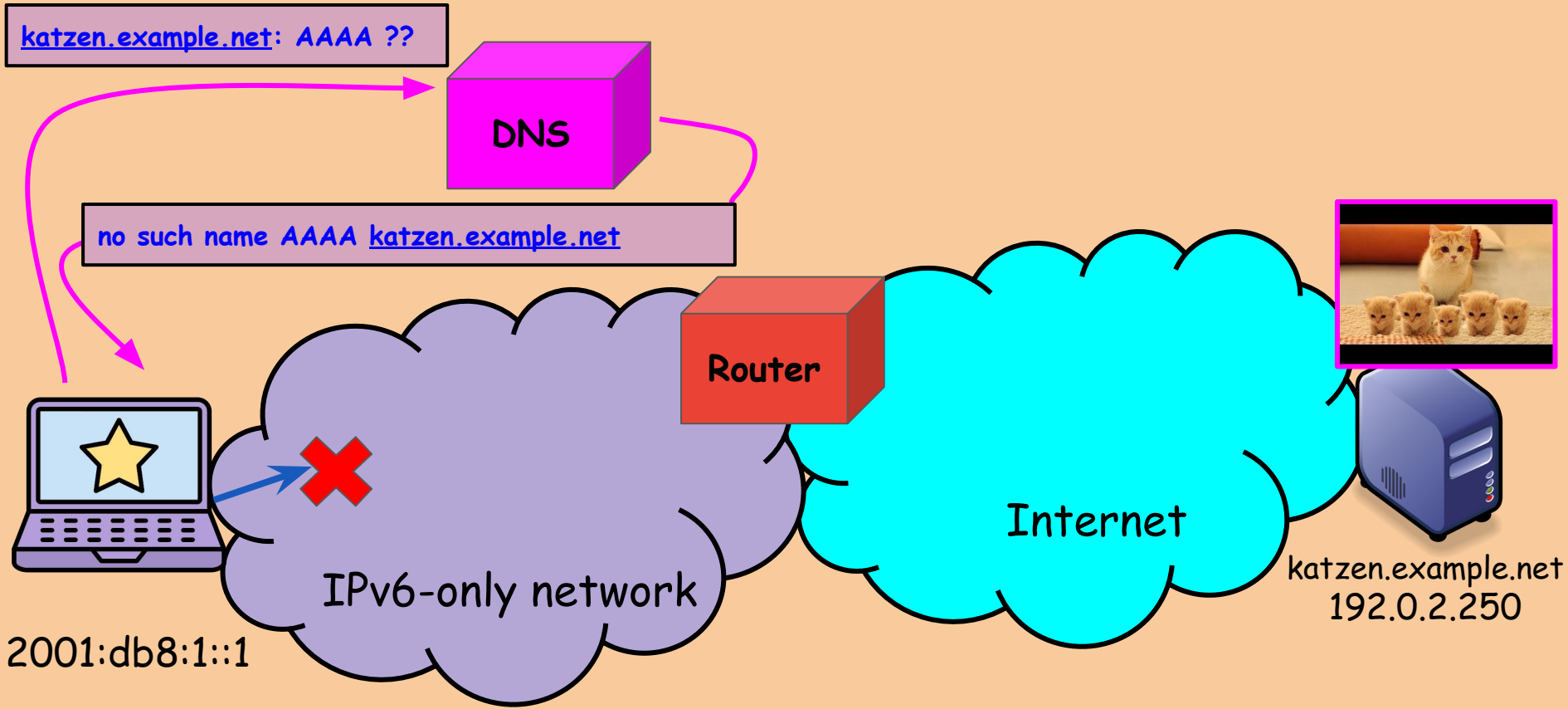
Let's stop using DNS

Let's stop using DNS64

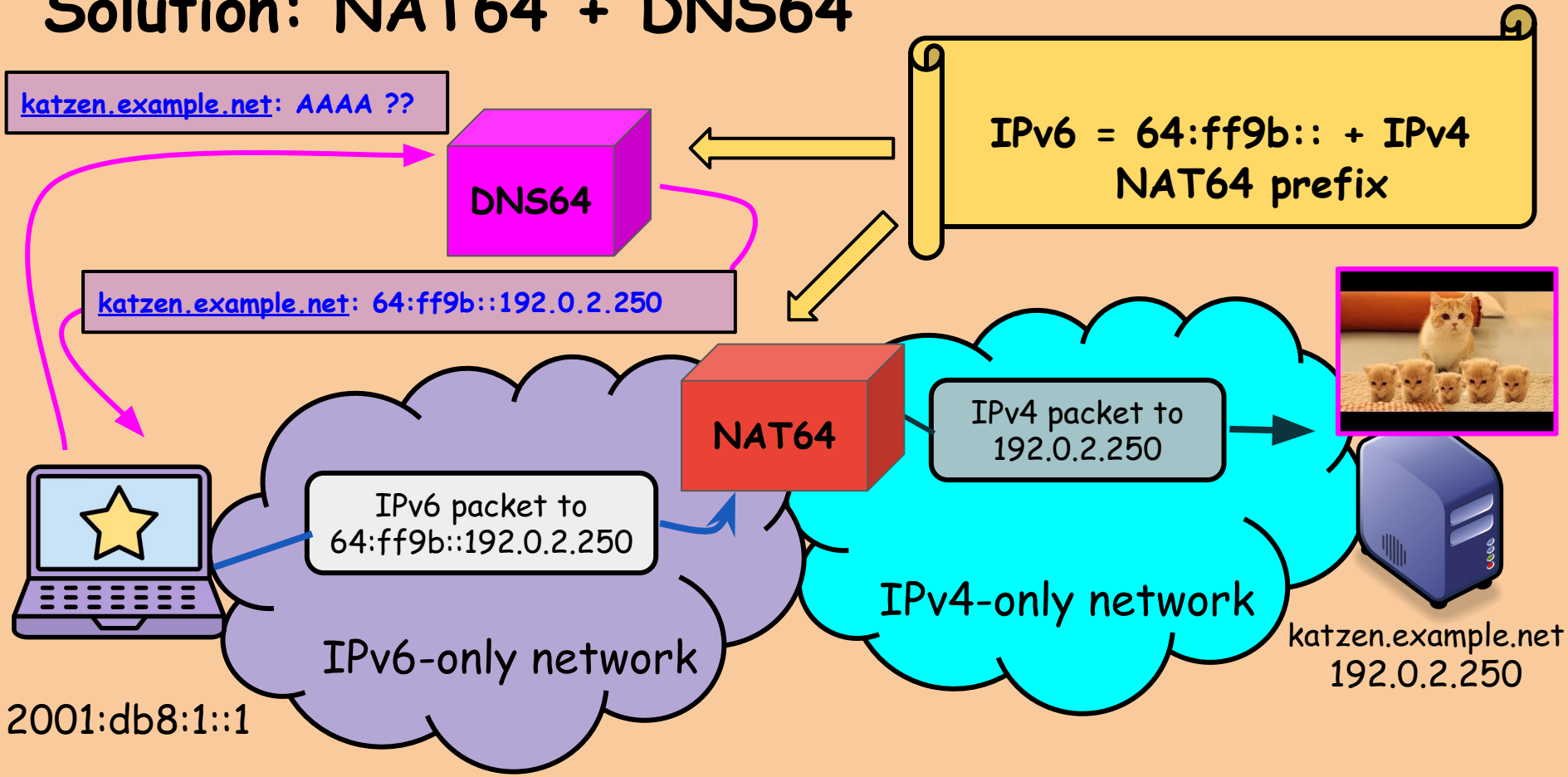
