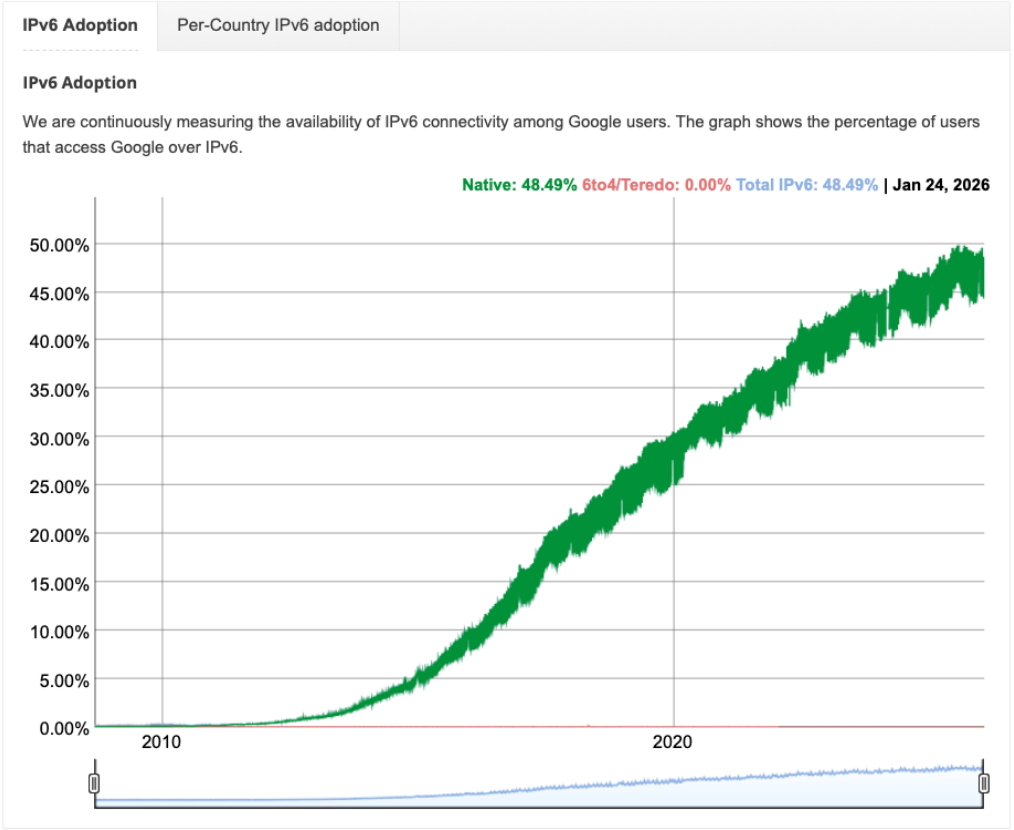
What IPv6 RFCs Don't Say About VPNs

How missing guidance led to IPv6 de-preference

Yejin Cho, John Heidemann
USC/ISI

IPv6 is growing



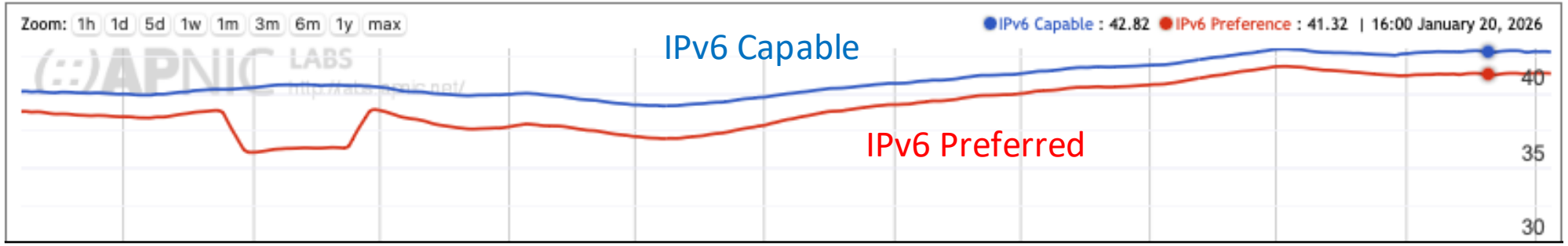
Yay!!

The Problem

But, ‘IPv6 Capable’ is not ‘IPv6 Preferred’.

