible by design
- First proposed in 1995 (RFC1883) [1]
- First standard dates back to 1998 (RFC2460) [2]
- Current standard was finalized in 2017 (RFC8200) [3]

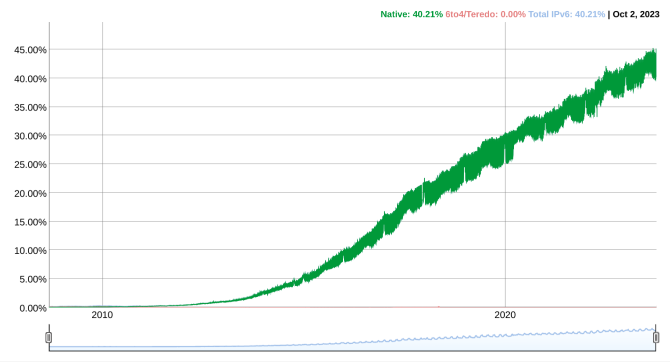


Figure 3: IPv6 adoption over time, `https://www.google.com/intl/en/ipv6/statistics.html`, 02.10.2023

Motivation
IPv6 adoption[5]

Region	IPv6 Capable			
	2023	2020	2019	2017
World	35.38%	27.33%	24.24%	15.80%
Americas	42.57%	33.34%	32.12%	23.96%
Asia	40.13%	31.34%	27.17%	15.42%
Oceania	35.05%	25.78%	19.72%	15.97%
Europe	31.80%	22.03%	18.29%	15.16%
Africa	1.99%	1.41%	2.02%	0.56%

5 Based on: `https://stats.labs.apnic.net/ipv6` (02.10.2023/27.10.2020/07.11.2017)

Motivation
IPv6 adoption[5]

Region	IPv6 Capable			
	2023	2020	2019	2017
World	35.38%	27.33%	24.24%	15.80%
Americas	42.57%	33.34%	32.12%	23.96%
Asia	40.13%	31.34%	27.17%	15.42%
Oceania	35.05%	25.78%	19.72%	15.97%
Europe	31.80%	22.03%	18.29%	15.16%
Africa	1.99%	1.41%	2.02%	0.56%

Country	IPv6 Capable
India	78.20%
France	67.24%
Malaysia	67.10%
Belgium	66.57%
Germany	64.14%
Uruguay	60.52%
Saudi Arabia	59.94%
Israel	59.08%
Montserrat	58.69%
Vietnam	57.79%

5 Based on: https://stats.labs.apnic.net/ipv6 (02.10.2023/27.10.2020/07.11.2017)

Motivation
Goals

- Better scalability
  - Larger address space size
- Easier deployment
  - Simplified header, easier implementation
  - Better auto-configuration (SLAAC)
  - Updated stateful configuration (DHCPv6)
- Easier analysis
  - Better flow associativity (see flow label)

Internet Protocol Version 6

Addresses

As for IPv4, an IPv6 unicast address identifies an interface connected to an IP subnet

IPv6 routinely allows each interface to be identified by several addresses

- Usually only one IPv4 address per interface

IPv6 addresses use 128 bits vs 32 bits for IPv4

- 2^{128} IPv6 addresses $\rightarrow 3.4 \times 10^{38}$ IPv6 addresses
- $10^{22} \rightarrow$ Estimated number of stars in the universe
- $3 \times 10^{31} \rightarrow$ Unique IPv6 addresses that would be assigned after 1 trillion years if a new IPv6 address was assigned at every picosecond

Addresses
IPv6 Address Representation

An IPv6 address is represented as 8 groups of 4 hexadecimal digits (16 bits), separated by a colon (`:`)

- Example: `2001:0db8:85a3:0000:0000:8a2e:0370:7334`

Leading zeroes in a group may be omitted, but each group must retain at least one hexadecimal digit.

- `2001:0db8:85a3:0000:0000:8a2e:0370:7334` → `2001:db8:85a3:0:0:8a2e:370:7334`

One or more consecutive groups of zero value may be replaced with a single empty group using two consecutive colons (`::`), but the substitution may only be applied once in the address

- `2001:db8:85a3:0:0:8a2e:370:7334` → `2001:db8:85a3::8a2e:370:7334`
- The localhost (loopback) address `0:0:0:0:0:0:0:1` is written as `::1`
- Representations are shortened as much as possible. The longest sequence of consecutive all-zero fields is replaced by double-colon.

Lowercase is recommended over uppercase:

- Example: `2001:db8::1` is preferred over `2001:DB8::1`

Representing IP and port numbers using brackets (`[ ]`)

- IPv4: `192.168.1.1:80`
- IPv6: `[2001:db8::1]:80`

IPv6 Address Representation - Examples

IPv6 Address		**Correct representation**
`2001:0db8::0001`	→	`2001:db8::1`
`2001:db8:0:0:0:0:2:1`	→	`2001:db8::2:1`
`2001:db8:0:0:1:0:0:1`	→	`2001:db8::1:0:0:1`
`2001:db8:0:1:1:1:1:1`	→	`2001:db8:0:1:1:1:1:1`

RFC5952: A Recommendation for IPv6 Address Text Representation

Note: The `2001:db8::/32` subnet is used only in documentation.

Noticeable IPv6 addresses

Facebook	`2a03:2880:fffe:c:face:b00c::35`
Sprint.net	`2600::`
BBC.com	`2001:41c1:400c::bbc:1`

IPv6 Address Subnets

The IPv4 CIDR notation is also used for IPv6:

- `address/prefix` → `2001:db8::1/64`

The design of the IPv6 address space implements a very different design philosophy than in IPv4.

In IPv6, the address space is deemed large enough for the foreseeable future, and a local area subnet always uses 64 bits for the host portion of the address, designated as the interface identifier, while the most-significant 64 bits are used as the routing prefix.

- First n bits: Global routing prefix
- Next 64 - n bits: Subnet ID
- Last 64 bits: Interface ID

Addresses
Address Types

Address scopes

- Unicast: From one sender to exactly one receiver
- Multicast: From one sender to all (multiples or one) members in a group
- Anycast: From one sender to one member of a group
- Broadcast: Not used in IPv6

Addresses
Address Types

Address scopes

- Unicast: From one sender to exactly one receiver
- Multicast: From one sender to all (multiples or one) members in a group
- Anycast: From one sender to one member of a group
- Broadcast: Not used in IPv6

Global-Unique IPv6 Addresses

- Globally unique and globally addressable

Local-Unique IPv6 Addresses

- Locally addressable
- Highly likely globally unique
- Part of the `fc00::/7` subnet

Link-Local IPv6 Addresses

- Addressable on the link
- Can be globally unique but only have to be unique on the link
- Part of the `fe80::/10` subnet

Site-Local IPv6 Addresses

- Developed at the beginning of IPv6, but dropped since then

Multicast Addresses

Multicast addresses have the following convention:

- First 8 bits set to `11111111`
- Next 4 bits as flag (permanent or transient address)
- Next 4 bits as scope:
  - `0000` Reserved
  - `0001` Interface local
  - `0010` Link local
  - `1000` Organization local
  - `1110` Global
  - `1111` Reserved
- Last 112 bits: Group ID

Predefined IPv6 Multicast Addresses

- All nodes multicast:
  - `ff01::1` (interface-local)
  - `ff02::1` (link-local)
  - In IPv4 `224.0.0.1` is used
- All routers multicast:
  - `ff01::2` (interface-local)
  - `ff02::2` (link-local)
  - In IPv4 `224.0.0.2` is used

Internet Protocol Version 6

Just as a reminder:

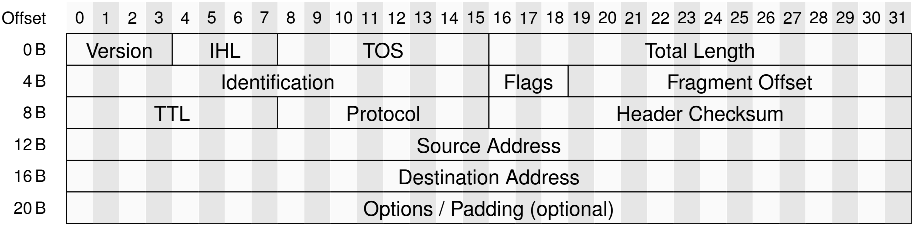


Figure 4: IPv4 header

- 12 Fields + Options
- 20 Bytes + Options

Header Format
IPv6 Header

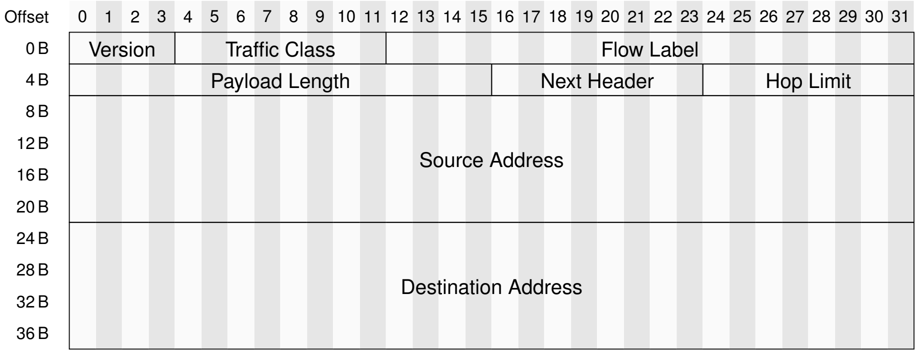


Figure 5: IPv6 header

Header Format
IPv6 Header

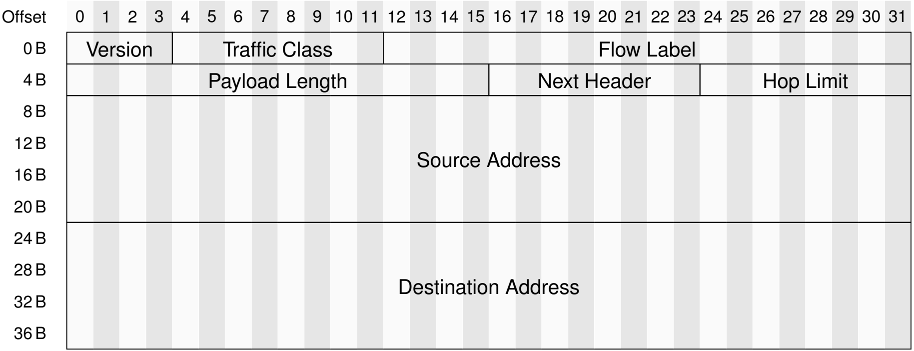


Figure 5: IPv6 header

Version	Always 0x6	Next Header	Type of payload (UDP, TCP, ...)
Traffic Class	QoS indicator	Hop Limit	No. of hops left until discarded
Flow Label	Same for each flow	Source Address	IPv6 address of sender
Payload length	Length of payload in octetts	Destination Address	IPv6 address of receiver

Header Format
IPv6 vs IPv4 Header

Changes between IPv4 and IPv6:

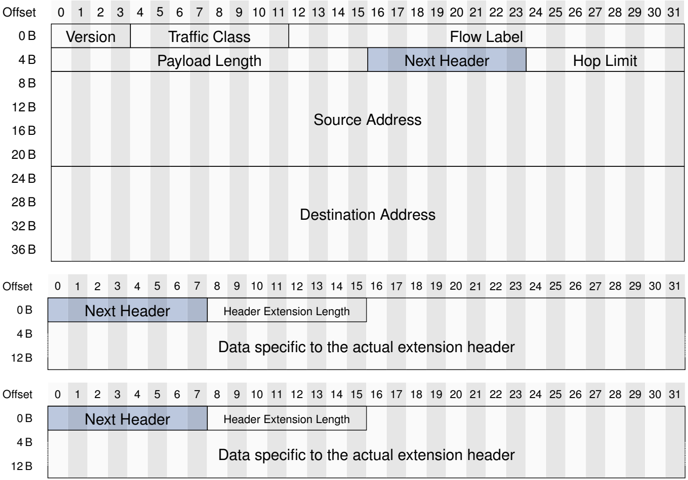
IPv4		**IPv6**
Header length	$\rightarrow$	Removed
Header checksum	$\rightarrow$	Removed
TOS	$\rightarrow$	Class
Total length	$\rightarrow$	Payload length
Fragmentation	$\rightarrow$	Extension header
TTL	$\rightarrow$	Hop limit
Protocol	$\rightarrow$	Next header
Option	$\rightarrow$	Extension headers

Header Format
IPv6 Flow Label

- 20 bit Flow Label field to identify specific flows needing special QoS
- Traditional IPv4 way of specifying flows:
  - 5-tuple: source and destination IP addresses, source and destination port numbers, and protocol type
- Some of these fields may be unavailable due to fragmentation, encryption, or locating them past extension headers
- With flow labels, each source chooses its own flow label value. Routers use IPv6 source address + flow label to identify distinct flows

Header Format
Extensions

The headers form a chain, using the Next Header fields.



Header Format
Extensions

Administered by IANA in a central registry[6]:

Number	Description	Reference
0	IPv6 Hop-by-Hop Option	[RFC8200]
43	Routing Header for IPv6	[RFC8200][RFC5095]
44	Fragment Header for IPv6	[RFC8200]
50	Encapsulating Security Payload	[RFC4303]
51	Authentication Header	[RFC4302]
60	Destination Options for IPv6	[RFC8200]
135	Mobility Header	[RFC6275]
139	Host Identity Protocol	[RFC7401]
140	Shim6 Protocol	[RFC5533]
253	Use for experimentation and testing	[RFC3692][RFC4727]
254	Use for experimentation and testing	[RFC3692][RFC4727]

6 https://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml

Header Format
Extensions

Hop-by-hop Options header

- TLV coded options processed by every hop along the path
- Jumbo payload option for packets larger than 65535 Bytes (RFC 2675)
- Router alert option (RFC 2711)

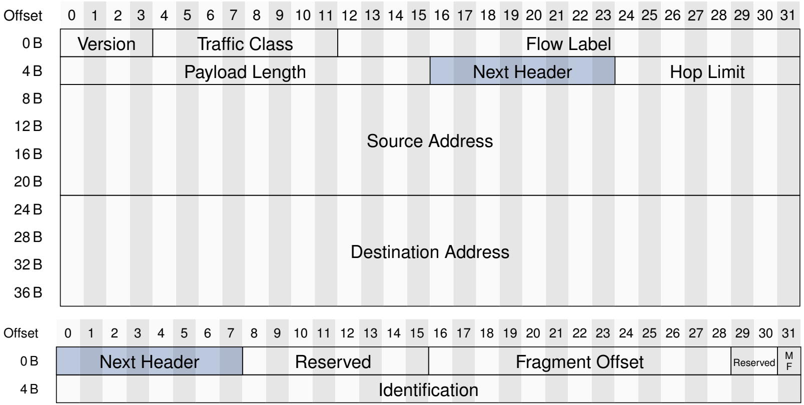
Routing header

- Strict or loose source routing
- Similar to the IPv4 Source route and Record route options

Fragment header

- Only source can fragment packets in IPv6
- Source must use Path MTU Discovery (RFC 8201) or send max 1240 Bytes payload
- Fragmentation information
  - Fragmentation offset shifted (as in IPv4)
  - Fragmentation ID is 32 bits (16 bits in IPv4)

Header Format
Fragmentation extension



Header Format
Extensions

Authentication header (IPSEC)

- To validate the message sender and ensure integrity of data

Encapsulated Security Payload header (IPSEC)

- To provide confidentiality and guard against eavesdropping

Destination Options header

- TLV coded options processed by destination only

Mobility header

- Parameters used with Mobile IPv6

Host Identity Protocol (HIP)

- For enabling continuity of communications across IP address changes

Shim6 Protocol

- Multihoming Shim Protocol for IPv6 with failover and load-sharing properties

Internet Protocol Version 6

Auto-configuration

Address resolution and ICMPv6

- ICMP has been revised along with the IPv6 development, with a new version: ICMPv6 (RFC4443)
- IPv6 does not use ARP but a neighbor detection scheme based on ICMPv6: Neighbor Discovery Protocol (RFC4861)

Neighbor Discovery Protocol (NDP)

Function

Goal

- IPv4 ARP Equivalent
- Resolve IPv6 addresses to MAC addresses
- Add parts for information about routers

Realization

- Part of ICMPv6 [4]
- Split into “Neighbor Solicitation” and “Neighbor Advertisement”

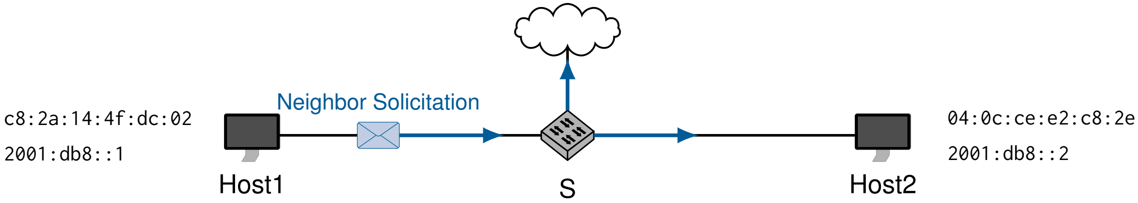
Neighbor Solicitation

- Ask for the MAC address of the interface configured for a given IPv6 address
- Destination IPv6 address (“Solicited Node Address”):
  - `ff02::1:ffXX:XXXX`
  - Insert lowest 24 bits of the given IPv6 address at the end of the destination IPv6 address
- Destination MAC address:
  - `33-33-XX-XX-XX-XX`
  - Insert lowest 32 bits of the “Solicited Node Address”

Neighbor Advertisement

- Answer a Neighbor Solicitation
- Use MAC and IPv6 addresses of the destination host