Especially for NAT64 prefix discovery

Jen Linkova, furry13@gmail.com, NANOG93

Problem: IPv6-Only Client, IPv4-Only Server



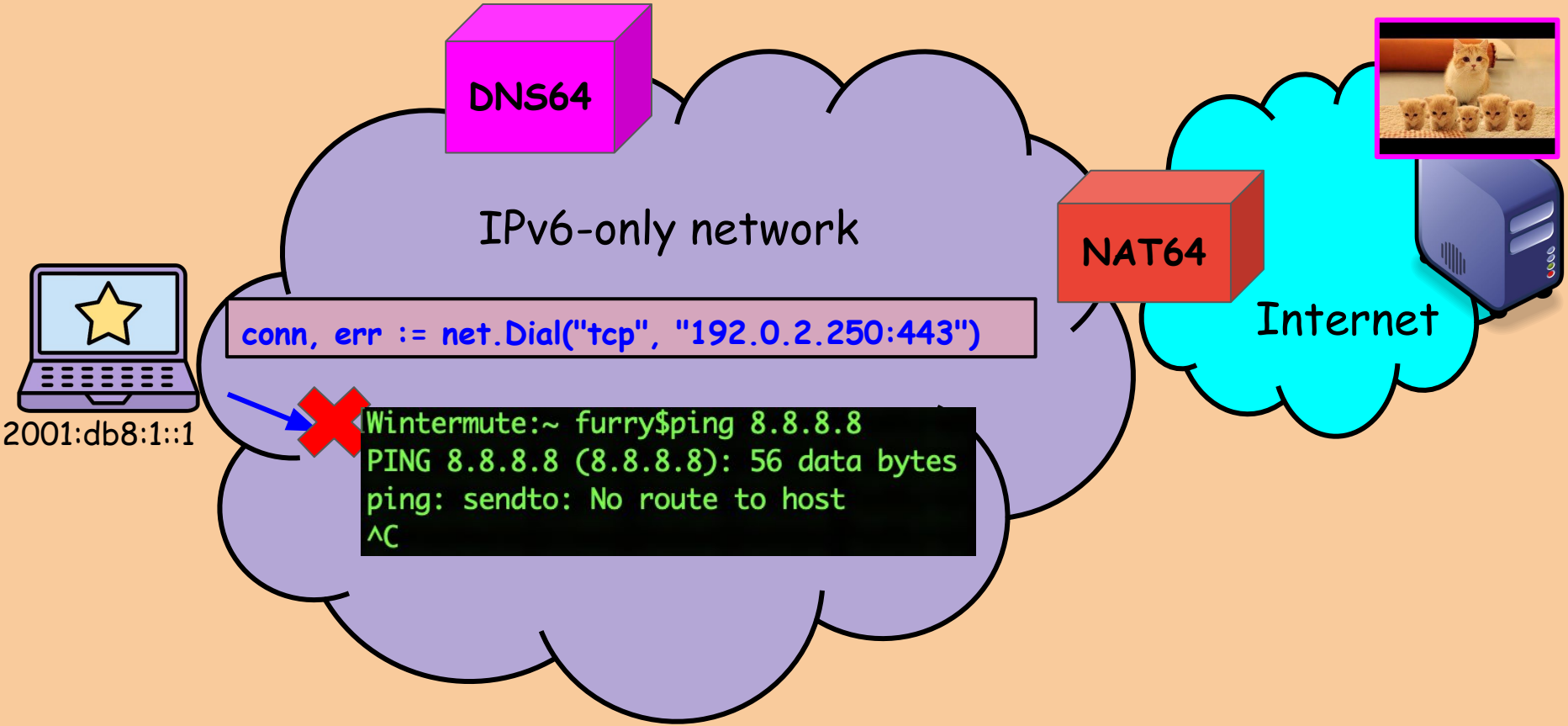
