ICANN Controlled Interruption IPv6 Research Study - Report 1

Author: John Dickinson, Sinodun IT
Date: 7th Oct 2025
Version: 1.3

Version	Date	Comments
0.1	26th June 2025	Preliminary draft report
1.0	8th July 2025	Updated report
1.1	17th July 2025	Include Mapped2 address as additional candidate for Stage 2 testing.
1.2	24th Sep 2025	Add links to other documents
1.3	7th Oct 2025	Add paragraph on expanding IPv6 loopback prefix

---

Table of Contents:

Introduction

This study is to identify one or more IPv6 addresses that may be suitable for use during Controlled Interruption. This study will involve a review of the IANA registries for IPv6 to produce a list of candidate addresses, and technical tests of popular end user operating systems to determine whether or not DNS lookups that respond with those addresses result in external network traffic.

This document is part of a set of documents covering the results of the study:

- ICANN Controlled Interruption IPv6 Research Study - Report 1 (this document)
- ICANN Controlled Interruption IPv6 Research Study - Report 2
- ICANN Controlled Interruption IPv6 Research Study - Summary Report

Controlled Interruption

Controlled Interruption is a phase in the establishment of a new generic top-level domain that is designed to address the risk of Name Collision [ICANN-NC]

During Controlled Interruption, certain DNS resource records are published at and below the gTLD name that are designed to interrupt resolution processes in such a way as to minimize any harm that arises from such interruption.

For example, A, MX, TXT and SRV resource records are published at the apex of the DNS zone, and as wildcards, that contain information intended to inform network administrators that a name collision has occurred:

```
example. IN A 127.0.53.53
example. IN MX 10 your-dns-needs-immediate-attention.example.
example. IN SRV 10 10 0 your-dns-needs-immediate-attention.example.
example. IN TXT ("Your DNS configuration needs immediate attention
                 see https://namecollisions.icann.org/")

* IN A 127.0.53.53
* IN MX 10 your-dns-needs-immediate-attention.example.
* IN SRV 10 10 0 your-dns-needs-immediate-attention.example.
* IN TXT ("Your DNS configuration needs immediate attention see
                     https://namecollisions.icann.org/")
```

Controlled Interruption uses A resource records containing the IPv4 address 127.0.53.53. This address is within the 127.0.0.0/8 prefix, which is used for "loopback" interfaces (see RFC 1122, Section 3.2.1.2). The use of this prefix ensures that any application that uses the result of a DNS query that returns this IPv4 address will not initiate a network connection. 127.0.53.53 is used instead of 127.0.0.1 so that if a user or administrator performs a web search for this address, they are more likely to be shown links to the Name Collision resources on ICANN's web site.

When the concept of Controlled Interruption was originally developed in 2014, no suitable equivalent address could be found for IPv6-enabled hosts. While this was noted as a limitation, at the time the impact was limited since fewer than 5% of internet hosts supported IPv6. In the intervening decade, IPv6 adoption has accelerated, and now more than 40% of all internet hosts have IPv6 connectivity. As a result, the lack of an IPv6 solution for Controlled Interruption has a much greater impact. Even though most clients are dual-stack, some clients may not query for both IPv4 and IPv6 addresses.

ICANNs preferred approach would be to use an IPv6 loopback address analogous to the existing IPv4 `127.0.53.53` address, however, IPv6 has only a single loopback address allocated (::1 [RFC 4291]). Two proposals within the IPv6 Operations working group at the IETF to assign a larger loopback prefix have not gained consensus and as a result there is still no direct equivalent IPv6 address available today. Should such an address become available in future ICANN would intend to utilize it for Controlled Interruption. In lieu of that, and with a recognition of the current impact of not publishing a AAAA record for Controlled Interruption, an alternative option that can be deployed as an interim solution is highly desirable.

Scope of work

The study shall carry out a review of all the registrations in the Internet Protocol Version 6 Address Space [IANA-IPv6AS] and IANA IPv6 Special-Purpose Address Registry registries, and any other relevant resources, to develop a list of [IANA-IPv6SPAR] candidate address prefixes.

A prefix may be considered a candidate if the specifications indicate that packets with a destination address within that prefix will remain local to the host (i.e., packets will not be transmitted outbound from the host). If no such prefixes are found, second-best options should also be considered, such as prefixes within the link-local scope which are unlikely to exist on the local network.

The consultant will then perform benchtop tests of popular end-user operating systems (including but not limited to Windows 10 and 11, and all not end-of-life versions of macOS, iOS, Android, Debian, and Red Hat Enterprise Linux) and applications such as browsers (including but not limited to Chrome, Firefox, Safari and Edge). Test systems must be freshly installed and up-to date, and configured with IPv6 connectivity only. Test systems should also be configurable as dual-stack in order to determine if system behaviour changes in case both IPv4 and IPv6 connectivity is available. The consultant will then test the candidate addresses using a range of browsers (including but not limited to Chrome, Firefox, Safari, and Edge, where available for the operating system), command-line tools (including but not limited to curl, ssh, netcat, openssl s_client), and popular applications (including those that act as clients even if they are categorized as servers, for example, HTTP proxy servers, and SMTP servers that relay email to other hosts) for those systems.