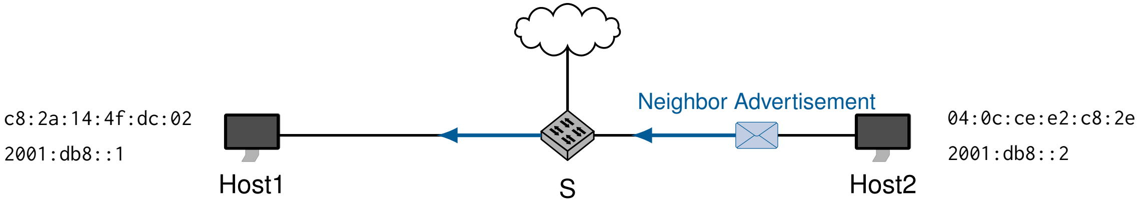
Example - Neighbor Solicitation / Advertisement



Destination MAC Address: `33:33:ff:00:00:02`

Destination IPv6 Address: `ff02::1:ff00:2`

Example - Neighbor Solicitation / Advertisement



Destination MAC Address: `c8:2a:14:4f:dc:02`

Destination IPv6 Address: `2001:db8::1`

Router Solicitation / Advertisement [4]

Router Solicitation (RS)

- Prompt all routers on this segment to send a Router Advertisement
- Normally sent when interface comes up

Router Advertisement (RA)

- Sent to the all-nodes multicast address in fixed intervals by all routers
- Contains Information about the network segment:
  - Autoconfiguration methods (SLAAC, DHCPv6)
  - Prefix Information
  - Route Information
  - MTU on link
  - Link-Layer address of the router

Neighbor Discovery Protocol (NDP)

Address resolution and ICMPv6

- ICMP has been revised along with the development IPv6, with a new version: ICMPv6 (RFC4443)
- IPv6 does not use ARP but a neighbor detection scheme based on ICMPv6: Neighbor Discovery Protocol (RFC4861)
- ICMPv6 support additional services:
  - Secure Neighbor Discovery (SEND) is an extension of NDP with extra security
  - Multicast Listener Discovery (MLD) is used by IPv6 routers for discovering multicast listeners on a directly attached link
  - Multicast Router Discovery (MRD) allows discovery of multicast routers
- Everybody can claim to be a router
  - Use RA Guard to filter unauthorized RAs (RFC 6105)
  - Secure Neighbor Discovery (SEND) is an extension of NDP with extra security (RFC 3971)

Neighbor Discovery Protocol (NDP)

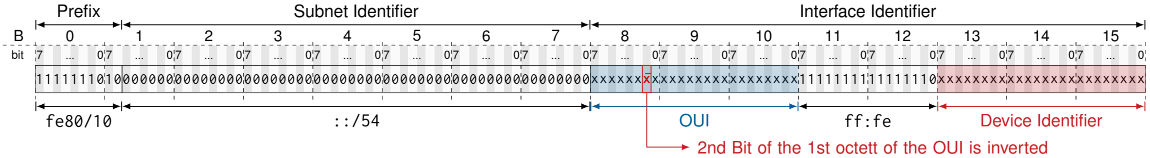
Address resolution and ICMPv6

- ICMP has been revised along with the development IPv6, with a new version: ICMPv6 (RFC4443)
- IPv6 does not use ARP but a neighbor detection scheme based on ICMPv6: Neighbor Discovery Protocol (RFC4861)
- ICMPv6 support additional services:
  - Secure Neighbor Discovery (SEND) is an extension of NDP with extra security
  - Multicast Listener Discovery (MLD) is used by IPv6 routers for discovering multicast listeners on a directly attached link
  - Multicast Router Discovery (MRD) allows discovery of multicast routers
- Everybody can claim to be a router
  - Use RA Guard to filter unauthorized RAs (RFC 6105)
  - Secure Neighbor Discovery (SEND) is an extension of NDP with extra security (RFC 3971)

Stateless auto-configuration / Serverless

- Automatic configuration of link-local addresses on system startup

SLAAC
StateLess Address AutoConfiguration



Use case

- Automatic configuration of link-local address

Procedure

1. Create EUI-64 Address
   - Always in subnet `fe80/64`
   - First 24 Bit of Interface Identifier: First 24 Bit of MAC address (flip second bit of the first octet)
   - “Middle” 16 Bit: Always `ff:fe`
   - Last 24 Bit: Last 24 Bit of MAC address
2. Perform Duplicate Address Detection (DAD)
3. Configure Address to interface

SLAAC
StateLess Address AutoConfiguration

- Use link-local address and interface ID
- Hosts join all-nodes multicast address (`ff02::1`)
- Hosts do DAD with all nodes based on multicast address
- Hosts communicate to routers using all-routers multicast address (`ff02::2`)
- ICMPv6 router solicitation sent by host to request additional information
- ICMPv6 router advertisement sent by router to inform host about prefixes for site and global addresses

Privacy and security considerations

SLAAC uses a modified MAC address, which makes it possible to trace a device

- RFC 4941 "Privacy Extensions"
- Use random 64 bit number for the host part
- Change the number regularly

SLAAC

Address resolution and ICMPv6

- ICMP has been revised along with the development IPv6, with a new version: ICMPv6 (RFC4443)
- IPv6 does not use ARP but a neighbor detection scheme based on ICMPv6: Neighbor Discovery Protocol (RFC4861)
- ICMPv6 support additional services:
  - Secure Neighbor Discovery (SEND) is an extension of NDP with extra security
  - Multicast Listener Discovery (MLD) is used by IPv6 routers for discovering multicast listeners on a directly attached link
  - Multicast Router Discovery (MRD) allows discovery of multicast routers
- Everybody can claim to be a router
  - Use RA Guard to filter unauthorized RAs (RFC 6105)
  - Secure Neighbor Discovery (SEND) is an extension of NDP with extra security (RFC 3971)

Stateless auto-configuration / Serverless

- Automatic configuration of link-local addresses on system startup

SLAAC

Address resolution and ICMPv6

- ICMP has been revised along with the development IPv6, with a new version: ICMPv6 (RFC4443)
- IPv6 does not use ARP but a neighbor detection scheme based on ICMPv6: Neighbor Discovery Protocol (RFC4861)
- ICMPv6 support additional services:
  - Secure Neighbor Discovery (SEND) is an extension of NDP with extra security
  - Multicast Listener Discovery (MLD) is used by IPv6 routers for discovering multicast listeners on a directly attached link
  - Multicast Router Discovery (MRD) allows discovery of multicast routers
- Everybody can claim to be a router
  - Use RA Guard to filter unauthorized RAs (RFC 6105)
  - Secure Neighbor Discovery (SEND) is an extension of NDP with extra security (RFC 3971)

Stateless auto-configuration / Serverless

- Automatic configuration of link-local addresses on system startup

Stateful configuration (managed)

- Flag in router advertisement tells whether to rely on auto-configuration or to use conventional managed configuration

→ DHCPv6

DHCPv6

SLAAC and router advertisements can configure an IPv6 host and provide connectivity.
Why do we need DHCPv6?

SLAAC and router advertisements can configure an IPv6 host and provide connectivity.
Why do we need DHCPv6?

- Can be used to provide prefix information
- Can be used to provide fixed addresses to device identifiers
- Can be used to provide DNS information
- Has to be used to provide boot information for Netboot

Internet Protocol Version 6

Summary

IPv6 offers:

- 128 bit address space $\rightarrow$ 340 trillion addresses
- Revised and simplified header format
- New options and extensions
- Increased security measures

IPv6 uses hexadecimal colon notation with abbreviation methods

IPv6 has three address types: unicast, anycast, and multicast

ICMPv4, ARP, RARP, and IGMP replaced with ICMPv6

IPv4 to IPv6 transition strategies are based on dual-stack and tunneling

IPv6 adoption:

- IPv6 is supported by all major operating systems and all major router vendors
- Still some work to be done for complete adoption

Internet Protocol Version 6

IPv6 Scans
ZMapv6

Original ZMap implementation supports only IPv4

- Extension of ZMap with IPv6 capabilities $\rightarrow$ ZMapv6
- https://github.com/tumi8/zmap
- Adaptation of scanning core to send and receive IPv6 packets
- Port probe modules for IPv6 scanning: ICMPv6, TCP over IPv6, UDP over IPv6

IPv6 Scans

Challenges

- Vast address space → “0/0” scan not possible
  - Scan rate 20k IP addr/s → 5.4×10^{26} years

→ Hitlists are required

IPv6 Scans

Challenges

- Vast address space → "0/0" scan not possible
  - Scan rate 20k IP addr/s → 5.4×10^{26} years

→ Hitlists are required

A list of **targets**, most likely **responsive**, of **feasible size**.

IPv6 Scans

Challenges

- Vast address space → “0/0” scan not possible
  - Scan rate 20k IP addr/s → 5.4×10^{26} years

→ Hitlists are required

A list of **targets**, most likely **responsive**, of **feasible size**.

Responsive:

- Responsive to at least one protocol (e.g., ICMP, HTTP, . . .)
- Different between addresses
- Changes over time

IPv6 Scans

Challenges

- Vast address space → "0/0" scan not possible
  - Scan rate 20k IP addr/s → 5.4×10^{26} years

→ Hitlists are required

A list of **targets**, most likely **responsive**, of **feasible size**.

Responsive:

- Responsive to at least one protocol (e.g., ICMP, HTTP, . . .)
- Different between addresses
- Changes over time

Feasible size:

- Scan duration
- Bandwidth limitations

IPv6 - Sources

Possible sources for an IPv6 hitlist

- List of addresses
- List of domains
  - Unranked
  - Ranked
- Active scans
- Machine learning

IPv6 Scans
IPv6 - List of Addresses

A list of known addresses from passive sources.