Solution: NAT64 + DNS64



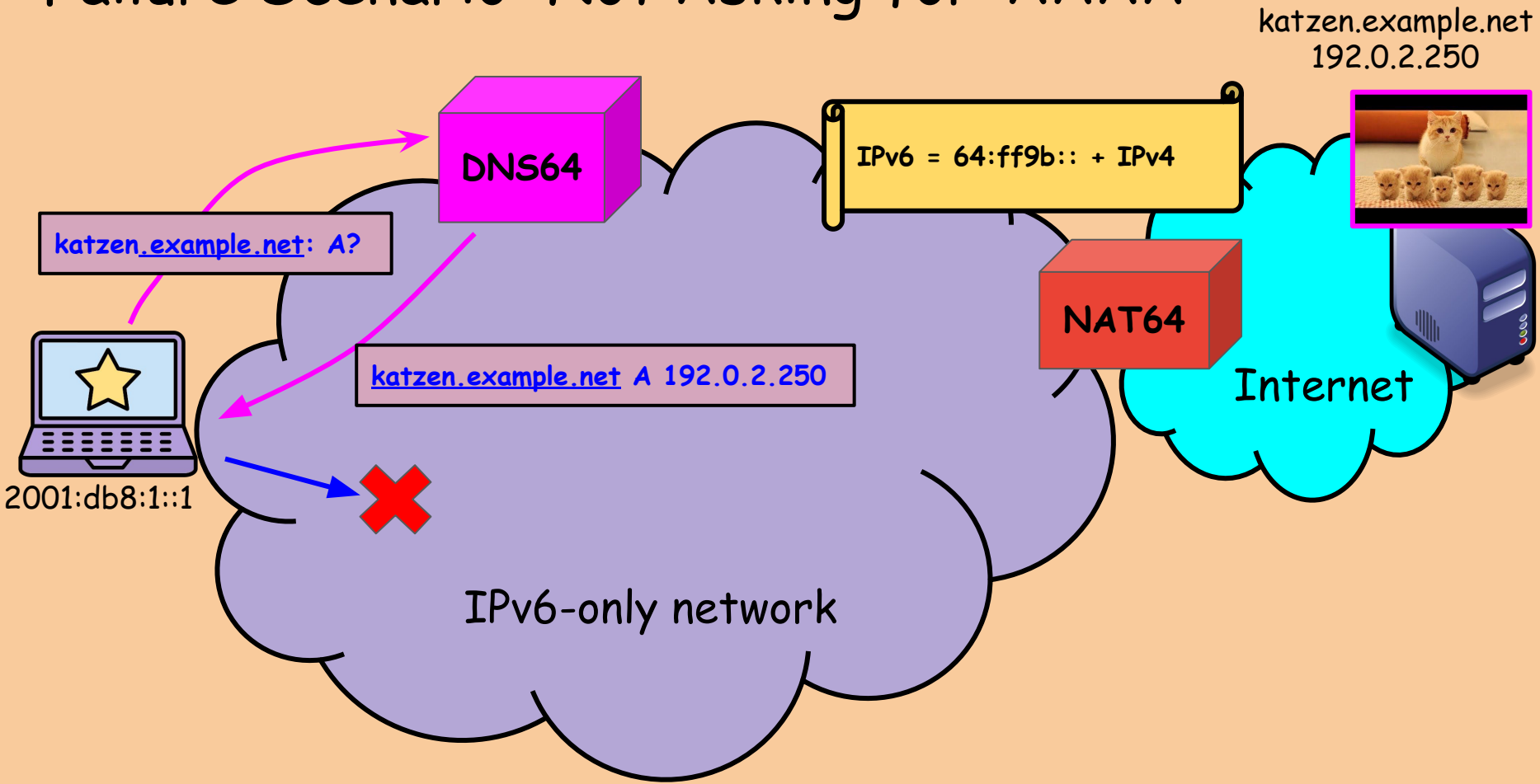
"Do not let us mistake necessary evils for good."

