# Using Unique Local Addressing (ULA) in IPv6 Enterprise Networks

Nick Buraglio
October, 2022

## About

Network and security professional since ~1997

IPv6 experience starting ~2002

Past IPv6 workshop instructor

Implementation co-lead for DOE IPv6-only initiative

Active in IETF v6 and SPRING WGs

## What is unique local addressing?

As defined by RFC4193:

*“This IETF standards document defines an IPv6 unicast address format that is globally unique and is intended for local communications.  These addresses are called Unique Local IPv6 Unicast Addresses and are abbreviated in this document as Local IPv6 addresses.  They are not expected to be routable on the global Internet.  They are routable inside of a more limited area such as a site.  They may also be routed between a limited set of sites.”*

## Depreciation of site-local (RFC 1884)

FEC0::/10 was reserved in RFC 1884 for use as site local address

Remnants of it still exist in configuration files and microsoft operating system***

*** This is important as it highlights the time required to fully realize systemic changes - this will become important later in the talk

## Depreciation of site-local (RFC 1884)

RFC 1884 has since been deprecated and replaced with FC00::/7 for used in private networks as defined in RFC 4193.

FC00::/7 is further divided into two /8 subnets:

- **fc00::/8** - the usage of this block has not been clearly defined.
- **fd00::/8** - A unique local prefix is formed by appending 40-bit of randomly-generated bit string (often using a mac address as a “random string”) in the format of FDxx:xxxx:xxxx::/48 leaving the network administrator with 16 bit for subnetting and 64 bit for network identifier.

# What is unique local addressing?

Realistically speaking:

- Unique Local Addressing (ULA) is a unique prefix allocated from the reserved block fc00::/7
- *A* ***depreferenced*** block of IPv6 address space meant for local communications inside of a network boundary, or privately between a subset of private network boundaries (think extranet connectivity).

## What unique local addressing is not

- Not designed as an analog for RFC1918, RFC6598, or RFC5737 IPv4 addressing**
- Not IPv6 mechanism for replicating RFC1918, RFC6598, or RFC5737 IPv4 behavior**
- **Not a good solution for dual stacking a network when IPv6 is expected to be preferred**

** Based on IETF intention, operating system preferences

## Why isn't this an analog for RFC1918?

- Operating systems treat IPv4 space equally
- Operating systems ***do not*** treat all IPv6 equally (by design)
- Care must be taken when ULA is used because:
  - **Operating systems will ignore its existence in the presence of IPv4 without intentional customization, requiring notable operational overhead**
  - While unique based on a 40bit randomization, there is the chance it can overlap.
  - Without quirky hacks, it is limited to a /48 in size

## OK, what does that actually mean?

- In IPv4, all addressing is treated equally
- In IPv6, this is not the case
- Because IPv6 is expected to have multiple addresses on each interface, these addresses must be duly considered by a preferencing structure (see RFC6724).
  - E.g. link local
  - Unique Local Addressing (ULA)
  - Global Unicast Addressing (GUA)
    - Secured / temporary
    - Static
    - Etc.