Possible sources:

- Raw packet traces
  - Extract IPv6 addresses from live traffic
- Flow data (NetFlow, IPFIX)
  - Export flow data from routers and collect at measurement point
  - Extract IPv6 addresses from flow data
- Traceroutes
  - Often used for the analysis of network paths and structure
  - Reveals addresses of hops on the path
  - e.g. with Scamper

IPv6 Scans
IPv6 - List of Domains

A list of existing domains can be resolved into used addresses.

- Unranked lists
- Extracted from other datasets
- Side products of other scans

→ Targets highly depend on the source

IPv6 - List of Domains

A list of existing domains can be resolved into used addresses.

- Unranked lists
- Extracted from other datasets
- Side products of other scans

→ Targets highly depend on the source

Possible sources of unranked lists:

- DNS zone files
  - Content of complete top-level domain name zone
  - `.com`, `.net`, `.org`, `...` are available via contract with Verisign or paid services (e.g. premiumdrops.com)
  - New gTLDs are available via ICANN's Centralized Zone Data Service (CZDS)
- Certificate Transparency (CT)
  - Extract domains from *Common Name*, *Subject Alternative Name* entries of logged certificates

IPv6 Scans
Top Lists

A Top List is a list of domains ranked by their popularity.

- Ranked list of domains
- Popularity calculated by different measures
- Normally one million entries
- Most popular top lists:
  - Alexa Top List ← Deprecated
  - Majestic Million
  - Umbrella
  - Cloudflare Radar
  - Google Crux

Alexa Top List (Deprecated since February 2023)

- Provided by Amazon
- Based on HTTP requests
  - Collected with a browser toolbar
  - Depends on volunteers to install the toolbar
  - Captures statistics about visited web pages
  - → Strong focus towards web pages

The Majestic Million[7]

- Independent organization
- Based on link metrics
  - Combination of outgoing and incoming links
  - Collected by a web crawler
  - Data updated several times a day
  - → Focus towards web pages

7 https://majestic.com/

IPv6 Scans
Top Lists

Umbrella[8]

- Provided by Cisco
- Based on DNS requests to the Umbrella global Network (formerly OpenDNS)
- Algorithm based on unique client IPs visiting a domain
- Calculates Internet popularity independent of the port

→ No focus towards web traffic

Cloudflare Radar[9]

- Collected and published by Cloudflare
- Based on resolver statistics (1.1.1.1)
- Combines client and
- servers, cloud infrastructure, IoT devices, home routers, and bots

→ No focus towards web traffic

8 http://s3-us-west-1.amazonaws.com/umbrella-static/index.html
9 https://radar.cloudflare.com/domains

IPv6 Scans
Top Lists

Google Crux[10]

- Provided by Google
- Based on Chrome histories and user behavior
- Only provides data in bins, e.g., [1,1000], [1000,5000]

→ Focus towards web traffic

Comparison of Top 5 on 22.10.2023

Rank	Majestic	Umbrella	Radar
1	facebook.com	google.com	google.com
2	google.com	www.google.com	googleapis.com
3	youtube.com	microsoft.com	apple.com
4	twitter.com	data.microsoft.com	facebook.com
5	instagram.com	digicert.com	gstatic.com

10 https://developer.chrome.com/docs/crux/

IPv6 Scans
Top Lists Changes and Clustering

Treat Top Lists carefully:

- Frequent changes over time [5]
- Weekend effect [5], [6]
  - Different user behavior changes lists on the weekend
  - Focus towards entertainment and streaming on the weekend
- Clustering Effect [6]
  - Large clusters with same rank
  - Ordered alphabetically
- Size is not always 1 million

IPv6 Scans
Top Lists Changes and Clustering

Treat Top Lists Carefully:

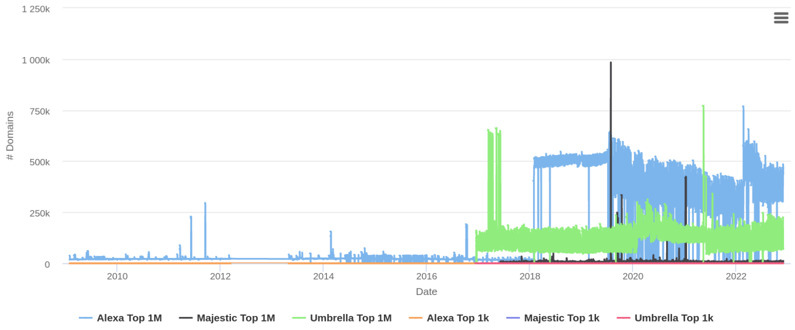


Figure 6: Day-to-Day Changes per List, 02.12.2022, `https://toplists.github.io/`

IPv6 Scans
IPv6 - List of Domains

Possible sources (continued):

- Rapid7 IPv4 rDNS
  - Complete reverse DNS resolution of IPv4 addresses
  - Published weekly on scans.io
- Rapid7 DNS ANY
  - Use domains gathered from other scans for DNS ANY scans
  - Published weekly on scans.io
- CAIDA IPv6 router DNS names
  - rDNS resolution of IPv6 addresses obtained from traceroute measurements on the Ark measurement infrastructure
  - Request access on caida.org

IPv6 Scans
IPv6 - rDNS Walking

rDNS Walking is an example of an active Scan resulting in a hitlist.

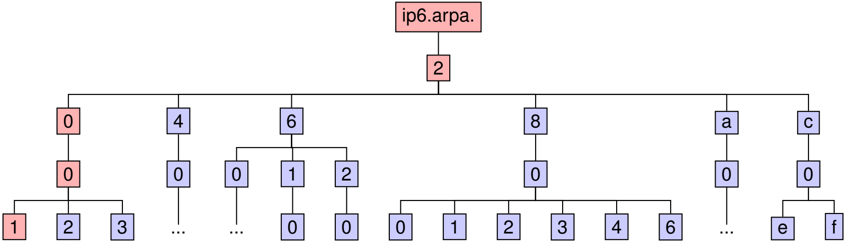
DNS:

- Resolve human readable domains into addresses
- Domain $\xrightarrow{\text{A/AAAA}}$ Address

Reverse DNS:

- Address $\xrightarrow{\text{PTR}}$ Domain
- Addresses represented as subdomains of ip6.arpa.
    - 1080::8:800:200c:417a
    - a.7.1.4.c.0.0.2.0.0.8.0.8.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.1.ip6.arpa.

IPv6 Scans
IPv6 - rDNS Walking



IPv6 Scans
IPv6 - rDNS Walking

Idea

- Mentioned by Peter van Dijk in a blog post[11] in 2012
- Added to RFC7707 [7]
- Implemented as Internet-wide Scan by Fiebig et al. [8] in 2017

Requirement - NXDomain

- DNS response code
- Neither the domain, nor any subdomain exists [9]

11 https://web.archive.org/web/20161121215042/http://7bits.nl/blog/posts/finding-v6-hosts-by-efficiently-mapping-ip6-arpa

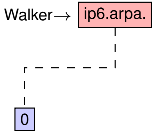
IPv6 Scans
IPv6 - rDNS Walking

- Start at root ip6.arpa.

Walker→ ip6.arpa.

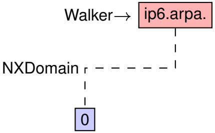
IPv6 Scans
IPv6 - rDNS Walking

- Start at root ip6.arpa.
- Query first nibble value



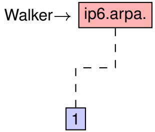
IPv6 Scans
IPv6 - rDNS Walking

- Start at root ip6.arpa.
- Query first nibble value
- Prune whole subtree in case of NXDomain



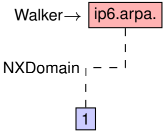
IPv6 Scans
IPv6 - rDNS Walking

- Start at root ip6.arpa.
- Query first nibble value
- Prune whole subtree in case of NXDomain
- Query next nibble value



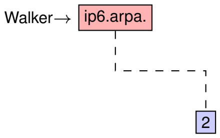
IPv6 Scans
IPv6 - rDNS Walking

- Start at root ip6.arpa.
- Query first nibble value
- Prune whole subtree in case of NXDomain
- Query next nibble value



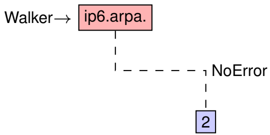
IPv6 Scans
IPv6 - rDNS Walking

- Start at root ip6.arpa.
- Query first nibble value
- Prune whole subtree in case of NXDomain
- Query next nibble value



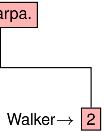
IPv6 Scans
IPv6 - rDNS Walking

- Start at root ip6.arpa.
- Query first nibble value
- Prune whole subtree in case of NXDomain
- Query next nibble value



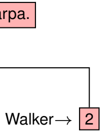
IPv6 Scans
IPv6 - rDNS Walking

- Start at root ip6.arpa.
- Query first nibble value
- Prune whole subtree in case of NXDomain
- Query next nibble value
- Descend into subtree