C. S. Lewis

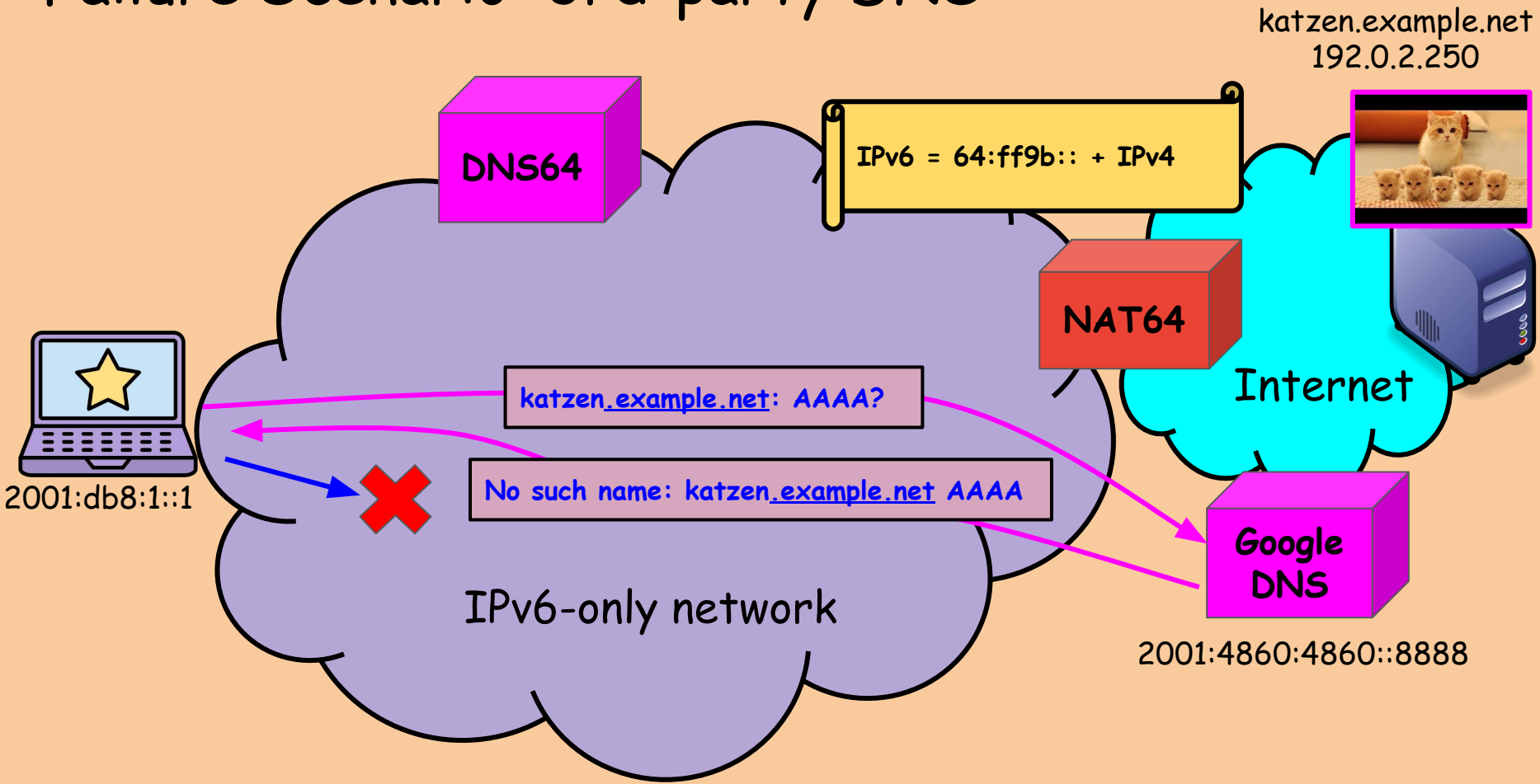
Failure Scenario: IPv4 Literals