Deliverables

The study shall will produce a report in a mutually agreed format which includes:

1. A summary of the document review, which includes the key data points as to why specific IPv6 prefixes were selected/rejected as candidates;

2. A description of the methodology for performing benchtop tests, including equipment and software used and test procedures in sufficient detail to allow independent replication

3. Results of the tests indicating for each IPv6 address, operating system, and application, whether or not the tests resulted in network packets leaving the test system, and under what circumstances such network traffic occurred.

4. A clear recommendation to ICANN regarding what IPv6 address to choose for controlled interruption

This report constitutes the first of the above deliverables. The remaining deliverables are covered in ICANN Controlled Interruption IPv6 Research Study - Report 2 and a summary of the findings in a format suitable for public review are available in the draft Summary report.

Terminology

It is assumed that readers of this document are familiar with the notation of IPv6 addresses. In particular the collapsing of multiple 0's. We introduce the following terminology, largely lifted from various RFC's relating to IPv6 addressing.

- **node**: a device that implements IPv6.
- **router**: a node that forwards IPv6 packets not explicitly addressed to itself.
- **host**: any node that is not a router.
- **link**: a communication facility or medium over which nodes can communicate at the link layer, i.e., the layer immediately below IPv6. Examples are Ethernets (simple or bridged); PPP links; X.25, Frame Relay, or ATM networks; and internet-layer or higher-layer "tunnels", such as tunnels over IPv4 or IPv6 itself.
- **neighbors**: nodes attached to the same link.
- **interface**: a node's attachment to a link.
- **address**: an IPv6-layer identifier for an interface or a set of interfaces.
- **unicast**: An identifier for a single interface. A packet sent to a unicast address is delivered to the interface identified by that address.
- **anycast**: An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to an anycast address is delivered to one of the interfaces identified by that address (the "nearest" one, according to the routing protocols' measure of distance).
- **multicast**: An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to a multicast address is delivered to all interfaces identified by that address.
- **scope**: Scope has more than one meaning depending on its use. It can refer to
    - the scope of a multicast address see [RFC4291].
    - the sin6_scope_id in a sockadder_in6 struct in the OS kernel (see [MANIPv6] and [FREEBSD-DH]. It is used to differentiate link-local or node-local addresses which can be the same on multiple interfaces.

IPv6 Standards and IANA Registries Overview

The following discussion reviews the most recent versions of IPv6 RFCs (the ‘base’ RFCs) and those that update them. Obsoleted RFCs are listed below for completeness but the following sections use the ‘base’ version taken from the obsoleting RFC. The full set of related documents is given below.

Base RFC	Name	Updated by	Obsoletes
RFC8200	Internet Protocol, Version 6 (IPv6) Specification	RFC9673	RFC2460 RFC1883
RFC4291	IP Version 6 Addressing Architecture	RFC5952 RFC6052 RFC7136 RFC7346 RFC7371 RFC8064	RFC3513 RFC2373 RFC1884

RFC8200: Internet Protocol, Version 6 (IPv6) Specification

Internet Standard RFC8200 is the **Internet Protocol, Version 6 (IPv6) Specification**. This document mostly describes the IPv6 header and doesn't concern itself with addressing details. It does introduce some useful terminology which is also used here.

It is updated by RFC9673 **IPv6 Hop-by-Hop Options Processing Procedures** which relates only to header handling, not addressing. Note that the obsoleted RFC2460 is updated by multiple drafts, none of which relate to addressing.

RFC4291: IP Version 6 Addressing Architecture

RFC4291 document specifies the **IP Version 6 Addressing Architecture**. It also gives some useful terminology which is used here.

Most importantly, it defines the ***IANA Internet Protocol Version 6 Address Space*** registry [IANA-IPv6AS], relevant parts of which are shown in the table below. (As a minor note, the registry contents are first referenced in RFC3513 and not directly reproduced in RFC4291)

Appendix A lists the various updates to RFC4291 which have been reviewed and deemed to have no direct relevance to this study. The exception below is referenced in the following sections:

- RFC7346 **IPv6 Multicast Address Scopes**

NOTES:

1. [IANA-IPv6SPAR] T:he table below makes reference to the IANA registry **iana-ipv6-special-registry** i.e. the **IANA IPv6 Special-Purpose Address Registry** for parts of the following prefixes ::/8, 100::/8, 4000::/3, fc00::/7 and fe80::/10 which are discussed in the next section

2. [IANA-IPv6UAA]: The table makes reference to the IANA registry **ipv6-unicast-address-assignments** i.e. the **IPv6 Global Unicast Address Assignments**
3. [IANA-IPv6MASR]: The table makes reference to the IANA registry **ipv6-mulitcast-addresses** i.e the **IPv6 Multicast Address Space Registry** which is reviewed for candidates for this study. RFC4291 deprecated ::/96 for use as IPv4-compatible IPv6 addresses.
4. RFC4048 deprecated 200::/7.