IPv6 Scans
IPv6 - rDNS Walking

- Start at root ip6.arpa.
- Query first nibble value
- Prune whole subtree in case of NXDomain
- Query next nibble value
- Descend into subtree
- Descend until full address is reached



IPv6 Scans
IPv6 - rDNS Walking

Full IPv6 scan:

- Query rate: 200 queries per name server
- Scan duration: 7 to 10 days
- Large query overhead
  - All 16 permutations of each nibble are queried
  - Majority replies are NXDomain

Results:

- 1.2 Mio /64 prefixes
- 9 Mio addresses
- Addresses cover > 5k autonomous systems
- Most popular ASes
  - KPN
  - Yandex
  - Yahoo

IPv6 Scans
IPv6 - rDNS Walking

Result distribution:

- Shows frequency of nibble values
- The first nibble is always 2
- Patterns can be seen, e.g. ff:fe

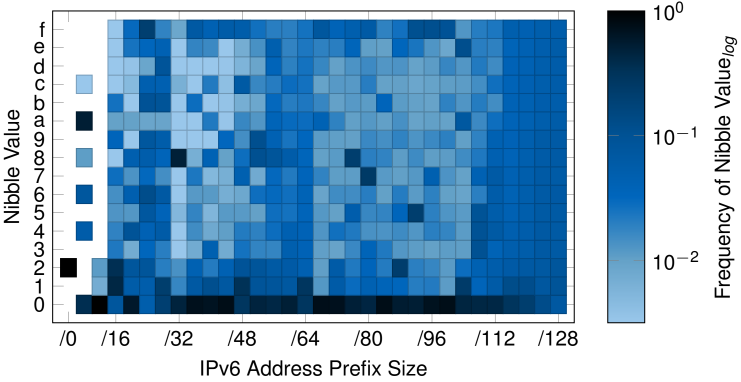


Figure 7: Probability Mass Function for each Nibble

IPv6 Scans
IPv6 - Machine Learning

Learn addresses from schemes in existing datasets

- Relies on responsive addresses as seed list
- Addresses are often assigned with a specific pattern
  - e.g. MAC-based IIDs with *ff:fe*
  - Servers with a fixed schema
- These patterns can be used to learn new addresses
- Entropy/IP [10]
  - Calculate entropy of addresses
  - Transform to a Bayesian network model
  - Walk the model to generate addresses
- 6Gen [11]
  - Cluster addresses

→ Good approach to extend existing hitlists with comparable responsiveness

Target Bias

Evaluate Interface ID (IID) portion of IPv6 addresses to determine device type

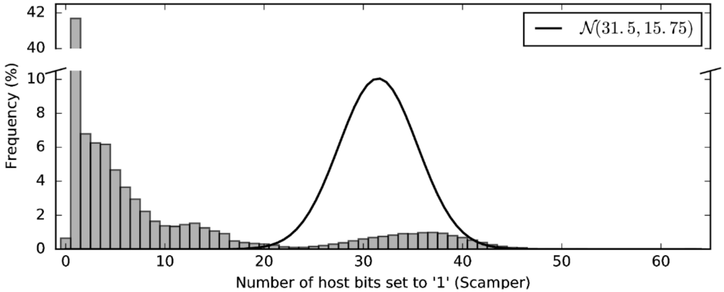


Figure 8: Hamming weight distribution of interface ID [12]

- Traceroute source (scamper) contains routers
- Router IP addresses are assigned mostly manually
- Most commonly only one bit of IID set to ‘1’ → e.g. ::1 for default gateway

Target Bias

Evaluate Interface ID (IID) portion of IPv6 addresses to determine device type

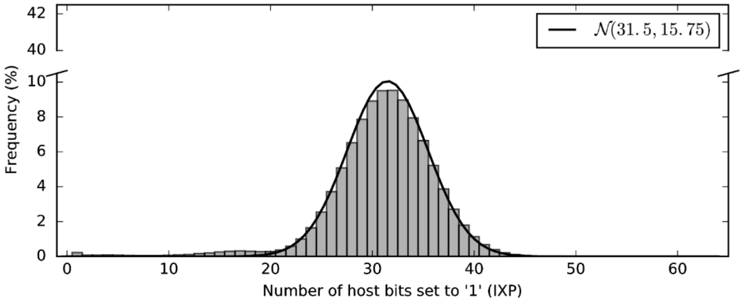


Figure 9: Hamming weight distribution of interface ID [12]

- IXP source contains many client devices
- Clients make extensive use of IPv6 Privacy Extensions
- Central limit theorem $\rightarrow$ sum of single-bit distributions approximates normal distribution

AS Bias

Hitlists can have a bias not only towards device types but also autonomous systems.

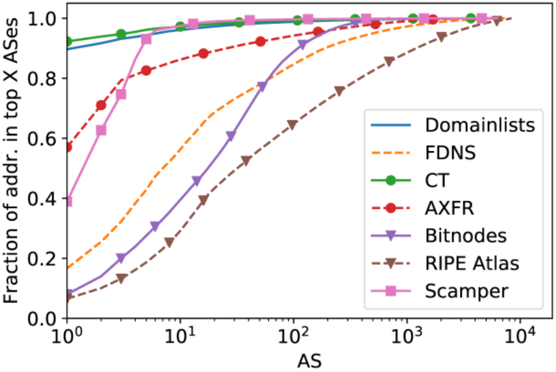


Figure 10: AS distribution for hitlist sources [13]

Aliased Prefixes

Problem:

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

Detection [13]:

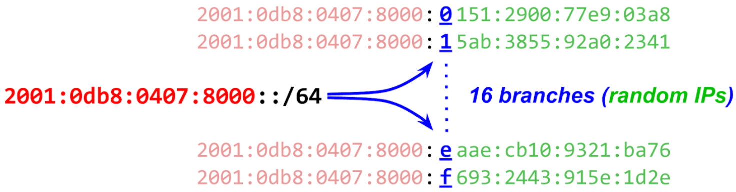
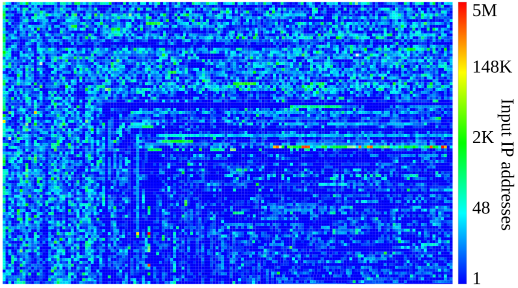


Figure 11: Aliased prefix detection using multi-level pseudo-random probing

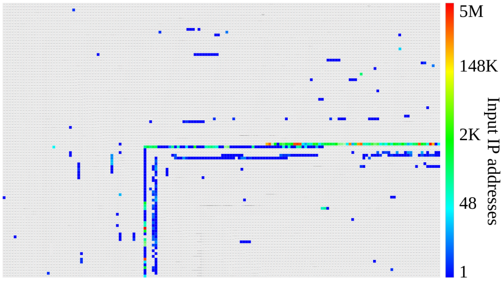
Aliased Prefixes

Results:

- Only small number of prefixes are aliased
- But nearly half of the addresses are in aliased prefixes
- Validated using fingerprinting (initial TTL, TCP options, timestamps)
- Aliased prefix detection is crucial to reduce bias



(a) Unfiltered Prefixes



(b) Aliased Prefixes

Figure 12: Responses to ICMP Echo requests [13]

Hitlist Biases

However, many of these are not necessarily aliased but fully responsive:

- Aliased prefixes are often announced by CDNs.
  - For Fastly, more than 98 % of announced IPv6 addresses are aliased.

Hitlist Biases

However, many of these are not necessarily aliased but fully responsive:

- Aliased prefixes are often announced by CDNs.
  - For Fastly, more than 98 % of announced IPv6 addresses are aliased.
- Many domains resolve to IPv6 addresses in aliased prefixes.
  - For Cloudflare, aliased prefixes host more than 10 M domains.

Hitlist Biases

However, many of these are not necessarily aliased but fully responsive:

- Aliased prefixes are often announced by CDNs.
  - For Fastly, more than 98 % of announced IPv6 addresses are aliased.
- Many domains resolve to IPv6 addresses in aliased prefixes.
  - For Cloudflare, aliased prefixes host more than 10 M domains.
- Fingerprinting reveals different behavior between hosts within the same aliased prefix.

However, many of these are not necessarily aliased but fully responsive:

- Aliased prefixes are often announced by CDNs.
  - For Fastly, more than 98 % of announced IPv6 addresses are aliased.
- Many domains resolve to IPv6 addresses in aliased prefixes.
  - For Cloudflare, aliased prefixes host more than 10 M domains.
- Fingerprinting reveals different behavior between hosts within the same aliased prefix.

We tested the responsiveness to other protocols for one address out of 60 k aliased prefix:

- 32.3 k are responsive to TCP/443
- 28.8 k are responsive to UDP/443
- 172 are responsive to UDP/53

However, many of these are not necessarily aliased but fully responsive:

- Aliased prefixes are often announced by CDNs.
  - For Fastly, more than 98 % of announced IPv6 addresses are aliased.
- Many domains resolve to IPv6 addresses in aliased prefixes.
  - For Cloudflare, aliased prefixes host more than 10 M domains.
- Fingerprinting reveals different behavior between hosts within the same aliased prefix.