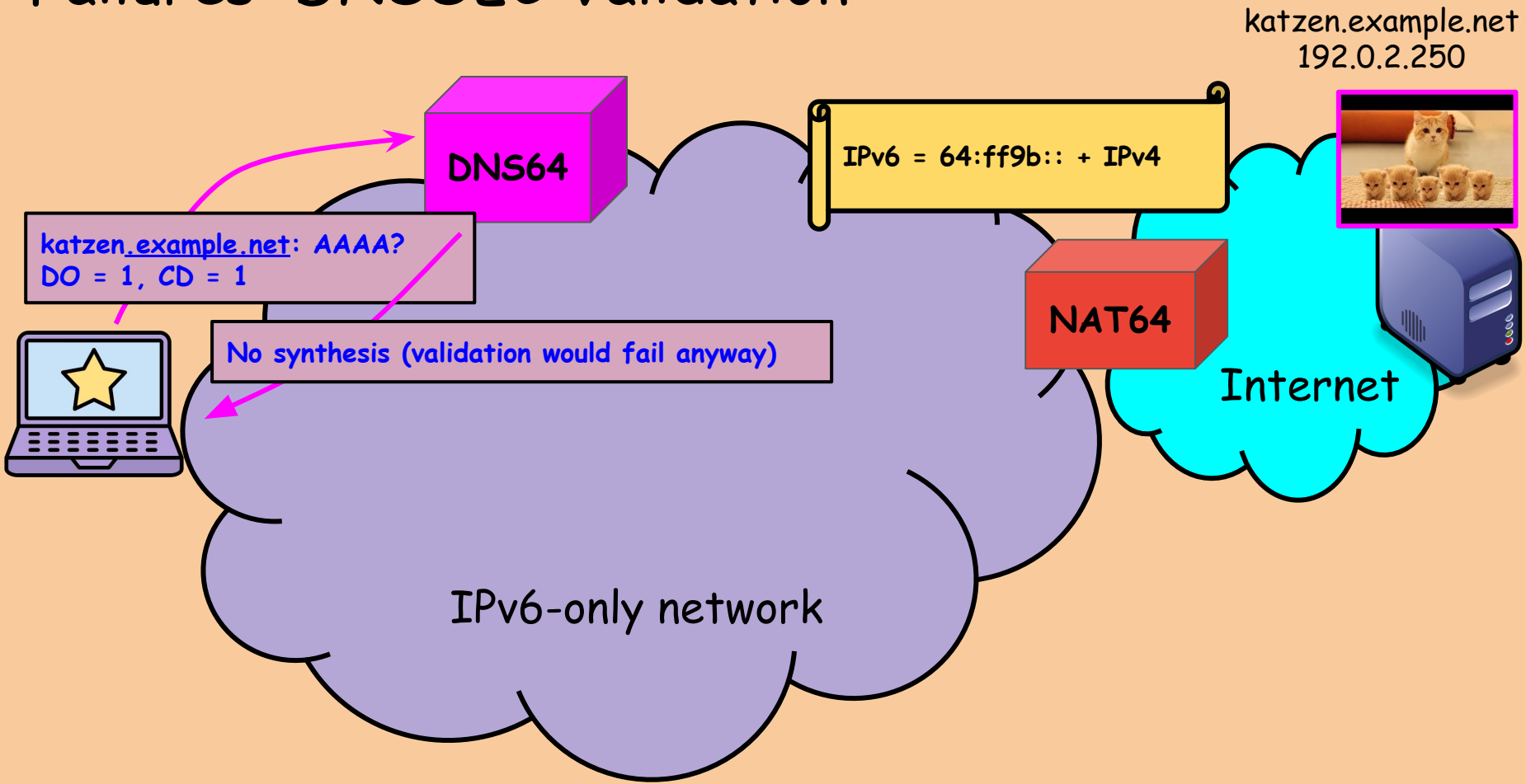
Failure Scenario: Not Asking for 'AAAA'



