ICMP Extensions for IP/ICMP translators (XLATs)

draft-equinox-intarea-icmpext-xlat-source

David 'equinox' Lamparter, Jen Linkova
IETF120, July 2024
WHY?
464XLAT

IPv6 packet traverses the network And reaches NAT64 device

clat translates v4 to v6:
Src: dedicated v6 address
Dst: 64:ff9b:: + ipv4

Application sends an ipv4 packet

NA64 translates v6 to v4
Src: NAT pool IP
Dst: 8.8.8.8

Src: clat46 ipv6
Dst: 64:ff9b::8.8.8.8

Src: 192.0.0.2
Dst: 8.8.8.8
ICMPv4, src: 198.51.100.1

RouterB
198.51.100.1

NAT64 (PLAT), prefix 64:ff9b::/96

RouterA
2001:db8:a::a

IPv6-only host

ICMPv4, src: 198.51.100.1

ICMPv6
src: 64:ff9b::198.51.100.1

ICMPv6
src: 2001:db8:a::a

ICMPv6
src: 64:ff9b::

RouterA
2001:db8:a::a

ICMPv6, src: ???????

problem!
Observed Operational Issues

● Traceroute can not represent IPv6-only hops
  ○ Confusing for users
  ○ Complicates troubleshooting

● ICMPv6 Packet Too Big are not translated
  ○ Broken PMTUD
WHAT?
ICMPv6 -> ICMPv4 Translation

- Define how to translate IPv6 source addresses which do not belong to the NAT64 prefix
- Preserve information about the original IPv6 source address.
HOW?
Terminology: Untranslatable

- “Untranslatable” IPv6 address:
  - doesn’t belong to NAT64 prefix
  - there are no explicit address mapping
Translating the Untranslatable

- Request IANA to allocate 192.0.0.11/32
- Use 192.0.0.11 to translate “untranslatable” IPv6 sources
  - ICMPv6 Destination Unreachable, Time Exceeded, PTB
- Some implementations already use 225.0.0.0
  - traceroute represents hop#X as ‘225.0.0.X’
Preserving the Original IPv6 Source Address

Introducing “IPv6 Original Source Extension” Object for **ICMPv4**
Using IPv6 Original Source Extension Object

- SHOULD be added to ICMPv4 when translating ICMPv6
  - IPv6 source is ‘untranslatable’
  - ICMPv6 Types are:
    - Destination Unreachable
    - Time Exceeded
    - Packet Too Big (*)
- MUST not be used in any other cases
Please Read the Draft

- Details on the translator behaviour:
  - adding new Extension Structure
  - appending extension object to existing structure

- Updates to RFC7915 (translation algorithm)
Similar Work: ICMP Extensions

- RFC5837, Interface IP Address Sub-Object
  
  If the eliciting datagram was IPv4, the IP Interface Sub-Object MUST represent an IPv4 address. Likewise, if the eliciting datagram was IPv6, the IP Interface Sub-Object MUST represent an IPv6 address.

- Draft-fenner-intarea-extended-icmp-hostid
  
  Possibly. Need to clarify NAT64 logic
Open Issues

● When to add the extension object?
  ○ Before or after the translation?
    ■ Before (current text): easier to translate
    ■ After: can be added when ICMPv6 PTB is translated to ICMPv4 "Fragmentation Needed"

● Are we over specifying?
  ○ Maybe just define how the packet must look like?
Next Steps

● Is problem worth solving?
● Does the proposed solution make sense?
● Comments? Suggestions?
● Adoption?