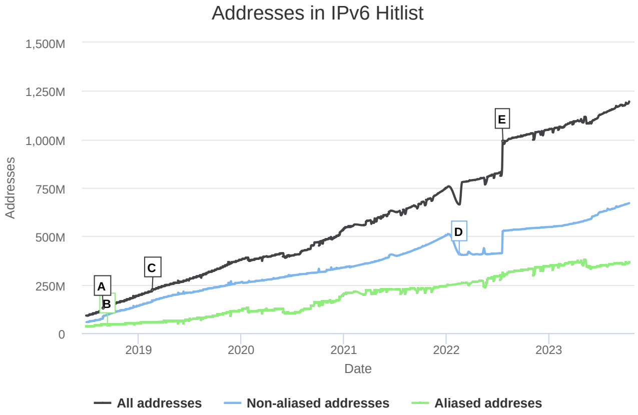
We tested the responsiveness to other protocols for one address out of 60 k aliased prefix:

- 32.3 k are responsive to TCP/443
- 28.8 k are responsive to UDP/443
- 172 are responsive to UDP/53

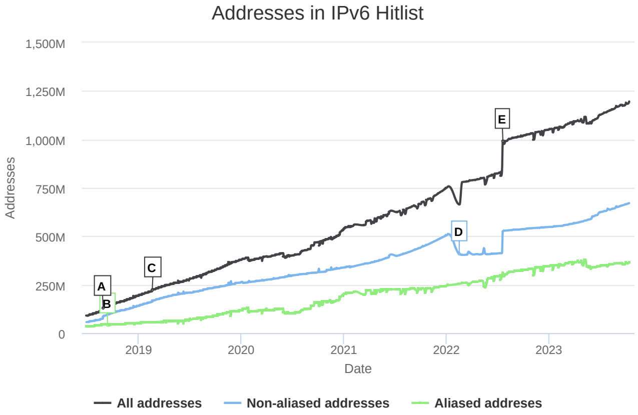
→ Including addresses from fully responsive prefixes should be considered in research relying on the IPv6 Hitlist.

- Research at this Chair
- Aggregates multiple inputs
- Filters aliased prefixes and applies blocklists
- Tests reachability daily
  - ICMPv6
  - TCP/80 (HTTP)
  - TCP/443 (HTTPS)
  - UDP/53 (DNS)
  - UDP/443 (QUIC)
- Uses ZMapv6

IPv6 Hitlist Service
Input

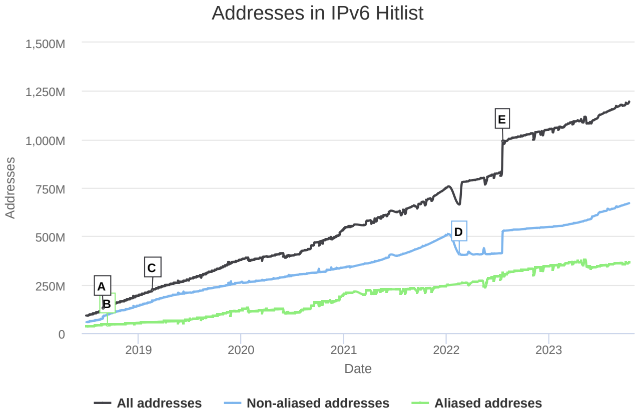


IPv6 Hitlist Service
Input



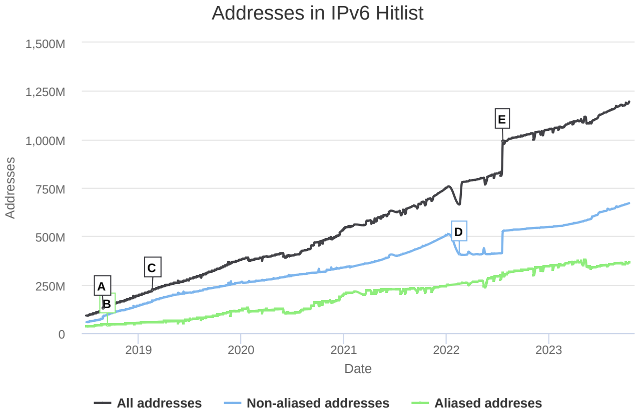
- A: Addition of IPv6 rDNS

Input



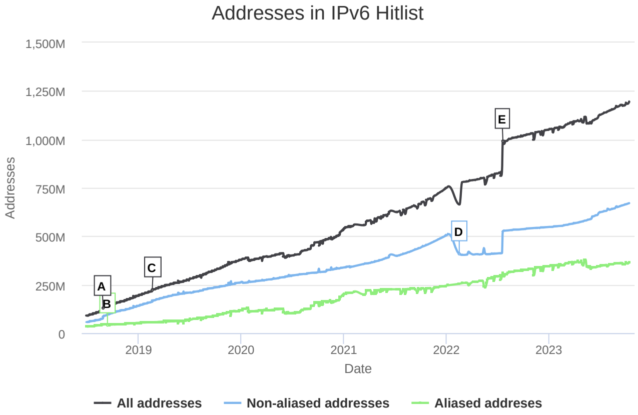
- A: Addition of IPv6 rDNS
- B: Withdrawal of two Amazon EC2 prefixes

IPv6 Hitlist Service
Input



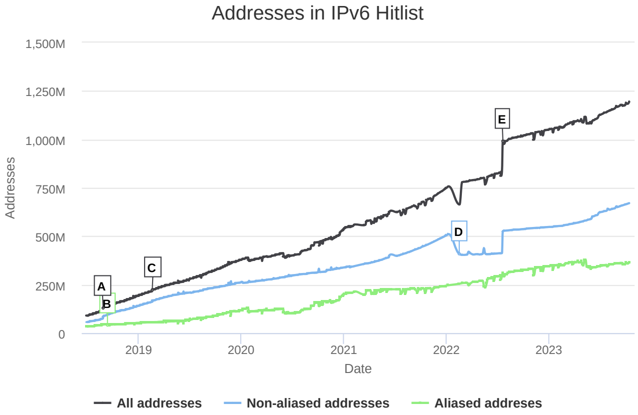
- A: Addition of IPv6 rDNS
- B: Withdrawal of two Amazon EC2 prefixes
- C: Additional IPv6 rDNS results

IPv6 Hitlist Service
Input



- A: Addition of IPv6 rDNS
- B: Withdrawal of two Amazon EC2 prefixes
- C: Additional IPv6 rDNS results
- D: Removal of addresses impacted by the GFW

IPv6 Hitlist Service
Input



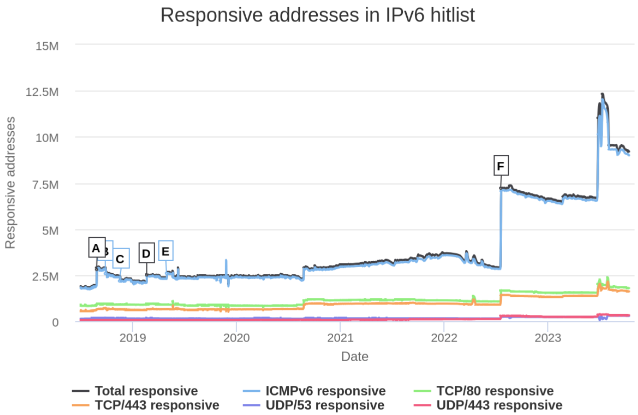
- A: Addition of IPv6 rDNS
- B: Withdrawal of two Amazon EC2 prefixes
- C: Additional IPv6 rDNS results
- D: Removal of addresses impacted by the GFW
- E: Addition of new addresses from passive sources and target generation methods

Filter

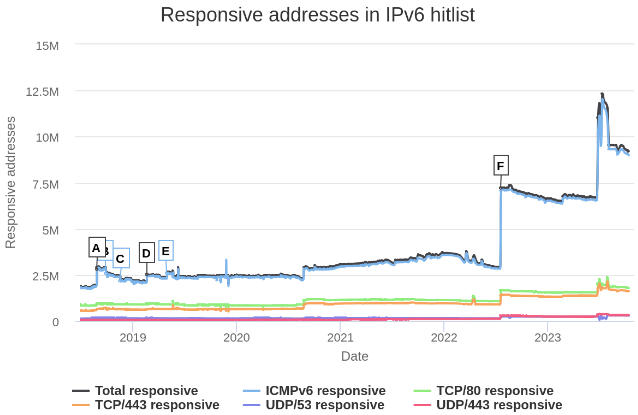
The input has to be filtered by several steps before the responsiveness can be tested.