- **IPv6 Capable:** client has IPv6 connectivity
- **IPv6 Preferred:** client chooses IPv6

Use of IPv6 for World (XA)



This gap!

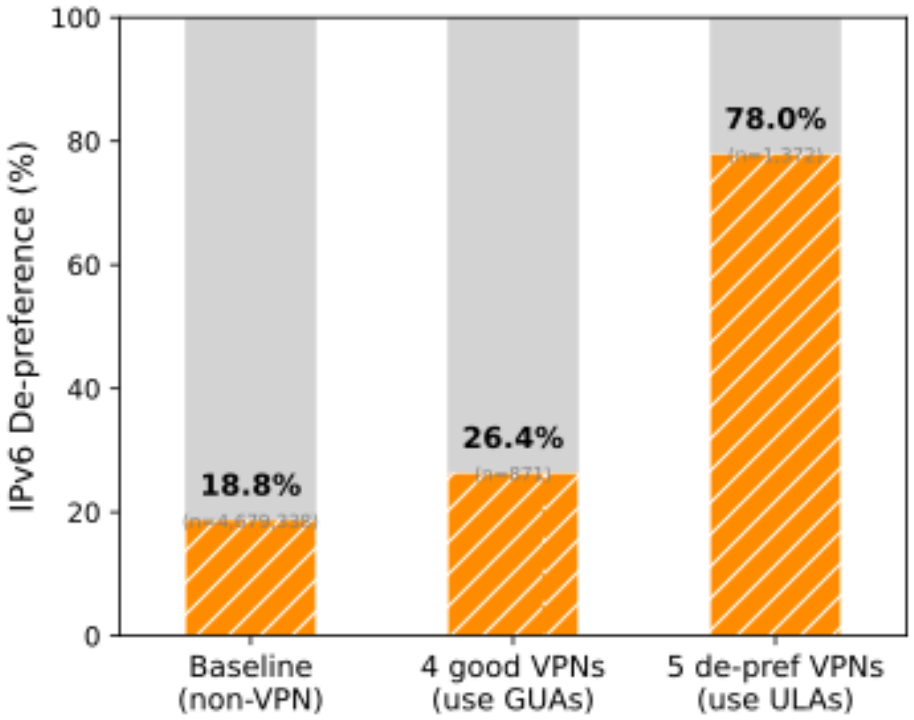
IPv6 De-Preference

= Client has IPv6 support but **IPv6 is de-preferred** to IPv4.

Normally, due to Happy Eyeballs Algorithms, IPv6 should be preferred over IPv4.

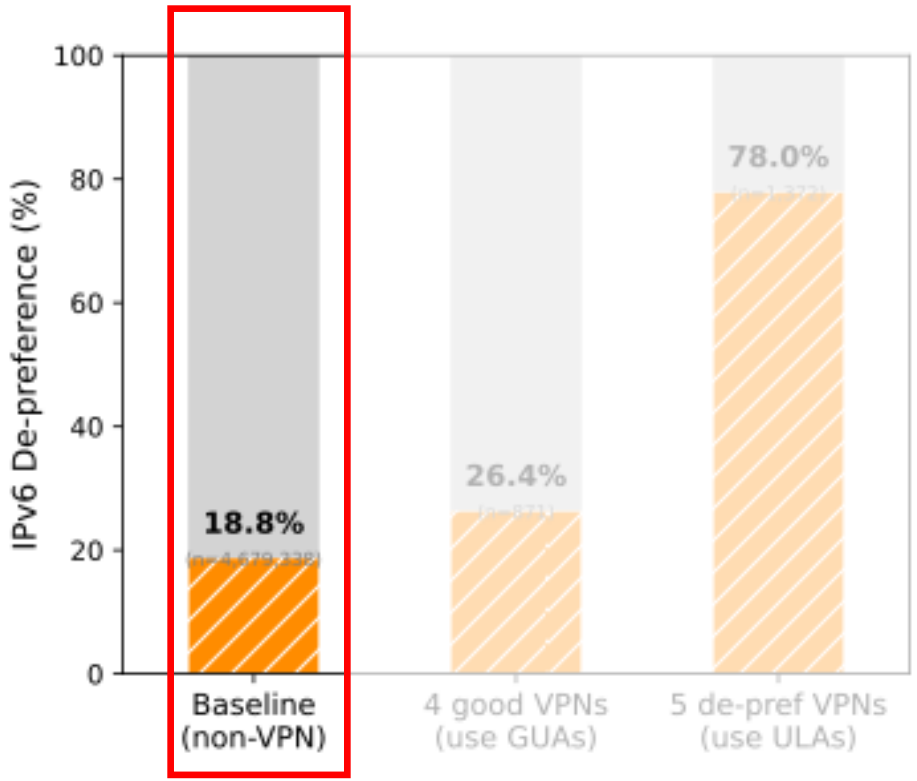
The Problem

We notice that with VPNs, v6 is often **de-preferred** severely.