IANA IPv6 Address Space

Prefix	Allocation	RFC	Notes
::/8	Reserved by IETF	[RFC3513] [RFC4291]	This range has been partially allocated. See [IANA registry iana-ipv6-special-registry] for details. ::/96, formerly defined as the "IPv4-compatible IPv6 address" prefix, was deprecated by [RFC4291].
100::/8	Reserved by IETF	[RFC3513] [RFC4291]	This range has been partially allocated. See [IANA registry iana-ipv6-special-registry] for details.
200::/7	Reserved by IETF	[RFC4048]	Deprecated as of December 2004 [RFC4048]. Formerly an OSI NSAP-mapped prefix set [RFC4548].
400::/6	Reserved by IETF	[RFC3513] [RFC4291]	
2000::/3	Global Unicast	[RFC3513] [RFC4291]	The IPv6 Unicast space encompasses the entire IPv6 address range with the exception of ff00::/8, per [RFC4291]. IANA unicast address assignments are currently limited to the IPv6 unicast address range of 2000::/3. IANA assignments from this block are registered in [IANA registry ipv6-unicast-address-assignments].
4000::/3	Reserved by IETF	[RFC3513] [RFC4291]	This range has been partially allocated. See [IANA registry iana-ipv6-special-registry] for details. 5f00::/8 (with 3ffe::/16, as noted at [IANA registry ipv6-unicast-address-assignments]) was used for the 6bone, but returned [RFC5156].
fc00::/7	Unique Local Unicast	[RFC4193]	See [IANA registry iana-ipv6-special-registry] for details.
fe00::/9	Reserved by IETF	[RFC3513] [RFC4291]	

Prefix	Allocation	RFC	Notes
fe80::/10	Link-Scoped Unicast	[RFC3513] [RFC4291]	See [IANA registry iana-ipv6-special-registry] for details.
fec0::/10	Reserved by IETF	[RFC3879]	Deprecated by [RFC3879] in September 2004. Formerly a Site-Local scoped address prefix.
ff00::/8	Multicast	[RFC3513] [RFC4291]	See [IANA registry ipv6-multicast-addresses] for details.

Key:

- Blue: Global Unicast
- Grey: Unique Local Unicast
- Green: Link-Scoped Unicast
- Red : Multicast

IANA IPv6 Special-Purpose Address Registry

Address Block	Name	RFC	Source	Destination	Forwardable	Globally Reachable	Reserved-by-Protocol
::1/128	Loopback Address	[RFC4291]	False	False	False	False	True
::/128	Unspecified Address	[RFC4291]	True	False	False	False	True
::ffff:0:0/96	IPv4-mapped Address	[RFC4291]	False	False	False	False	True
64:ff9b::/96	IPv4-IPv6 Translat.	[RFC6052]	True	True	True	True	False
64:ff9b:1::/48	IPv4-IPv6 Translat.	[RFC8215]	True	True	True	False	False
100::/64	Discard-Only Address Block	[RFC6666]	True	True	True	False	False
100:0:0:1::/64	Dummy IPv6 Prefix	[RFC9780]	True	False	False	False	False
2001::/23	IETF Protocol Assignments	[RFC2928]	False [1]	False [1]	False [1]	False [1]	False
2001::/32	TEREDO	[RFC4380] [RFC8190]	True	True	True	N/A [2]	False
2001:1::1/128	Port Control Protocol Anycast	[RFC7723]	True	True	True	True	False
2001:1::2/128	Traversal Using Relays around NAT Anycast	[RFC8155]	True	True	True	True	False
2001:1::3/128	DNS-SD Service Registration Protocol Anycast	[RFC9665]	True	True	True	True	False
2001:2::/48	Benchmarking	[RFC5180] [RFC Errata 1752]	True	True	True	False	False
2001:3::/32	AMT	[RFC7450]	True	True	True	True	False
2001:4:112::/48	AS112-v6	[RFC7535]	True	True	True	True	False
2001:10::/28	Deprecated (previously ORCHID)	[RFC4843]					

Address Block	Name	RFC	Source	Destination	Forwardable	Globally Reachable	Reserved-by-Protocol
2001:20::/28	ORCHIDv2	[RFC7343]	True	True	True	True	False
2001:30::/28	Drone Remote ID Protocol Entity Tags (DETs) Prefix	[RFC9374]	True	True	True	True	False
2001:db8::/32	Documentation	[RFC3849]	False	False	False	False	False
2002::/16 [3]	6to4	[RFC3056]	True	True	True	N/A [3]	False
2620:4f:8000::/48	Direct Delegation AS112 Service	[RFC7534]	True	True	True	True	False
3fff::/20	Documentation	[RFC9637]	False	False	False	False	False
5f00::/16	Segment Routing (SRv6) SIDs	[RFC9602]	True	True	True	False	False
fc00::/7	Unique-Local	[RFC4193] [RFC8190]	True	True	True	False [4]	False
fe80::/10	Link-Local Unicast	[RFC4291]	True	True	False	False	True

[1] Unless allowed by a more specific allocation.
[2] See Section 5 of [RFC4380] for details.
[3] See [RFC3056] for details.
[4] See [RFC4193] for more details on the routability of Unique-Local addresses. The Unique-Local prefix is drawn from the IPv6 Global Unicast Address range, but is specified as not globally routed.

For context, the definition of a destination address is the following:

- Destination - A boolean value indicating whether an address from the allocated special-purpose address block is valid when used as the destination address of an IP datagram that transits **two devices.**

IPv6 Multicast Address Space Registry

This includes several registries:

- IPv6 Multicast Address Scopes
- Node-Local Scope Multicast Addresses
- Link-Local Scope Multicast Addresses
- Site-Local Scope Multicast Addresses
- Variable Scope Multicast Addresses
- Unicast-based (Including SSM) Multicast Group IDs

Of particular interest to this study are the Node-Local Scope Multicast Addresses, Link-Local Scope Multicast Addresses since they include addresses with an 'Interface-Local scope' and also addresses with a 'Link-Local scope'.