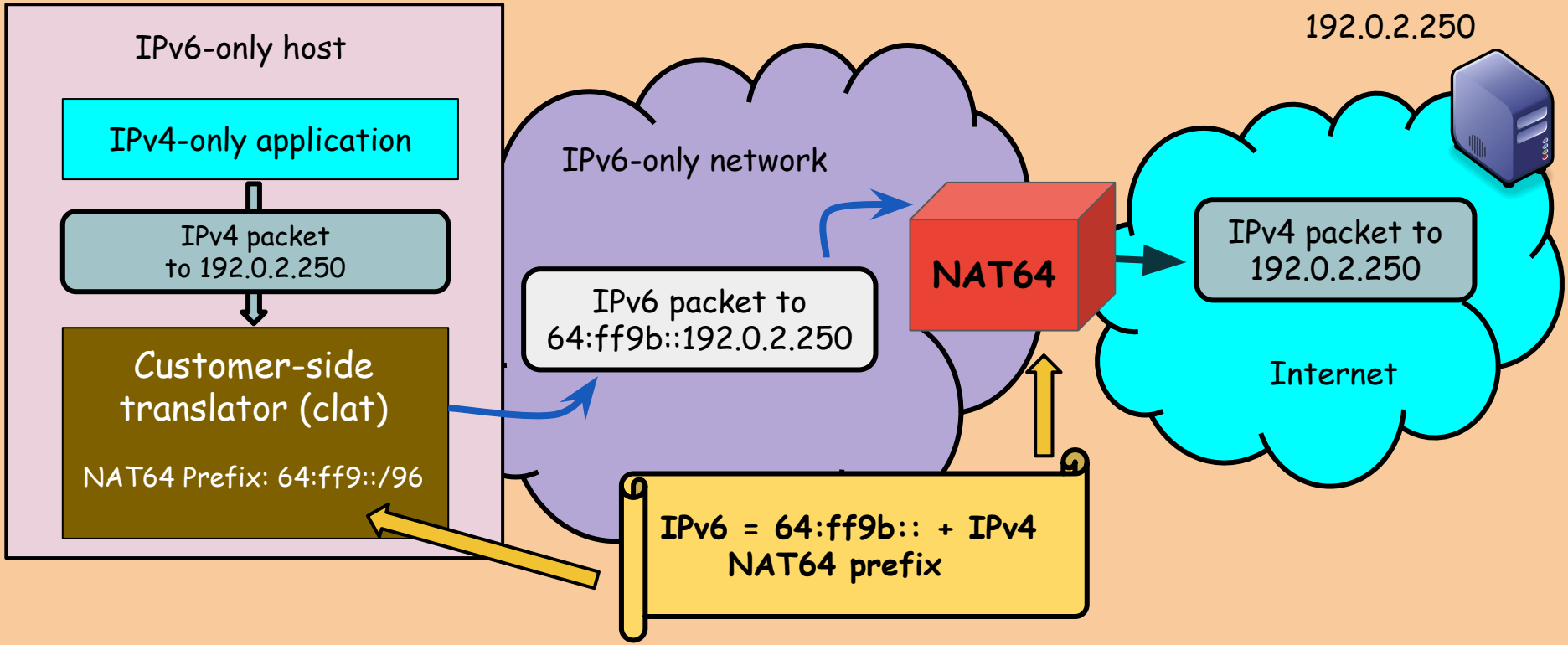
Failure Scenario: 3rd-party DNS



Failures: DNSSEC Validation



Solution: 464XLAT (RFC6877)

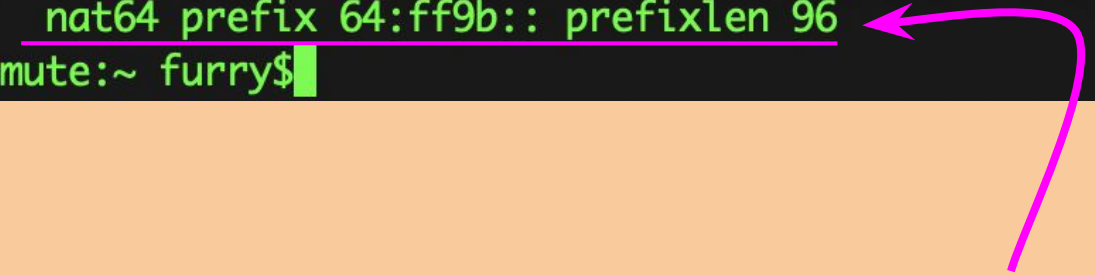


The Importance of Knowing NAT64 Prefix

- Key elements of 464XLAT:
  - Provider-side translator (PLAT, NAT64): network edge
  - Customer-side translator (CLAT): IPv6-only clients
- PLAT and CLAT must use the same NAT64 prefix
  - Well-known prefix: 64:ff9b::/64
  - Can be network-specific

Example: MacOS 13+ in IPv6-only Mode

```
Wintermute:~ furry$ifconfig en0 | grep -E "inet6|nat64"
        inet6 fe80::1433:248b:a7c8:3fe0%en0 prefixlen 64 secured scopeid 0xf
        inet6 2001:67c:64:42:1829:fab0:5dbd:b4d1 prefixlen 64 autoconf secured
        inet6 2001:67c:64:42:e980:98f6:17f0:3cd9 prefixlen 64 autoconf temporary
        inet6 2001:67c:64:42:cd6:7c7b:cb6:9adb prefixlen 64 clat46
        nat64 prefix 64:ff9b:: prefixlen 96
Wintermute:~ furry$
```



How does the host obtain the prefix?