The Problem

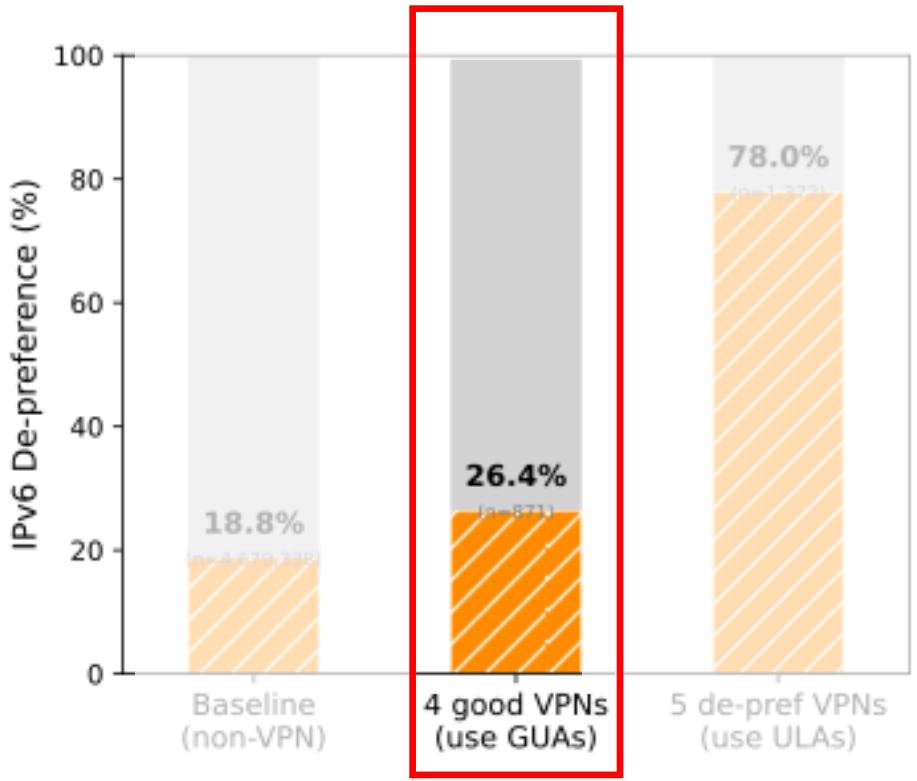
We notice that with VPNs, v6 is often **de-preferred** severely



- non-VPN users use IPv6 most of the times

The Problem

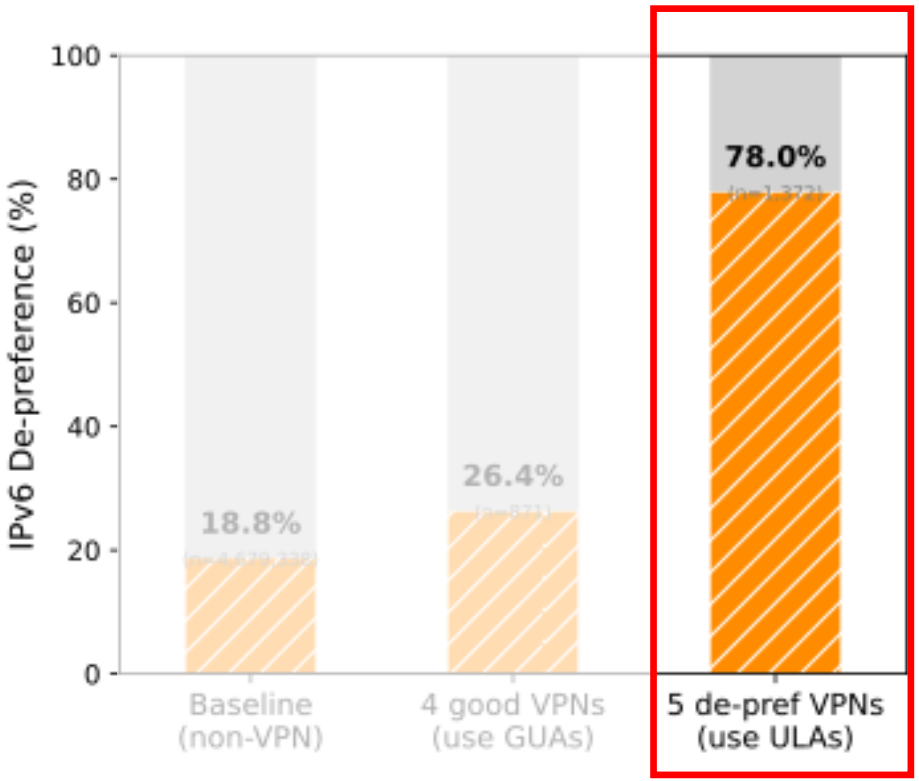
We notice that with VPNs, v6 is often **de-preferred** severely