- Not globally routed
  - Datasets might contain addresses that are not routed
  - Infrastructure changes might change the reachability of addresses
- Blocklisted
  - Aggregated list of blocklisting requests from all scans at our chair
- Aliased prefixes
- Not responsive for 30 consecutive days

IPv6 Hitlist Service
Results

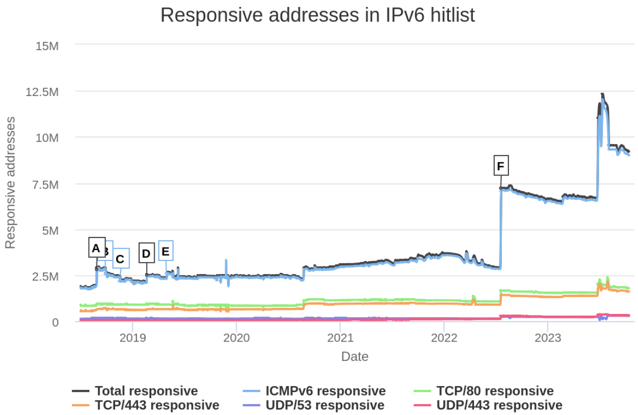


IPv6 Hitlist Service
Results



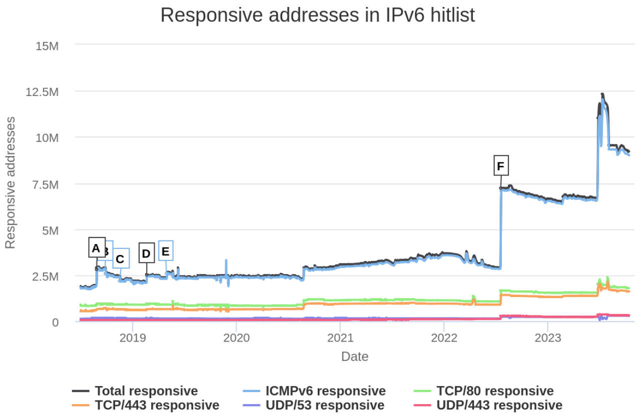
- A, D, E: Addition of IPv6 rDNS

Results



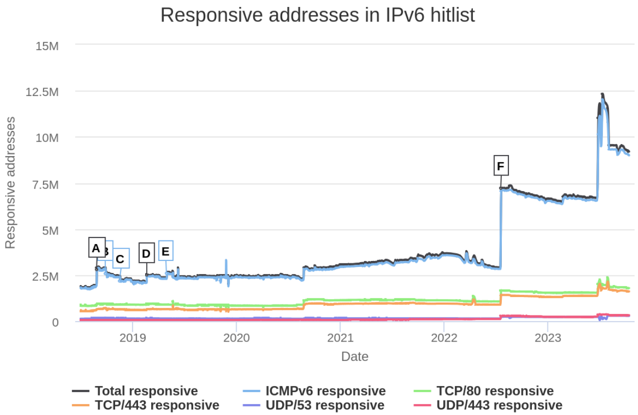
- A, D, E: Addition of IPv6 rDNS
- B: LATNET SERVISS stopped responding

IPv6 Hitlist Service
Results



- A, D, E: Addition of IPv6 rDNS
- B: LATNET SERVISS stopped responding
- C: Online S.A.S. stopped responding

Results



- A, D, E: Addition of IPv6 rDNS
- B: LATNET SERVISS stopped responding
- C: Online S.A.S. stopped responding
- F: Addition of new address candidates

New Sources

While the existing IPv6 Hitlist sources regularly update the input, new sources have not been added.

We evaluated different approaches to extend our hitlist [14]:

- Target Generation:
  - 6Tree, 6Graph, 6GAN, 6VecLM,
  - Distance Clustering (DC) (our custom algorithm)
- 30-day unresponsive addresses

Method	Addr	Responsive Addr. ↓	Responsive ASes
6Graph	125.8 M		
6Tree	37.6 M		
DC	5.3 M		
6GAN	3.3 M		
6VecLM	70.3 k		
30-day Unresp.	405.0 M		

New Sources

While the existing IPv6 Hitlist sources regularly update the input, new sources have not been added.

We evaluated different approaches to extend our hitlist [14]:

- Target Generation:
  - 6Tree, 6Graph, 6GAN, 6VecLM,
  - Distance Clustering (DC) (our custom algorithm)
- 30-day unresponsive addresses

Method	Addr	Responsive Addr. ↓	Responsive ASes
6Graph	125.8 M	3.8 M	10.7 k
6Tree	37.6 M	2.2 M	11.5 k
DC	5.3 M	651.9 k	5.5 k
6GAN	3.3 M	4.3 k	39
6VecLM	70.3 k	1.0 k	105
30-day Unresp.	405.0 M	1.3 M	9.0 k

→ All sources contribute additional responsive addresses.
→ We identified 5.6 M new responsive IPv6 addresses from 14.6 k ASes.

Summary

Hitlists:

- Lots of possible sources
- Knowledge about sources is important
- Number of IP addresses is not only metric $\rightarrow$ evaluate reachability and stability
- Optimal sources depend on type of measurement (end-user devices, servers, routers,. . .)
- Be aware of biases in your hitlist (address distribution, prefix/AS distribution, aliased prefixes)

Is client tracking impossible?

The privacy extension prevents tracking of clients by randomization of the interface ID.

- In general, most end devices implement the privacy extension
- 90% of IPv6 addresses seen by a large CDN are only seen once in long-running analyses [15]

Is client tracking impossible?

The privacy extension prevents tracking of clients by randomization of the interface ID.

- In general, most end devices implement the privacy extension
- 90% of IPv6 addresses seen by a large CDN are only seen once in long-running analyses [15]
- But how about Customer Premises Equipment (CPE)?
  - e.g., private networks use a single, fixed router (the CPE) as gateway to the Internet

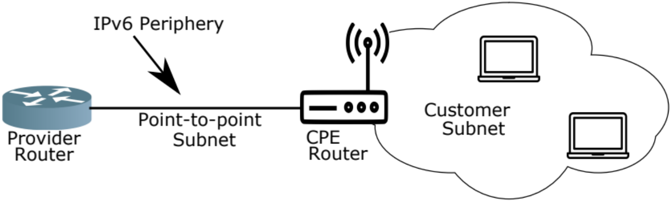


Figure 13: Common IPv6 architecture [16]

Is client tracking impossible?

The privacy extension prevents tracking of clients by randomization of the interface ID.

- In general, most end devices implement the privacy extension
- 90% of IPv6 addresses seen by a large CDN are only seen once in long-running analyses [15]
- But how about Customer Premises Equipment (CPE)?
  - e.g., private networks use a single, fixed router (the CPE) as gateway to the Internet
  - → In 2020, an approach to discover the IPv6 periphery was presented [16]
  - → A measurement revealed 64.8M router addresses
  - → 30M addresses were found using EUI-64

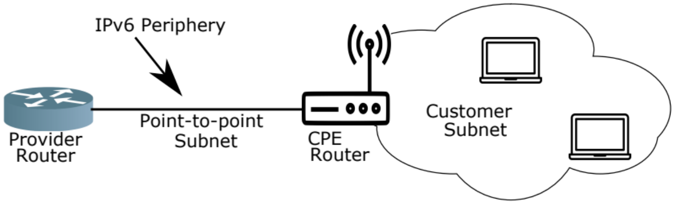


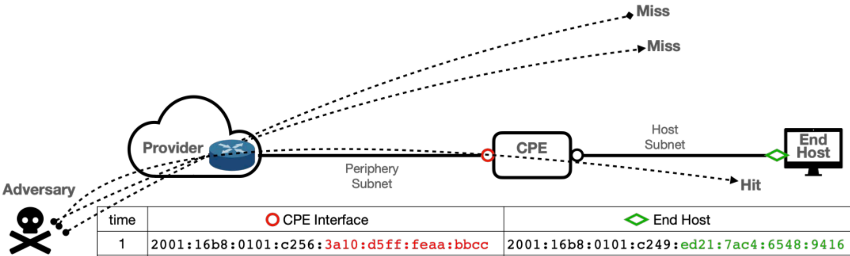
Figure 13: Common IPv6 architecture [16]

Is client tracking impossible?

- To prevent tracking based on the assigned prefix, providers often rotate their assignments

Is client tracking impossible?

- To prevent tracking based on the assigned prefix, providers often rotate their assignments
- But behavioral analysis of providers and CPE's using EUI-64 can be used to track prefixes [17]
- While clients can not be tracked, CPEs using EUI-64 identifiers can be actively found.



time	CPE Interface	End Host
1	2001:16b8:0101:c256:3a10:d5ff:feaa:bbcc	2001:16b8:0101:c249:ed21:7ac4:6548:9416
2	2001:16b8:0103:74fe:3a10:d5ff:feaa:bbcc	2001:16b8:0103:7421:42c1:02b5:5ff3:4bec
3	2001:16b8:0101:1f20:3a10:d5ff:feaa:bbcc	2001:16b8:0101:1529:ac1a:1e66:0801:c844

Figure 14: CPE with missing privacy extension [17]

Is client tracking impossible?