Node-Local Scope Multicast Addresses

Address(es)	Description	Reference
FF01:0:0:0:0:0:0:1	All Nodes Address	[RFC4291]
FF01:0:0:0:0:0:0:2	All Routers Address	[RFC4291]

Link-Local Scope Multicast Addresses

Address(es)	Description	Reference
FF02:0:0:0:0:0:0:1	All Nodes Address	[RFC4291]
FF02:0:0:0:0:0:0:2	All Routers Address	[RFC4291]

Candidate Prefixes Review

Discussion of IPv6 Address Assignment Characteristics

We briefly note here the definitions of the loopback addresses to provide context on the characteristics of candidate prefixes

- The unicast address ::1 is called the **loopback address**. It may be used by a node to send an IPv6 packet to itself. It must not be assigned to any physical interface.

  The loopback address must not be used as the source address in IPv6 packets that are sent outside of a single node. An IPv6 packet with a destination address of loopback must never be sent outside of a single node and must never be forwarded by an IPv6 router. A packet received on an interface with a destination address of loopback must be dropped.

Requirements for first class candidate prefix

1. It must be capable of being stored in a zone file and returned by a DNS server in a AAAA resource record as defined in RFC3596. This means it cannot contain characters that are invalid in AAAA records, such as % which is used to indicate a specific interface.

2. It must not cause user software (command line tools, browser, and other applications) to fail in an inelegant way (i.e. crashing/freezing) when provided via DNS.

3. It must not cause OS libraries or network stacks to fail in an inelegant way when used as the destination of a network connection.

4. It must not cause a successful connection (logical or physical) to be made.

5. Its use *must not* result in packets exiting the host

Requirements for second class candidate prefixes

Requirement 4 is updated and requirement 5 is added:

4. Its use *should not* result in packets exiting the host
5. Its use *must not* result in packets exiting the network segment they are used on

Elegant failure modes can include timeouts, recognised software errors and recognised OS errors.

Non-requirements

1. That the behaviour of any application is identical to that observed when resolving the Controlled interruption A record (127.0.0.1). The behaviour may be similar but, for example, where the A record results in a timeout, the AAAA record may result in an error message.
2. That any direct indication is made to the user of anything other than a generic error (i.e. no mention of Controlled Interruption is required in the error message).

Potential first class candidates

IANA IPv6 Special-Purpose Address Registry

The subset of Address Blocks from the IANA IPv6 Special-Purpose Address Registry for which Destination is false (and therefore should share behaviour with the loopback address) is:

	Short name (for this study)	Address Block	Name	RFC	Source	Destination	Forwardable	Globally Reachable	Reserved-by-Protocol
1	Loopback	::1/128	Loopback Address	[RFC4291]	False	False	False	False	True
2	Unspecified	::/128	Unspecified Address	[RFC4291]	True	False	False	False	True
3	Mapped	::ffff:0:0/96	IPv4-mapped Address	[RFC4291]	False	False	False	False	True
4	Dummy	100:0:0:1::/64	Dummy IPv6 Prefix	[RFC9780]	True	False	False	False	False

5	Protocol	2001::/23	IETF Protocol Assignments	[RFC2928]	False [1]	False [1]	False [1]	False [1]	False
6	Doc1	2001:db8::/32	Documentation	[RFC3849]	False	False	False	False	False
7	Doc2	3fff::/20	Documentation	[RFC9637]	False	False	False	False	False

[1] Unless allowed by a more specific allocation.

Each of the candidates is reviewed below:

1. Loopback: Discounted since it is the only loopback address available. Multiple applications are known to listen on various ports on this address, for example most linux distributions have an ssh client that listens on all addresses (including both the IPv4 and IPv6 the loopback addresses).
2. Unspecified: Discounted by virtue of its definition in RFC4291
3. Mapped: Potential candidate
4. Dummy: Potential candidate
5. Protocol**:** Discounted - assignments to RIRs via the IPv6 Global Unicast Address Assignments registry.
6. Doc1 and Doc2: For the Documentation Address blocks, RFC9637 makes the explicit statement:

   "4. Filtering and Appropriate Use

   > *Documentation prefixes are for the use of relaying configuration and documentation examples, and as such, they MUST NOT be used for actual traffic, MUST NOT be globally advertised, and SHOULD NOT be used internally for routed production traffic or other connectivity. Documentation prefixes should be considered bogon [BOGON] and filtered in routing advertisements as appropriate."*

   While the designation as a "destination=False" for these addresses is unlikely to be implemented in host OS's (particularly for the most recent assignment), we intend to include these address blocks as potential first class candidates for completeness.

IPv6 Multicast Address Space Registry

RFC4291 says:

"Interface-Local scope spans only a single interface on a node and is useful only for loopback transmission of multicast."