Internet Engineering Task Force (IETF)
Request for Comments: 7050
Category: Standards Track
ISSN: 2070-1721

T. Savolainen
Nokia
J. Korhonen
Broadcom
D. Wing
Cisco Systems

November 2013

Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis

Abstract

This document describes a method for detecting the presence of DNS64 and for learning the IPv6 prefix used for protocol translation on an access network. The method depends on the existence of a well-known IPv4-only fully qualified domain name "ipv4only.arpa.". The information learned enables nodes to perform local IPv6 address synthesis and to potentially avoid NAT64 on dual-stack and multi-interface deployments.

```
Wintermute:~ furry$ ipconfig getra en0
RA Received 11/01/2024 03:59:33 from fe80::42:1, length 96, hop limit 0, lifetime 3600s, reachable 0ms,
 retransmit 0ms, flags 0x40=[ other ], pref=medium
       prefix info option (3), length 32 (4):  2001:67c:64:42::/64, Flags [ auto ], valid time 86400s,
 pref. time 3600s
       pref64 option (38), length 16 (2): 64:ff9b::/96 lifetime 3600s
       rdnss option (25), length 24 (3):  lifetime 86400s, addr: 2001:67c:64:53::53:1
       source link-address option (1), length 8 (1): 00:50:56:bb:b5:0b
```

```
Wintermute:~ furry$dig +short @2001:4860:4860::8888 ipv4only.arpa a
192.0.0.171
192.0.0.170
Wintermute:~ furry$dig +short @2001:4860:4860::8888 ipv4only.arpa aaaa
Wintermute:~ furry$dig +short @2001:67c:64:53::53:1 ipv4only.arpa aaaa
64:ff9b::c000:ab
64:ff9b::c000:aa
Wintermute:~ furry$
```

Google public DNS

ripemtg DNS64

RFC7050: What Could Possibly Go Wrong?

3.1. Validation of Discovered Pref64::/n

If a node is using an insecure channel between itself and a DNS64 server or the DNS64 server is untrusted, it is possible for an attacker to influence the node's Pref64::/n discovery procedures. This may result in denial-of-service, redirection, man-in-the-middle, or other attacks.

To mitigate against attacks, the node SHOULD communicate with a trusted DNS64 server over a secure channel or use DNSSEC. NAT64 operators SHOULD provide facilities for validating discovery of Pref64::/n via a secure channel and/or DNSSEC protection.

Secure Channel: a communication channel a node has between itself and a DNS64 server protecting DNS protocol-related messages from interception and tampering. The channel can be, for example, an IPsec-based virtual private network (VPN) tunnel or a link layer utilizing data encryption technologies.

What Else Could ~~Possibly~~ Go Wrong?

```
Wintermute:~ furry$dig +short ipv4only.arpa aaaa
Wintermute:~ furry$
```

```
Wintermute:~ furry$dig  ipv4only.arpa aaaa

; <<>> DiG 9.10.6 <<>> ipv4only.arpa aaaa
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40570
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;ipv4only.arpa.                 IN      AAAA

;; AUTHORITY SECTION:
ipv4only.arpa.          728     IN      SOA     sns.dns.icann.org. noc.dns.icann.org. 2022072100 7200 3600 604800 3600

;; Query time: 54 msec
;; SERVER: ::1#53(::1)
;; WHEN: Tue Oct 29 19:25:51 AEDT 2024
;; MSG SIZE  rcvd: 125

Wintermute:~ furry$
```

Local resolver

RFC7050: Other Issues

- Stack initialization delay:

  At least 1 RTT is required to discover the prefix

- No way to signal the prefix change:

  The prefix is used until the AAAA TTL expired

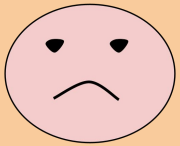
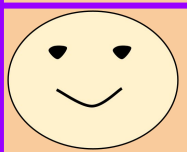
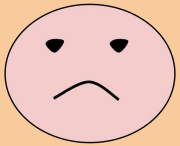
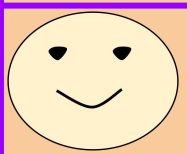
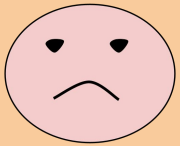
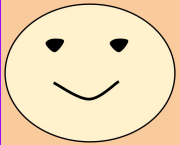
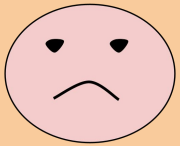
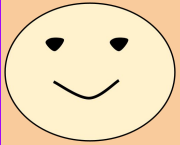
Another Way to Discover the Prefix