- Good-VPN users have similar IPv6 usage

The Problem

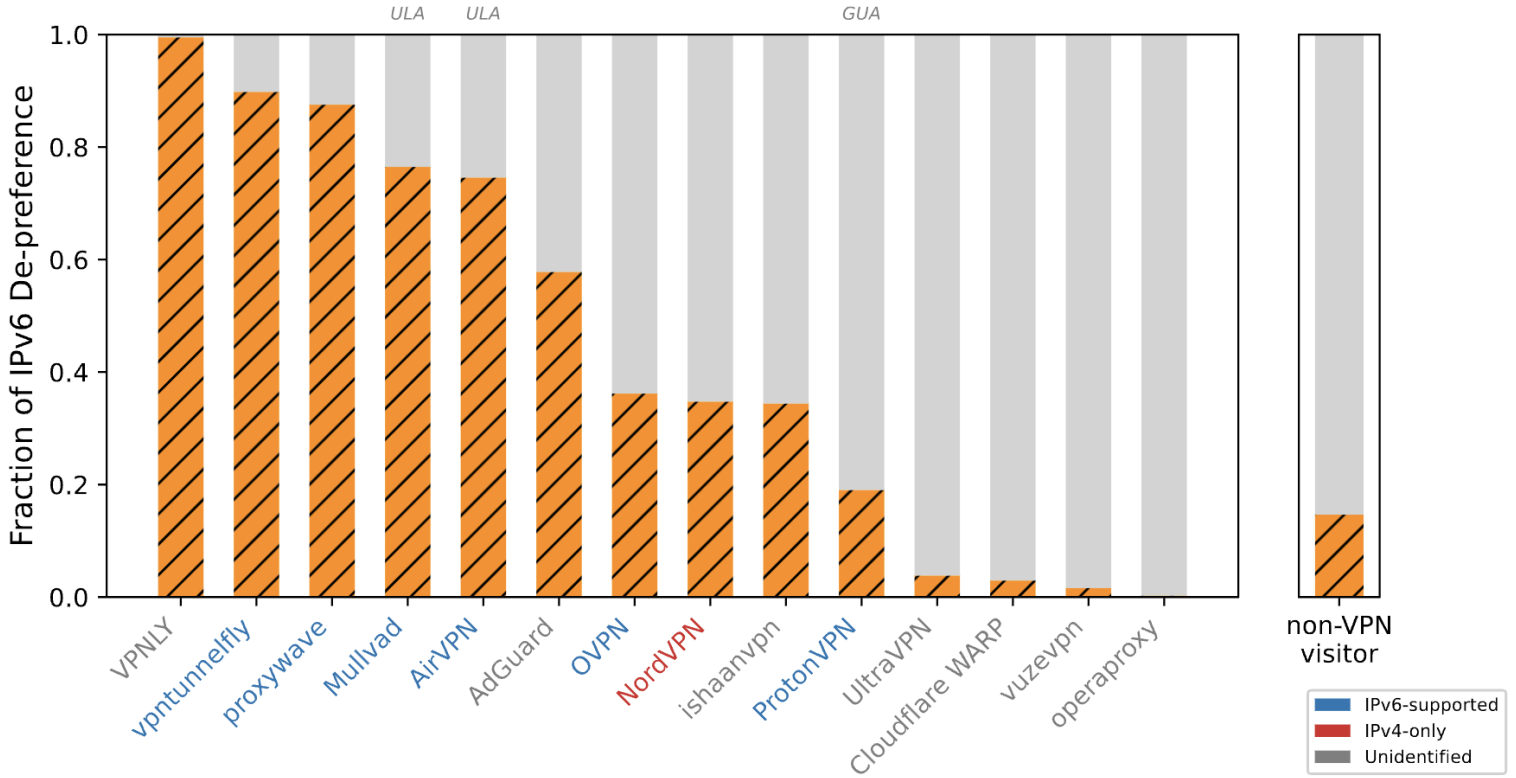
We notice that with VPNs, v6 is often **de-preferred** severely