- In theory, testing all targets in a /48 is infeasible
- How can we reduce the search space and effectively track EUI-64 identifier?
  - Customers receive at least /64 prefixes and often larger
  - Providers often use only parts of their owned prefixes
  - Prefixes are often assigned at nibbles (e.g., /56, /60, /64)

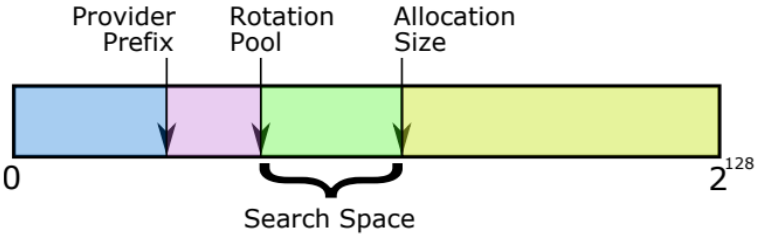


Figure 15: Limiting the search space to track IPv6 hosts: for a provider, infer the : i) size of the allocations to customers; and ii) range of prefixes used for rotation. [17]

Is client tracking impossible?

- A recent study reveals that high speed probing, behavioral analysis of providers and CPE's using EUI-64 can be used to track prefixes [17]
- Different allocation schemes can be found, e.g., /56 or /64 assignments

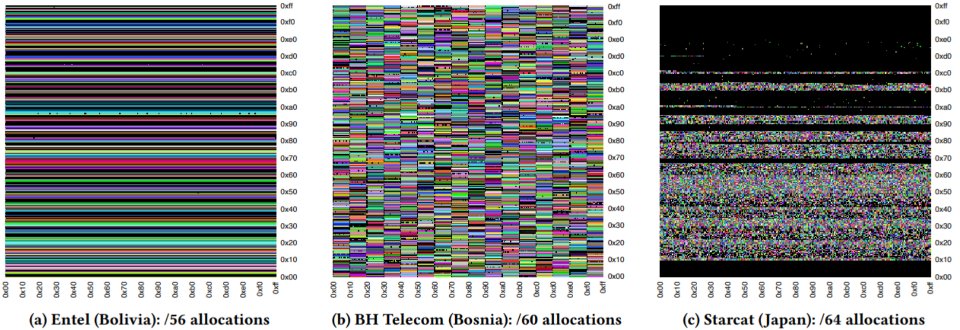


Figure 16: The y-axis of plots represents the 7^{th} byte of a probed address, while the x-axis denotes the 8^{th} byte; each pixel represents a probed /64 network. Each color represents a different responsive source address, while black indicates no response was received when probing to an address in that /64 network. [17]

Is client tracking impossible?

- Active scans can be used to identify specific CPEs even if providers rotate their assignments
- Based on a reduced search space, an adversary can effectively implement tracking

→ The tracking relies on CPEs using EUI-64

→ CPE manufacturers should change the device behavior

- The given paper resulted in a change of behavior within a large manufacturer
- For more details see the original work [17]

Internet Protocol Version 6

How to run out of Addresses

Based on a talk at RIPE77 from Benedikt Stockebrand [18]

Host-Density Ratio [19]: $HD = \frac{log_2(k)}{log_2(n)}$

- Address assignment efficiency
- Measured logarithmically
- k is the number of encoded addresses
- $log_2(k)$ is the number of required bits
- e.g. for 5 addresses, $log_2(5) \approx 2.3$ bits
- $log_2(n)$ is the number of overall address bits available

$\rightarrow$ Waste bits not Addresses

How to run out of Addresses
Inifinite Amount of Addresses

Total IPv6 Addresses:

- $2^{128} = 340282366920938463463374607431768211456 \approx 3.4 \times 10^{38}$

How to run out of Addresses

Inifinite Amount of Addresses

Total IPv6 Addresses:

- $2^{128} = 340282366920938463463374607431768211456 \approx 3.4 \times 10^{38}$

64 bits used for local area subnets:

- $2^{64} = 18446744073709551616 \approx 1.8 \times 10^{19}$
- Only $\approx 5.4^{-20}\%$ remaining

How to run out of Addresses
Multi-Layer Delegations

Each delegation requires 1 bit :

- IANA
- RIR
- LIR
- Company
- Branches
- Teams

$\rightarrow$ Waste 6 bits

How to run out of Addresses
Multi-Layer Delegations

Each delegation requires 1 nibble:

- IANA
- RIR
- LIR
- Company
- Branches
- Teams

$\rightarrow$ Waste 24 bits

How to run out of Addresses
Waste Bits

Excursion: Bit Magic in programming

- We have fixed pointers of size 32 bits
- We only reference 4 million items
- $\rightarrow log_2(4000000) \approx 22$ bits
- $\rightarrow$ 10 bits can be used to encode information to save memory

Application: Encode information into Addresses

- Example: A company with several branches in Germany wants to encode the postal code of each branch into addresses.
- Germany has 5 digits postal codes (00000-99999)
- $log_2(99999) \approx 17$ bits

How to run out of Addresses
Missing Regulations/Knowledge

A lot of addresses are simply lost due to missing regulations and knowledge about actual requirements.

- There are enough addresses, therefore I want as many as possible.
- I have no idea how many addresses I need for my company.
- I want more addresses than other companies.

How to run out of Addresses
What know?

It is not that bad yet!

- Theoretically, there are enough IPv6 addresses
- Early IPv4 deployments had similar problems
  - → Transition to CIDR
- Prefix delegation and address assignments hast to be done carefully

Internet Protocol Version 6

Internet Protocol Version 6

[1] S. Deering and R. Hinden, Internet Protocol, Version 6 (IPv6) Specification, http://tools.ietf.org/html/rfc1883, 1995.

[2] ——,Internet Protocol, Version 6 (IPv6) Specification, http://tools.ietf.org/html/rfc2460, 1998.

[3] D. S. E. Deering and B. Hinden, Internet Protocol, Version 6 (IPv6) Specification, RFC 8200, Jul. 2017. DOI: 10.17487/RFC8200. [Online]. Available: https://rfc-editor.org/rfc/rfc8200.txt.

[4] E. Nordmark, W. Simpson, and H. Soliman, Neighbor Discovery for IP version 6 (IPv6), http://tools.ietf.org/html/rfc4861, 2007.

[5] Q. Scheitle, O. Hohlfeld, J. Gamba, J. Jelten, T. Zimmermann, S. D. Strowes, and N. Vallina-Rodriguez, “A long way to the top: Significance, structure, and stability of internet top lists,” in Proceedings of the Internet Measurement Conference 2018, ACM, 2018, pp. 478–493.

[6] W. Rweyemamu, T. Lauinger, C. Wilson, W. Robertson, and E. Kirda, “Clustering and the weekend effect: Recommendations for the use of top domain lists in security research,” in International Conference on Passive and Active Network Measurement, Springer, 2019, pp. 161–177.

[7] F. Gont and T. Chown, “Network reconnaissance in ipv6 networks,” RFC 7707, 2016.

[8] T. Fiebig, K. Borgolte, S. Hao, C. Kruegel, and G. Vigna, “Something From Nothing (There): Collecting Global IPv6 Datasets From DNS,” PAM, 2017.

[9] S. Bortzmeyer and S. Huque, “NXDOMAIN: There Really Is Nothing Underneath,” RFC 8020, 2016.

[10] P. Foremski, D. Plonka, and A. Berger, “Entropy/ip: Uncovering structure in ipv6 addresses,” ACM IMC, 2016.

[11] A. Murdock, F. Li, P. Bramsen, Z. Durumeric, and V. Paxson, “Target Generation for Internet-wide IPv6 Scanning,” ACM IMC, 2017.

[12] O. Gasser, Q. Scheitle, S. Gebhard, and G. Carle, “Scanning the ipv6 internet: Towards a comprehensive hitlist,” in Proc. of 8th Int. Workshop on Traffic Monitoring and Analysis, Louvain-la-Neuve, Belgium, Apr. 2016.

[13] O. Gasser, Q. Scheitle, P. Foremski, Q. Lone, M. Korczyński, S. D. Strowes, L. Hendriks, and G. Carle, “Clusters in the expanse: Understanding and unbiasing ipv6 hitlists,” in Proceedings of the Internet Measurement Conference 2018, ACM, 2018, pp. 364–378.

[14] J. Zirngibl, L. Steger, P. Sattler, O. Gasser, and G. Carle, “Rusty Clusters? Dusting an IPv6 Research Foundation,”, Nice, France, 2022.

[15] D. Plonka and A. Berger, “Temporal and spatial classification of active ipv6 addresses,” in Proceedings of the 2015 Internet Measurement Conference, ser. IMC ’15, Tokyo, Japan: Association for Computing Machinery, 2015, 509–522, ISBN: 9781450338486.

[16] E. C. Rye and R. Beverly, “Discovering the ipv6 network periphery,” in Passive and Active Measurement, A. Sperotto, A. Dainotti, and B. Stiller, Eds., Cham: Springer International Publishing, 2020, pp. 3–18.

[17] E. Rye, R. Beverly, and K. C. Claffy, “Follow the scent: Defeating ipv6 prefix rotation privacy,” in Proceedings of the 21st ACM Internet Measurement Conference, ser. IMC ’21, Virtual Event: Association for Computing Machinery, 2021, 739–752, ISBN: 9781450391290.

[18] B. Stockebrand, The Art of Running Out of IPv6 Addresses, 2018. [Online]. Available: `https://ripe77.ripe.net/archives/video/2287/`.

[19] A. Durand and C. Huitema, The Host-Density Ratio for Address Assignment Efficiency: An update on the H ratio, https://tools.ietf.org/html/rfc3194, 2001.