# Let’s see the difference

```
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.9.9.5  netmask 255.255.255.192  broadcast 10.9.9.63
        inet6 2607:f000:1204:9:221:9bff:fe94:d214  prefixlen 64  scopeid 0x0<global>      EUI-64 GUA
        inet6 fda7:8645:dccd:9:221:9bff:fe94:d214  prefixlen 64  scopeid 0x0<global>      ULA
        inet6 2607:f000:1204:9::5  prefixlen 64  scopeid 0x0<global>                      Static GUA
        inet6 fe80::221:9bff:fe94:d214  prefixlen 64  scopeid 0x20<link>                  link-local
        ether 00:21:9b:94:d2:14  txqueuelen 1000  (Ethernet)
        RX packets 611272622  bytes 790266405906 (790.2 GB)
        RX errors 0  dropped 100  overruns 0  frame 0
        TX packets 613987024  bytes 706072259842 (706.0 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

# Why this matters...

- Preference tables for address source selection is complex and inconsistent
- All comes down to RFC6724 (which has inconsistencies being addressed)

RFC 6724 Default Address Selection for IPv6 September 2012

| Prefix | Precedence | Label | |
|---|---|---|---|
| ::1/128 | 50 | 0 | |
| ::/0 | 40 | 1 | |
| ::ffff:0:0/96 | 35 | 4 | IPv4 |
| 2002::/16 | 30 | 2 | |
| 2001::/32 | 5 | 5 | |
| fc00::/7 | 3 | 13 | ULA |
| ::/96 | 1 | 3 | |
| fec0::/10 | 1 | 11 | |
| 3ffe::/16 | 1 | 12 | |

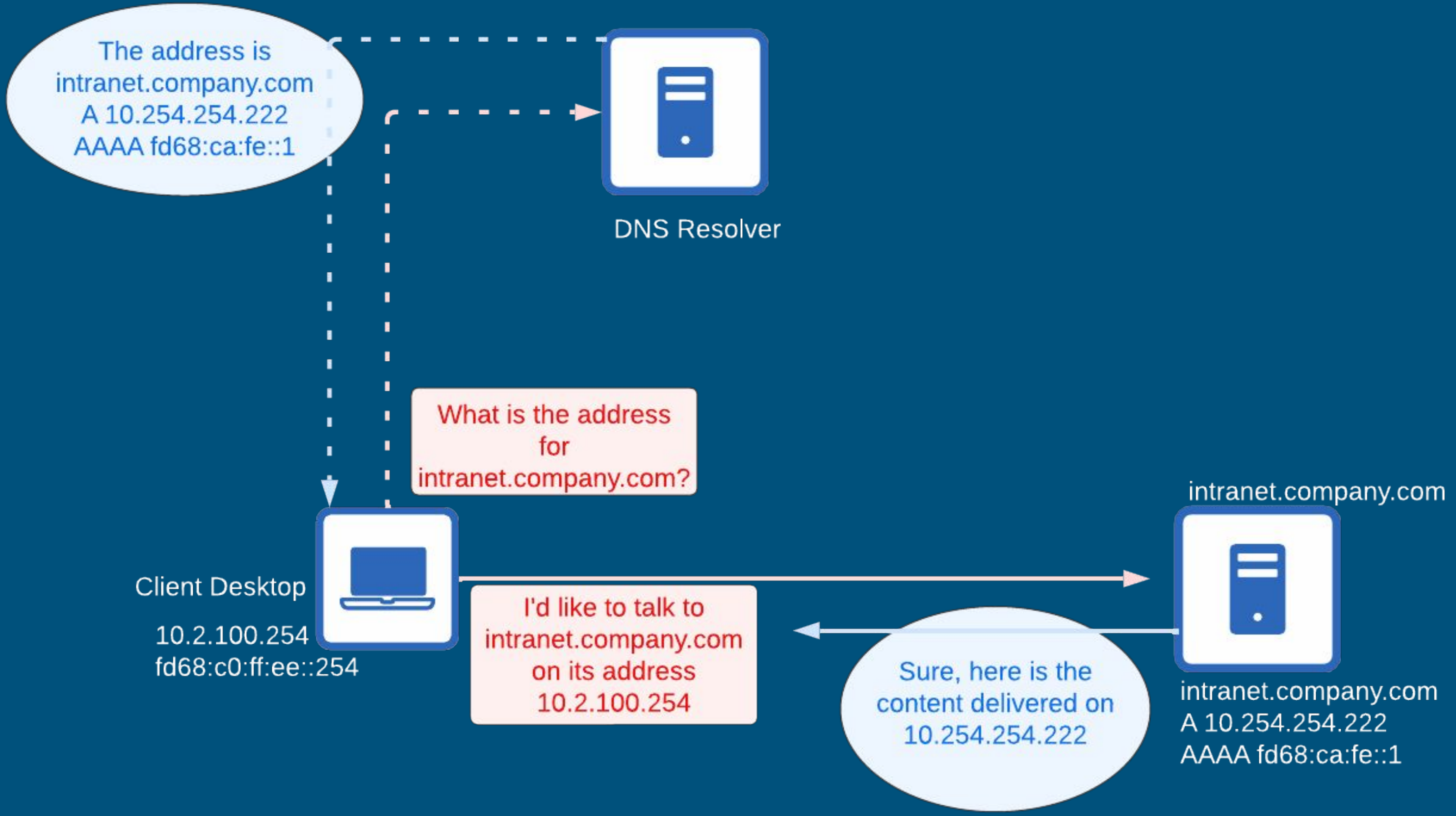
## Address selection is the key

Because of RFC6724, ULA addressing will not be used by default if IPv4 is present.

Further, RFC6724 has inconsistent wording in later sections that allow for inconsistent implementations.

`Since ULAs are defined to have a /48 site prefix, an implementation` ***might*** `choose to add such a row automatically on a machine with a ULA.` (emphasis added by presenter)

## For Example...


The address is intranet.company.com A 10.254.254.222 AAAA fd68:ca:fe::1
DNS Resolver
What is the address for intranet.company.com?
Client Desktop
10.2.100.254
fd68:c0:ff:ee::254
I'd like to talk to intranet.company.com on its address 10.2.100.254
intranet.company.com
Sure, here is the content delivered on 10.254.254.222
intranet.company.com
A 10.254.254.222
AAAA fd68:ca:fe::1

## Functional but unsupportable 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**

## Additionally.....

- 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.

## ULA use cases

- Sensor networks
  - Energy sector, power meters
  - Specialized scientific sensors that are numerous and isolated
- Single stack networks (i.e. IPv6-only)
- Air gapped networks (that are single stacked)
- Networks that
  - Do not have legacy equipment requiring dual-stack or IPv4-only
  - Are able to configure preference for IPv6 across *all* nodes
  - Environments where consistency of configuration is unimportant or understood to not exist
  - Are willing and able to use methods such as AAAA-only records and / or split DNS in order to control resource records

## ULA use cases *in enterprise*

To reiterate - ULA can be successfully used if….

- The network is single stacked (i.e. IPv6-only)
- The network is air gapped, and single stacked IPv6-only
- The environment does not have legacy equipment requiring dual-stack or IPv4-only
- The ability exists to configure preference for IPv6 across *all* nodes
- The organization has expertise and willingness to use DNS to control access (AAAA-only records internally / split DNS views)
- The environment has no requirement for consistency of configuration (i.e. host configuration can vary)

## What problems does this really solve, then?

ULA has specific use cases that work well, but,

- They are *very* specific
- Often require significant control and or resources to enable in a ubiquitous manner
- Care should be taken when considering using it, including significant testing to ensure expected behavior

ULA is [**significantly**] more implementation specific than general use.

## Further reading

Unintended Operational Issues With ULA:

https://datatracker.ietf.org/doc/draft-ietf-v6ops-ula/