- De-pref VPN users **often** don't use IPv6

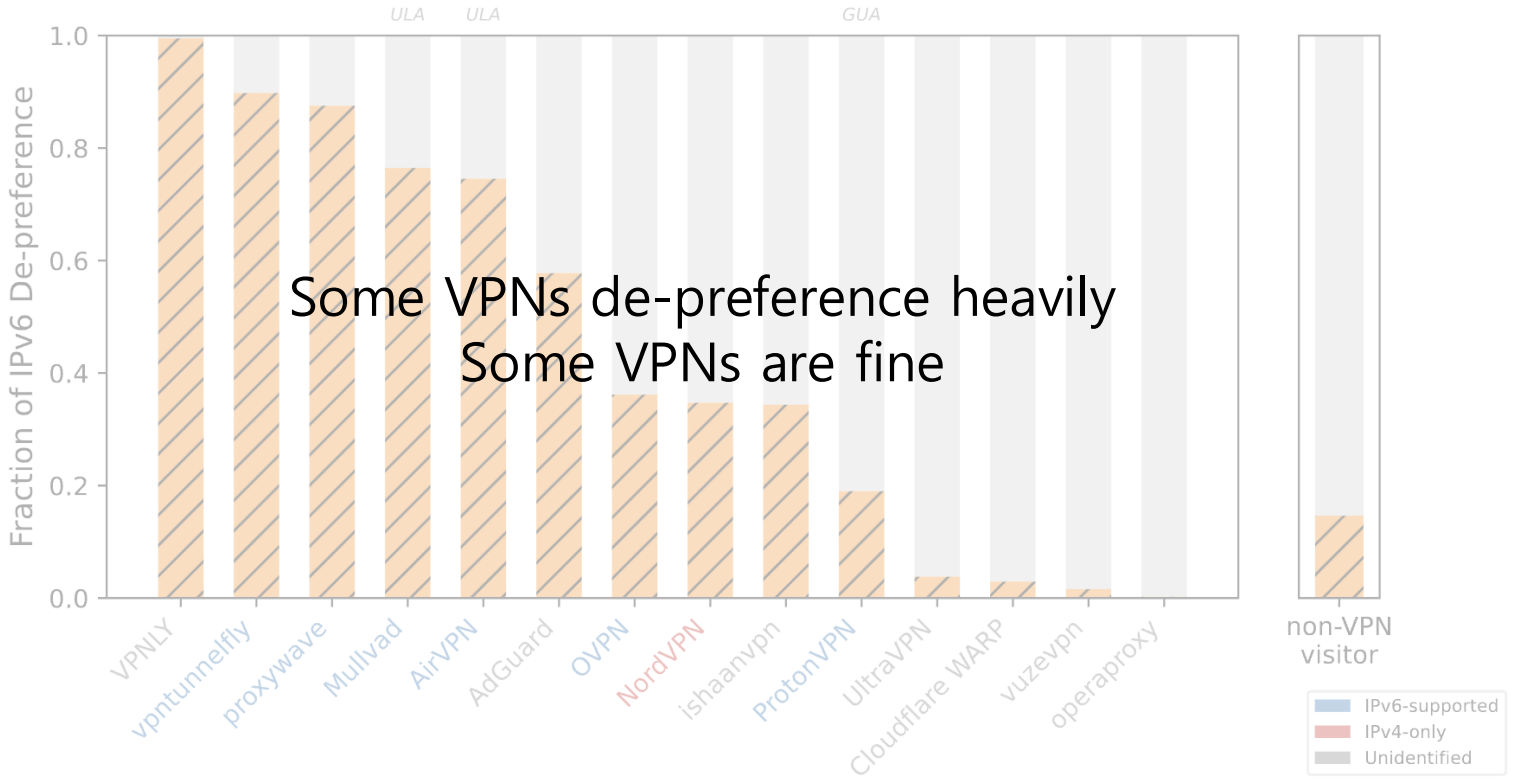
Approach

To access de-preference, we investigated 40 Million visitors logs that includes visitors from 123 VPNs.