…*"A host is required to recognize the following addresses as identifying itself:*

o …

o The loopback address.

o ***The All-Nodes multicast addresses defined in Section 2.7.1.****"…*

Section 2.7.1 says:

" All Nodes Addresses: FF01:0:0:0:0:0:0:1
FF02:0:0:0:0:0:0:1
The above multicast addresses identify the group of all IPv6 nodes, within scope 1 (interface-local) or 2 (link-local)."
…
"All Routers Addresses: FF01:0:0:0:0:0:0:2
FF02:0:0:0:0:0:0:2
FF05:0:0:0:0:0:0:2

The above multicast addresses identify the group of all IPv6 routers, within scope 1 (interface-local), 2 (link-local), or 5 (site-local)."

Given the definition of Interface-Local scope here, we include both the Interface-local addresses as potential candidates. It is noted that their definitions would usually imply that packets sent to these addresses are qualified with an interface identifier, we include them as their usage without an interface identifier may fail in an acceptable way.

Full list of potential first class candidates

Note the IPv4 address 127.0.53.53 represented in (a partial) IPv6 presentation format is 7f00:3535.

TABLE 1

Test Name	Address Block/Address	Proposed Test Address(es)	
Mapped	::ffff:0:0/96	::ffff::0:1,	::ffff::7f00:3535
Dummy	100:0:0:1::/64	100:0:0:1::1,	100::1:0:0:7f00:3535
DocumentationA	2001:db8::/32	2001:db8::1,	2001:db8::7f00:3535
DocumentationB	3fff::/20	3fff::1,	3fff::7f00:3535
MultNodeInterface	ff01::1	ff01::1	
MultNodeLink	ff02::2	ff02::2	
MultRouterInterface	ff01::2	ff01::2	

Potential second class candidates

At this point we only consider a few potential second class candidates which seem worthwhile also testing. Should all the potential candidates fail full testing, we will return to a more indepth review of other candidates.

::7f00:3535

This address is in the reserved but only partially allocated ::/8 prefix, and does not fall within any of the currently allocated address blocks. As such its behaviour is undefined but is included as a potential candidate for completeness

Unique Local Unicast

Multiple objections to allocation of a larger loopback prefix for IPv6 were based on the premise that Unique Local Unicast addresses can be used for local testing. Indeed provisioning a ULA on the loopback interface does indeed provide an additional address to behave as the IPv6 loopback address does. However this requires manual configuration on the host, and can additionally require an interface identifier when used to send traffic and therefore does not fit the requirements for CI.

However testing of unconfigured unique local unicast should be considered as they may fail to send any traffic in an acceptable way.

Name	Address Block/Address	Proposed Test Address(es)
UniqueLocalUnicast	fc00::/10	fc00::1, fe80::ffff:7f00:3535

Link-Local Unicast

Whilst the link-local unicast address block is defined as fe80::/10, in practice today it is routine to use only the lower /64 of this block. Additionally the address is only meaningful when qualified with an address identifier. As such, we propose to include both fe80::/1 and fe80::7f00:5353 as potential second class candidates.

Full list of potential second class candidates

TABLE 2

Test Name	Address Block/Address	Proposed Test Address(es)
Unallocated	::7f00::3535	::0:0:7f00:3535
UniqueLocalUnicast	fc00::/10	fc00::1, fc00::7f00:3535
LinkLocalUnicast	fe80::/10	fe80::1, fe80::7f00:3535

Basic automated testing

The candidate test names and IPv6 addresses are taken from tables 1 and 2 above. The names used here include uppercase letters for readability but were all lowercase before use in the tests. Some additional names/addresses are included to provide a baseline for the test e.g. icann.org and the IPv6 loopback address.

For the purposes of an initial survey of the candidates, a basic test script was developed to do the following:

- Configure a test recursive resolver/authoritative pair of nameservers where
    - The test recursive resolver performs normal recursive resolution for all queries except the list of test TLD names which are sent to the test authoritative server
    - The test authoritative servers has custom test zones provisioned for each test TLD name with the ICANN Controlled interruption contents and the test IPv6 address in the AAAA record
- Configure a test host OS as dual stack
- Configure the test host system resolver to point to this test recursive resolver
- Read the set of test names/IPv6 addresses. For each test name:
    - Loop over a list of commands:
        - Flush the system DNS cache and any application DNS cache
        - Perform `dig` commands to ensure resolution is returning the expected A/AAAA record for each test TLD name
        - Capture traffic using tcpdump to monitor DNS resolution and also monitor traffic on both the loopback interface and the network interface
        - Launch the command to access the test name prepended by an additional label e.g.

          ```
          ssh a.test_name
          ```

          and log the output. An additional label is prepended to model the real-world case where a lookup for a single label is highly unlikely. This also results in a response matched from the wildcard to avoid peculiarities associated with resolving a request for an A/AAAA record for a single label on linux (see notes below).
        - Inspect the traffic to confirm DNS resolution occurred and the test A/AAAA record was returned
        - Inspect the traffic for packets containing the test A/AAAA address on both interfaces
- Generate a report of observed traffic for each test name

This test methodology is a simple black box approach primarily intended to rule out candidates where traffic to the test address is observed being transmitted off the host after DNS resolution of the test name for all the commands included in this test suite.

This testing provides only a high level report of the failure modes of those test addresses where traffic is not observed to leave the host. Technical consideration and consequences of those failure modes and also any restriction imposed by specifications but not implemented are out of scope for these addresses at this stage.

