

Unintended Operational Issues With ULA

Nick Buraglio, ESnet
Chris Cummings, ESnet
Russ White, Juniper Networks

IETF 114

26-July-2022

Current ULA standards present unintended consequences

TL;DR

- **This draft is intended to identify and codify existing issues**
- **We are not attempting to propose any solutions**
- The behavior of ULA addressing as defined by [RFC6724] is preferred below legacy IPv4 addressing
- ULA deployments using de facto best practices for DNS (A and AAAA for a resource) will not, generally, use ULA by default
- “IPv6 by Default” is largely understood to be both default and correct behavior for dual-stacked hosts. Most implementations do not perform in this manner.

Functional but unsupported solutions

There are ways to change this default behavior, which is in most cases controlled by `getaddrinfo()`, however.....

These techniques are:

- Problematic to scale across diverse multi functional organizations
- Impose significant additional impediment to operations where implementing IPv6 is already a difficult undertaking for many enterprise organizations
- Functionally impossible for many systems (tablets, embedded systems, operational technology, systems with compliance requirements, guest or partner equipment, legacy equipment) to modify the prefix policy table

In addition...

- We still see remnants of RFC3484 in actively deployed systems.
- RFC6724 was approved in 2012.
- Mean time to implementation is clearly over 10 years, that means even with an update to RFC6724 it would take approximately 10+ years for that change to be widely deployed.
- That timeline doesn't not align with current enterprise deployment needs and schedule.

Simple example of existing behavior

Both systems dual stacked

host gw-test.buragl.io

gw-test.buragl.io has address 10.255.255.3

gw-test.buragl.io has IPv6 address fd58:1e02:dc1a:ffff:3

```
buraglio@netmon:~ $ ping gw-test.buragl.io
```

```
PING gw-test.buragl.io (10.255.255.3) 56(84) bytes of data.
```

```
PING 10.255.255.3 (10.255.255.3) 56(84) bytes of data.
```

```
64 bytes from 10.255.255.3: icmp_seq=1 ttl=63 time=0.569 ms
```

```
64 bytes from 10.255.255.3: icmp_seq=2 ttl=63 time=0.437 ms
```

```
64 bytes from 10.255.255.3: icmp_seq=3 ttl=63 time=0.433 ms
```

Any reason not to adopt this?

Current draft:

[https://www.ietf.org/archive/id/draft-burag
lio-v6ops-ula-03.html](https://www.ietf.org/archive/id/draft-burag
lio-v6ops-ula-03.html)

Questions?