Approach

To access de-preference, we investigated 40 Million visitors logs that includes visitors from 123 VPNs.



The Problem: Interactions!

Prioritization rules (of kernel network stack, RFC6724)

Rank	Source	Destination	Reason
1*	IPv6 GUA	IPv6 GUA	Same label (1) and high precedence (40); most preferred pair.
2	IPv6 ULA	IPv6 ULA	Same label (13) and precedence (3); for internal communication.
3*	IPv4 public	IPv4 public	Same label (4) and precedence 35; valid global IPv4 path.
4	IPv4 private	IPv4 private	Same label (4) with reachable private addressing; for LANs.
5*	IPv4 private	IPv4 public	Same label (4) and precedence 35, but requires NAT traversal.
6*	IPv6 ULA	IPv6 GUA	Label mismatch (13 vs. 1) causes strong penalty.

Table 2: Address prioritization rules from RFC-6724 [14]. Label groups addresses into the same class (only equality matters). Precedence is a numeric preference score (higher is preferred).

The Problem: Interactions!

Reason:

Interaction between:

- Prioritization rules (of kernel network stack, RFC6724)
- VPN interface address assignment

Device has multiple interfaces & VPN users always select v4 address as source address

The Problem: ULA(Unique Local Address)-based VPNs

Example: Mullvad VPN's interface

```
utun7: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1320
        inet 10.131.148.184 --> 10.0.0.255 netmask 0xff000000
        inet6 fe80::9afc:84ff:fee7:1c37%utun7 prefixlen 64 scopeid 0x18
        inet6 fc00:bbbb:bbbb:bb01:d:0:3:94b8 prefixlen 64
        nd6 options=201<PERFORMNUD,DAD>
```

The Problem: ULA-based VPNs

Example: Mullvad VPN

IPv4 Private

```
utun7: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1320
        inet 10.131.148.184 --> 10.0.0.255 netmask 0xff000000
        inet6 fe80::9afc:84ff:fee7:1c37%utun7 prefixlen 64 scopeid 0x18
        inet6 fc00:bbbb:bbbb:bb01:d:0:3:94b8 prefixlen 64
        nd6 options=201<PERFORMNUD,DAD>
```

IPv6 ULA (Unique Local Address)

And the prioritization rule prefers V4 private >> V6 ULA!

Why does de-preference happen?

- By design, IPv6 is pure -- ULA source means this traffic never reaches outside (NAT will NOT happen!!)
- VPN implementors often carry over IPv4 design assumptions
- Equivalent of v4 private is v6 ULA

Why De-preference Was Surprising

- **No explicit guidance exists**
- No IPv6 RFC specifies

  Which address type should be used as a VPN inner-tunnel source

Result

: Implementers must infer behavior

Which is not easy!

Possible Solution

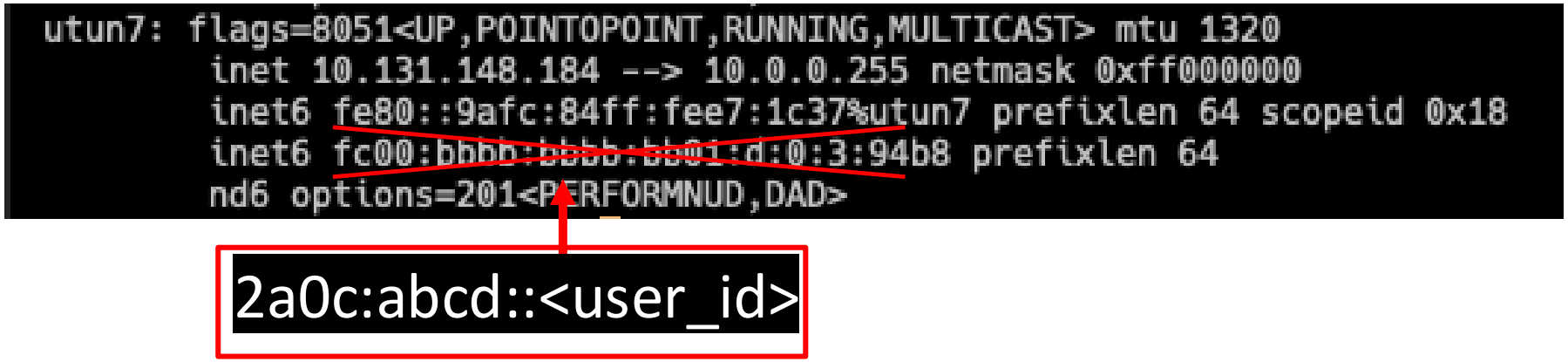
(1) VPNs assigning each user a unique GUA address
(2) VPN assigns a shared static GUA ★
(3) Create a new Address Class (TLA) ★
(4) Status Quo -- stick with ULA