Results

- The tests were run on a fresh install of Ubuntu 25.04, a slightly customised Ubuntu 24.04 and also a macOS Sequoia 15.5 desktop.
- The commands tested were `ssh`, `curl` and either Chrome or Firefox browser.
- The results were consistent across all platforms and across all 3 commands and so are summarised below.
- For the Mapped2 address it is noted that the local traffic seem was IPv4 traffic to 127.0.53.53

Test Name	Test Address	Local traffic seen	Network traffic seen
REFERENCE ADDRESSES			
icann.org	-	N	Y
IPv4CI	127.0.53.53	Y	N
IPv6Localhost	::1	Y	N
TEST ADDRESSES			
Mapped1	::ffff:0:1	N	N
Mapped2	::ffff:7f00:3535	Y *	N
Dummy1	100:0:0:1::1	N	Y
Dummy2	100::1:0:0:7f00:3535	N	Y
DocumentationA1	2001:db8::1	N	Y
DocumentationA2	2001:db8::7f00:3535	N	Y
DocumentationB1	3fff::1	N	Y
DocumentationB2	3fff::7f00:3535	N	Y
MultNodeInterface	ff01::1	N	N
MultNodeLink	ff01::2	N	N
MultRouterInterface	ff02::1	N	N
Unallocated	::0:0:7f00:3535	N	Y
UniqueLocalUnicast1	fc00::1	N	Y
UniqueLocalUnicast2	fc00::7f00:3535	N	Y
LinkLocalUnicast1	fe80::1	N	N

LinkLocalUnicast2	fe80::7f00:3535	N	N

Conclusions

- In our basic testing, the majority of the candidate prefixes resulted in traffic being sent off the host machine over the external network interface. Thus violating the requirements for first class candidates.
- The exceptions were
  - The mapped address Mapped2 i.e :ffff:7f00:3535, which maps to an IPv4 address of 127.0.53.53. For the subset of testing scenarios here, all OS's and apps initiated IPv4 connections to 127.0.53.53 after resolving this name.
  - The multicast addresses: all of those tested generated no network traffic, presumably because of their defined scope.
    - `curl` reported '`Couldn't connect to server`' in all cases.
    - The failure mode reported by `ssh` was either
      - `Address family not supported by protocol family` (macOS)
      - `Network is unreachable` (Linux)
  - The link local unicast address: both addresses tested generated no network traffic, presumably because no interface identifier is specified.
    - Again, `curl` reported '`Couldn't connect to server`' in all cases.
    - The failure mode reported by `ssh` was either
      - `No route to host` (macOS)
      - `Invalid argument` (Linux)
- All the above addresses spaces will be further tested in the next stage of the project
  - We propose the more detailed testing of the second phase of the study should investigate not only the 3 multicast address test here but some additional addresses in the same prefix space, and similarly for the link local unicast address block.
  - For the Mapped2 address, we need to investigate the behaviour on both dual stack and IPv6 only hosts. The definition of an IPv6 only host will need clarification since some may still have an active IPV4 loopback interface, while others may not. The acceptability of initiating an IPv4 connection after resolving a AAAA record also needs some consideration.
- If necessary the other addresses would be tested in more detail as second class candidates

Candidate address prefixes for Stage 2 testing

Test Name	Test Address	Local traffic seen	Network traffic seen
Mapped2	::ffff:7f00:3535	Y (IPv4 traffic to 127.0.53.53)	N
MultNodeInterface	ff01::1	N	N
MultNodeLink	ff01::2	N	N
MultRouterInterface	ff02::1	N	N
LinkLocalUnicast1	fe80::1	N	N
LinkLocalUnicast2	fe80::7f00:3535	N	N

Notes on testing

After a quick test it appears that systemd-resolved refuses A and AAAA queries for single labels

```
$ dig com79 a

; <<>> DiG 9.20.4-3ubuntu1.1-Ubuntu <<>> com79 a
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 49333
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;com79.                        IN    A

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Wed Jun 25 11:37:36 BST 2025
;; MSG SIZE  rcvd: 34
```

A `tcpdump` of this shows

```
11:37:36.820637 lo    In  IP 127.0.0.1.54057 > 127.0.0.53.53: 49333+ [1au]
A? com79. (46)
11:37:36.820835 lo    In  IP 127.0.0.53.53 > 127.0.0.1.54057: 49333
```

```
Refused* 0/0/1 (34)
```

It works fine for NS queries

```
$ dig com79 ns

; <<>> DiG 9.20.4-3ubuntu1.1-Ubuntu <<>> com79 ns
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 43708
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;com79.                       IN    NS

;; AUTHORITY SECTION:
.                 3510  IN    SOA   a.root-servers.net.
nstld.verisign-grs.com. 2025062500 1800 900 604800 86400

;; Query time: 93 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Wed Jun 25 11:37:22 BST 2025
;; MSG SIZE  rcvd: 109
```

A `tcpdump` looks like this

```
11:37:22.750293 lo    In  IP 127.0.0.1.39769 > 127.0.0.53.53: 43708+ [1au]
NS? com79. (46)
11:37:22.750595 enp1s0 Out IP 192.168.15.37.52490 > 192.168.12.120.53:
19311+ [1au] NS? com79. (34)
11:37:22.843178 enp1s0 In  IP 192.168.12.120.53 > 192.168.15.37.52490:
19311 NXDomain 0/1/1 (109)
11:37:22.843744 lo    In  IP 127.0.0.53.53 > 127.0.0.1.39769: 43708
NXDomain 0/1/1 (109)
```

As shown above this could impact Controlled Interruption for IPv4 as well.