Router Advertisements contain all info to configure the network stack:

- Default gateway address
- /64 prefix(es) to configure addresses
- DNS servers

```
Wintermute:~ furry$ ipconfig getra en0
RA Received 11/01/2024 03:59:33 from fe80::42:1, length 96, hop limit 0, lifetime 3600s, reachable 0ms,
 retransmit 0ms, flags 0x40=[ other ], pref=medium
       prefix info option (3), length 32 (4):  2001:67c:64:42::/64, Flags [ auto ], valid time 86400s,
 pref. time 3600s
       pref64 option (38), length 16 (2): 64:ff9b::/96 lifetime 3600s
       rdnss option (25), length 24 (3):  lifetime 86400s, addr: 2001:67c:64:53::53:1
       source link-address option (1), length 8 (1): 00:50:56:bb:b5:0b
```

Include PREF64 into Router Advertisements, RFC8781 (April 2020)

	RFC7050 (ipv4only.arpa.)	RFC8781 (PREF64 in RA)
Security	 DNS response can be spoofed	 Fate sharing with IPv6 config
Compatibility with 3-rd party resolvers	 Clients MUST use network-provided resolvers	 Any resolvers can be used
Initialization Speed	 1 RTT to the DNS	 No delay
Prefix Change Flexibility	 Controlled by the AAAA TTL	 Instant prefix change

Deployment Recommendations

Operators of IPv6-only networks:

Provide PREF64 in RA (and DNS64 for the time being…)

Operators of IPv6-Mostly networks (DHCPv4 Option 108, RFC8925):

Provide PREF64 in RA (DNS64 is optional)

Developers:

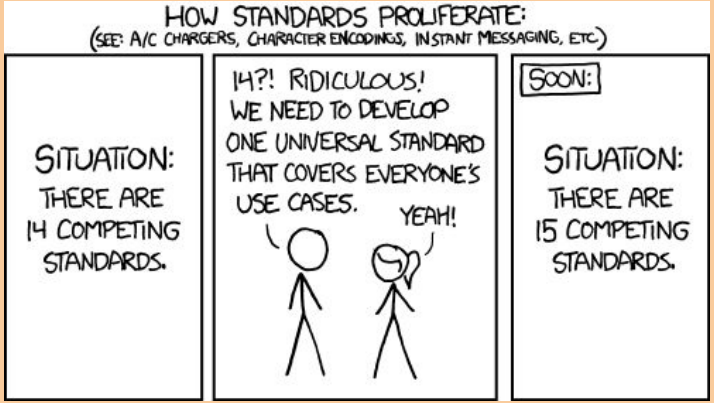
Implement PREF64 support instead of resolving ipv4only.arpa

ietf.org/archive/id/draft-ietf-v6ops-prefer8781-00.html

Prefer use of RFC8781 for Discovery of IPv6 Prefix Used for IPv6 Address Synthesis

Abstract

RFC7050 describes a method for detecting the presence of DNS64 and for learning the IPv6 prefix used for protocol translation (RFC7915). This methodology depends on the existence of a well-known IPv4-only fully qualified domain name "ipv4only.arpa.". Because newer methods exist that lack the requirement of a higher level protocol, instead using existing operations in the form of native router advertisements, discovery of the IPv6 prefix used for protocol translation using RFC7050 should be discouraged. RFC7050 MAY only be used if other methods (such as RFC8781]) can not be used.



HTTPS://XKCD.COM/927/

4. Recommendations for PREF64 Discovery

4.1. Deployment Recommendations

Operators deploying NAT64 networks SHOULD provide PREF64 information in Router Advertisements as per [RFC8781].

4.2. Mobile network considerations

Use of [RFC8781] may not be currently practical for networks that have more complex network control signaling or rely on slower network component upgrade cycles, such as mobile networks. These environments are encouraged to incorporate [RFC8781] when made practical by infrastructure upgrades or software stack feature additions.

4.3. Clients Implementation Recommendations

Clients SHOULD obtain PREF64 information from Router Advertisements as per [RFC8781] instead of using [RFC7050] method. In the absence of the PREF64 information in RAs, a client MAY choose to fall back to RFC7050.

https://datatracker.ietf.org/doc/draft-ietf-v6ops-prefer8781/

QUESTIONS?
COMMENTS?