★: our recommendation

Solution #1 **VPNs assigning each user a unique GUA address**

Pros: address will have actual meaning, can be used to specify users

Cons: complexity of managing address allocations without much purpose

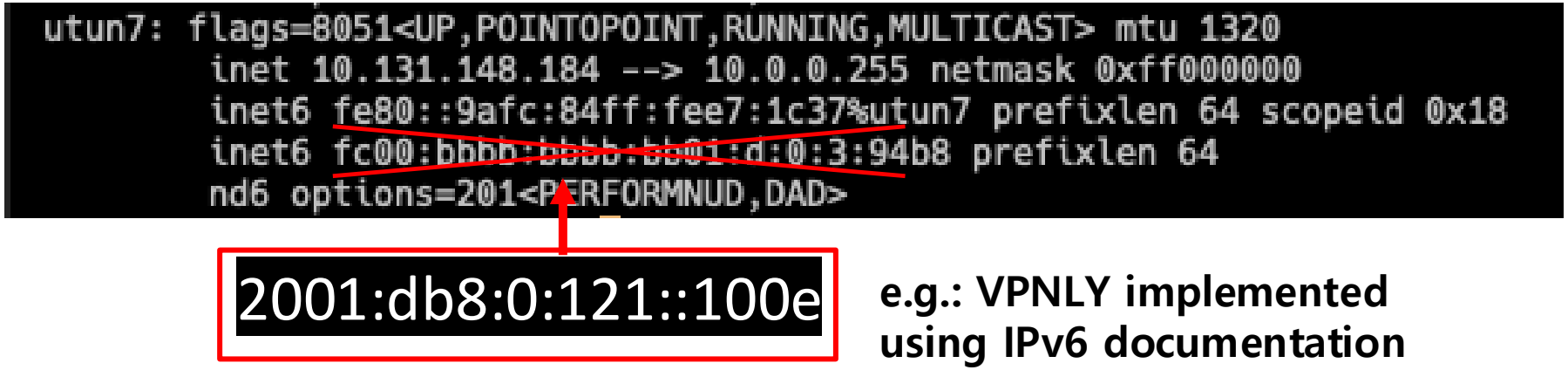


Solution #2 VPN assigns a shared static GUA

Pros: ease of implementation

Cons: unclear which one to use!

implementors often use others' addresses



e.g.: VPNLY implemented using IPv6 documentation prefix

Solution #3 Create a new Address Class (TLA: Tunnel Local Address)

- to specify that this traffic is tunneling through, will be encapsulated later and thus should not be de-preferred.
- Use the remaining half ULA space

Pros: Appropriate address

Cons: Difficulty of adding a new class

Solution #3 Create a new Address Class (TLA: Tunnel Local Address)

TLA would be similar to Teredo (2001:0::/32)

- Teredo: IPv6 over UDP over IPv4
  - Inner IPv6 address indicates that this packet is Teredo'ed!

Solution #4 **change prioritization rules**

Pros: no need to change VPN addressing

Cons: might break things

In fact, a lot of people are changing their /etc/gai.conf files.

Conclusion

- IPv6 is de-preferred in VPNs
- We suggested several solutions,
  - (1) VPNs assigning each user a unique GUA address
  - (2) VPN assigns a shared static GUA ★
  - (3) Create a new Address Class (TLA) ★
  - (4) Status Quo -- change prioritization rules
- which solution do you prefer?