Queries for A and AAAA at a lower level work fine.

```
$ dig a.linklocalunicast aaaa
```

```
; <<>> DiG 9.20.4-3ubuntu1.1-Ubuntu <<>> a.linklocalunicast aaaa
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3754
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;a.linklocalunicast.          IN    AAAA

;; ANSWER SECTION:
a.linklocalunicast.     3600  IN    AAAA  fe80::53:53

;; Query time: 4 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Wed Jun 25 11:47:38 BST 2025
;; MSG SIZE  rcvd: 75
```

Changelog

Changes in v1.1

1. Due to 2 bugs in our testing scripts found during review of Stage 2 testing, we have revised the results relating to the Mapped2 address (::ffff:7A00.3535) and now include this as a candidate for the Stage 2 testing (it causes apps to initiate an IPv4 connection to 127.0.53.53 on the loopback interface)
2. Clarify testing performed on dual stack hosts
3. Add table of full set of candidate prefixes for Stage 2 testing

Changes in v1.0:

1. Clarify why the IPv6 loopback address is discounted as a candidate.
2. Minor updates to requirements section.
3. Correct the original LinkLocalUnicast address - 2 are now specified and tested.
4. Minor editorial clarifications to test methodology.
5. Updates and improvements were made to our test script, but did not affect the results. However this removed the transient error observed with Firefox in a few scenarios.
6. Add a brief discussion of failure modes to the Conclusions section.
7. Editorial updates to references and References section.
8. Added version control table and changelog section.

Appendix A

The following list RFCs which update RFC4291 but which are not of interest here:

1. [RFC6052] IPv6 Addressing of IPv4/IPv6 Translators.
2. [RFC5952] A Recommendation for IPv6 Address Text Representation.
3. [RFC7136] Significance of IPv6 Interface.
4. [RFC7346] IPv6 Multicast Address Scopes.
5. [RFC7371] Updates to the IPv6 Multicast Addressing Architecture.
6. [RFC8064] Recommendation on Stable IPv6 Interface Identifiers.

References

RFC and Interned drafts in order of RFC number

1. RFC1752 Scott O. Bradner and Allison J. Mankin. The Recommendation for the IP Next Generation Protocol. RFC 1752. Jan. 1995. doi: 10.17487/RFC1752. url: https://www.rfc-editor.org/info/rfc1752.
2. RFC2928 Dr. Steve E. Deering, Robert L. Fink, Tony L. Hain, and Bob Hinden. Initial IPv6 Sub-TLA ID Assignments. RFC 2928. Sept. 2000. doi: 10.17487/RFC2928. url:https://www.rfc-editor.org/info/rfc2928.
3. RFC3056 Brian E. Carpenter and Keith Moore. Connection of IPv6 Domains via IPv4 Clouds. RFC 3056. Feb. 2001. doi: 10.17487/RFC3056. url: https://www.rfc-editor.org/info/rfc3056.
4. RFC3596 Vladimir Ksinant, Christian Huitema, Dr. Susan Thomson, and Mohsen Souissi. DNS Extensions to Support IP Version 6. RFC 3596. Oct. 2003. doi: 10.17487/RFC3596. url:https://www.rfc-editor.org/info/rfc3596.
5. RFC3849 Geoff Huston, Anne Lord, and Dr. Philip F. Smith. IPv6 Address Prefix Reserved for Documentation. RFC 3849. July 2004. doi: 10.17487/RFC3849. url: https://www.rfc-editor.org/info/rfc3849.
6. RFC4048 Brian E. Carpenter. RFC 1888 Is Obsolete. RFC 4048. Apr. 2005. doi: 10.17487/RFC4048. url: https://www.rfc-editor.org/info/rfc4048.
7. RFC4291 Dr. Steve E. Deering and Bob Hinden. IP Version 6 Addressing Architecture. RFC 4291. Feb. 2006. doi: 10.17487/RFC4291. url: https://www.rfc-editor.org/info/rfc4291.
8. RFC4380 Christian Huitema. Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs). RFC 4380. Feb. 2006. doi: 10.17487/RFC4380. url: https://www.rfc-editor.org/info/rfc4380.
9. RFC5180 Diego Dugatkin, Ahmed Hamza, Gunter Van de Velde, and Chip Popoviciu. IPv6 Benchmarking Methodology for Network Interconnect Devices. RFC 5180. May 2008. doi: 10.17487/RFC5180. url: https://www.rfc-editor.org/info/rfc5180.
10. RFC5952 Seiichi Kawamura and Masanobu Kawashima. A Recommendation for IPv6 Address Text Representation. RFC 5952. Aug. 2010. doi: 10.17487/RFC5952. url: https://www.rfc-editor.org/info/rfc5952.

11. RFC6052 Xing Li, Mohamed Boucadair, Christian Huitema, Marcelo Bagnulo, and Congxiao Bao. IPv6 Addressing of IPv4/IPv6 Translators. RFC 6052. Oct. 2010. doi: 10.17487/RFC6052. url: https://www.rfc-editor.org/info/rfc6052.
12. RFC6666 Nick Hilliard and David Freedman. A Discard Prefix for IPv6. RFC 6666. Aug. 2012. doi: 10.17487/RFC6666. url: https://www.rfc-editor.org/info/rfc6666.
13. RFC6890 Michelle Cotton, Leo Vegoda, Ron Bonica, and Brian Haberman. Special-Purpose IP Address Registries. RFC 6890. Apr. 2013. doi: 10.17487/RFC6890. url: https://www.rfc-editor.org/info/rfc6890.
14. RFC7136 Brian E. Carpenter and Sheng Jiang. Significance of IPv6 Interface Identifiers. RFC 7136. Feb. 2014. doi: 10.17487/RFC7136. url: https://www.rfc-editor.org/info/rfc7136.
15. RFC7343 Julien Laganier and Francis Dupont. An IPv6 Prefix for Overlay Routable Cryptographic Hash Identifiers Version 2 (ORCHIDv2). RFC 7343. Sept. 2014. doi: 10.17487/RFC7343. url: https://www.rfc-editor.org/info/rfc7343.
16. RFC7346 Ralph Droms. IPv6 Multicast Address Scopes. RFC 7346. Aug. 2014. doi: 10.17487/RFC7346. url: https://www.rfc-editor.org/info/rfc7346.
17. RFC7371 Mohamed Boucadair and Stig Venaas. Updates to the IPv6 Multicast Addressing Architecture. RFC 7371. Sept. 2014. doi: 10.17487/RFC7371. url: https://www.rfc-editor.org/info/rfc7371.
18. RFC7534 Joe Abley and William F. Maton. AS112 Nameserver Operations. RFC 7534. May 2015. doi: 10.17487/RFC7534. url: https://www.rfc-editor.org/info/rfc7534.
19. RFC7723 Sebastian Kiesel and Reinaldo Penno. Port Control Protocol (PCP) Anycast Addresses. RFC 7723. Jan. 2016. doi: 10.17487/RFC7723. url: https://www.rfc-editor.org/info/rfc7723.
20. RFC8064 Fernando Gont, Alissa Cooper, Dave Thaler, and Will (Shucheng) LIU. Recommendation on Stable IPv6 Interface Identifiers. RFC 8064. Feb. 2017. doi: 10.17487/RFC8064. url: https://www.rfc-editor.org/info/rfc8064.
21. RFC8155 Prashanth Patil, Tirumaleswar Reddy.K, and Dan Wing. Traversal Using Relays around NAT (TURN) Server Auto Discovery. RFC 8155. Apr. 2017. doi: 10.17487/RFC8155. url: https://www.rfc-editor.org/info/rfc8155.
22. RFC8200 Dr. Steve E. Deering and Bob Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 8200. July 2017. doi: 10.17487/RFC8200. url: https://www.rfc-editor.org/info/rfc8200.
23. RFC8215 Tore Anderson. Local-Use IPv4/IPv6 Translation Prefix. RFC 8215. Aug. 2017. doi: 10.17487/RFC8215. url: https://www.rfc-editor.org/info/rfc8215.
24. RFC9374 Robert Moskowitz, Stuart W. Card, Adam Wiethuechter, and Andrei Gurtov. DRIP Entity Tag (DET) for Unmanned Aircraft System Remote ID (UAS RID). RFC 9374. Mar. 2023. doi: 10.17487/RFC9374. url: https://www.rfc-editor.org/info/rfc9374.
25. RFC9602 Suresh Krishnan. Segment Routing over IPv6 (SRv6) Segment Identifiers in the IPv6 Addressing Architecture. RFC 9602. Oct. 2024. doi: 10.17487/RFC9602. url: https://www.rfc-editor.org/info/rfc9602.
26. RFC9673 Bob Hinden and Gorry Fairhurst. IPv6 Hop-by-Hop Options Processing Procedures. RFC 9673. Oct. 2024. doi: 10.17487/RFC9673. url: https://www.rfc-editor.org/info/rfc9673.
27. RFC9637 Geoff Huston and Nick Buraglio. Expanding the IPv6 Documentation Space. RFC 9637. Aug. 2024. doi: 10.17487/RFC9637. url: https://www.rfc-editor.org/info/rfc9637.

28. RFC9780 Greg Mirsky, Gyan Mishra, and Donald E. Eastlake 3rd. Bidirectional Forwarding Detection (BFD) for Multipoint Networks over Point-to-Multipoint MPLS Label Switched Paths (LSPs). RFC 9780. May 2025. doi: 10.17487/RFC9780. url: https://www.rfc-editor.org/info/rfc9780.

Other references

1. MANIPv6 url: https://man7.org/linux/man-pages/man7/ipv6.7.html.
2. FREEBSD-DH url: https://docs.freebsd.org/en/books/developers-handbook/ipv6/#ipv6-scope-index.
3. IANA-IPv6SPAR IANA IPv6 Special-Purpose Address Registry. url: https://www.iana.org/assignments/iana-ipv6-special-registry/iana-ipv6-special-registry.xhtml.
4. IANA-IPv6AS Internet Protocol Version 6 Address Space. url: https://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xhtml.
5. IANA-IPv6-GUAA IPv6 Global Unicast Address Assignments. url: https://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xhtml.
6. IANA-IPv6MA IPv6 Multicast Address Space Registry. url: https://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xhtml.
7. ICANN-NC Name Collision. url: https://www.icann.org/resources/pages/namecollision-2013-12-06-en.