IP/ICMP Translation Algorithm

X. Li, C. Bao, F. Baker

2009-07-27

Outline

- Introduction and Motivation
  - Translation Model
  - Applicability and Limitations
  - Stateless vs. Stateful
  - IPv4-embedded IPv6 addresses and IPv4-related IPv6 addresses
- Translating from IPv4 to IPv6
  - Translating IPv4 Headers into IPv6 Headers
  - Translating UDP over IPv4
  - Translating ICMPv4 Headers into ICMPv6 Headers
  - Translating ICMPv4 Error Messages into ICMPv6
  - Transport-layer Header Translation
  - Knowing when to Translate
- Translating from IPv6 to IPv4
  - Translating IPv6 Headers into IPv4 Headers
  - Translating ICMPv6 Headers into ICMPv4 Headers
  - Translating ICMPv6 Error Messages into ICMPv4
  - Transport-layer Header Translation
  - Knowing when to Translate

Translation Model

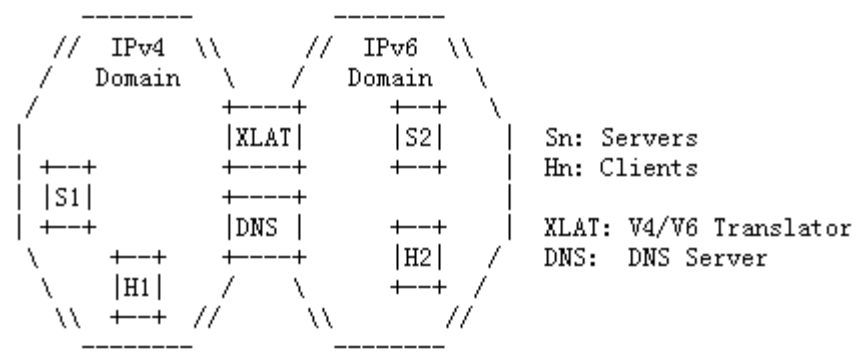

Figure 1: Translation Model

Translating from IPv4 to IPv6

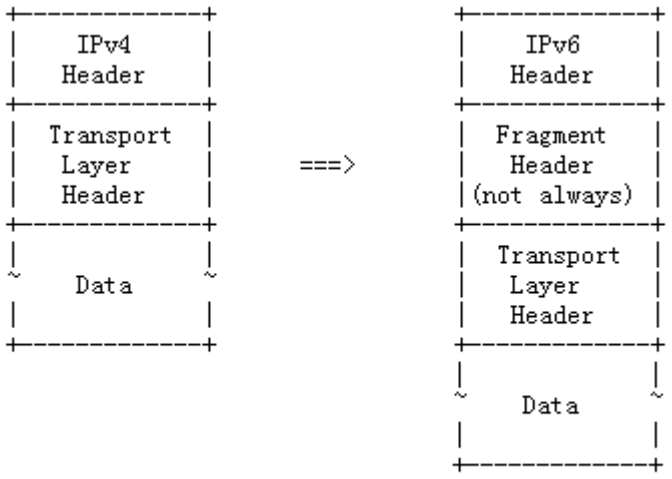

Figure 2: IPv4-to-IPv6 Translation

- Transport layer checksum issues
  - In general, recalculate
  - For stateless, we can make it checksum neutral
  - Unfragmented UDP IPv4 packet and the checksum field is zero, if<1280, leave it untouched, if >1280, recalculate (configurable).
- Path MTU issue
  - DF=1
    - Send back ICMP packet too big
  - DF=0
    - Packet size fragment to 1280
    - Add fragmentation header

Translating IPv4 Headers into IPv6 Headers

- Source Address:
  - The source address is derived from the IPv4 source address to form an IPv4-derived IPv6 address.
- Destination Address:
  - In stateless mode, which is to say that if the IPv4 destination address is within the range of the stateless translation prefix, the destination address is derived from the IPv4 destination address.
  - In stateful mode, which is to say that if the IPv4 destination address is not within the range of the stateless translation prefix, the IPv4-related IPv6 address and corresponding transport layer destination port are derived from the database reflecting current session state in the translator.

Translating ICMPv4 Headers into ICMPv6 Headers

- All ICMP messages that are to be translated require that the ICMP checksum field be updated as part of the translation since ICMPv6 unlike ICMPv4 has a pseudo-header checksum just like UDP and TCP.
- In addition all ICMP packets need to have the Type value translated and for ICMP error messages the included IP header also needs translation.

Translating ICMPv4 Error Messages into ICMPv6

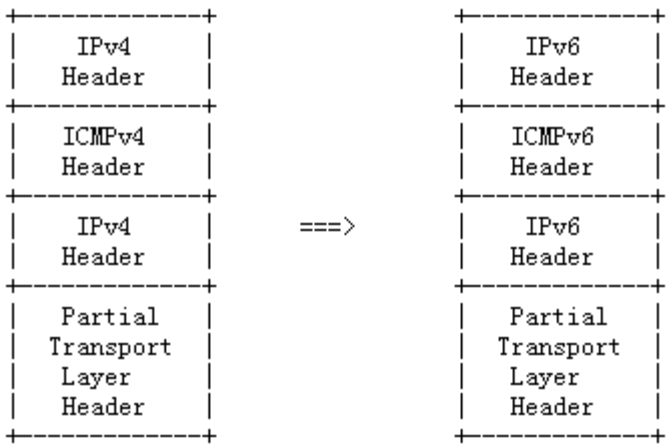

Figure 3: IPv4-to-IPv6 ICMP Error Translation

Translating from IPv6 to IPv4

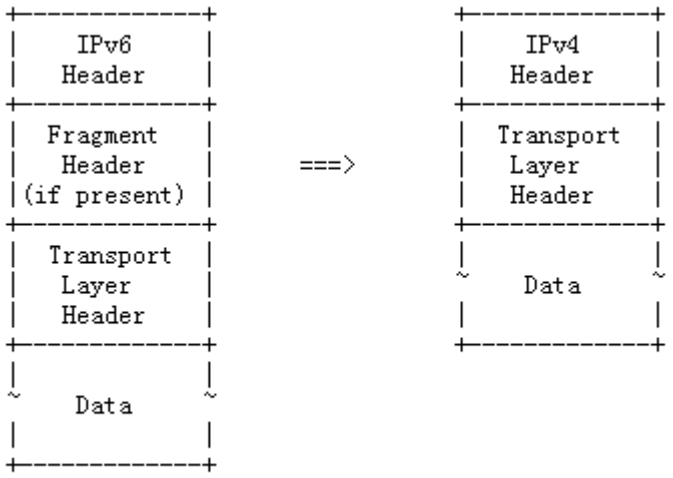

Figure 4: IPv6-to-IPv4 Translation

- Transport layer checksum issues
  - In general, recalculate
  - For stateless, we can make it checksum neutral
- Path MTU issue
  - RFC2460
    - However, [RFC2460] section 5 requires that IPv6 nodes handle such an ICMP "packet too big“ message by reducing the path MTU to 1280 and including an IPv6 fragment header with each packet.

Translating IPv6 Headers into IPv4 Headers

- Source Address:
  - In stateless mode, which is to say that if the IPv6 source address is within the range of the stateless translation prefix, the source address is derived from the IPv4-derived IPv6 address.
  - In stateful mode, which is to say that if the IPv6 source address is not within the range of the stateless translation prefix, the IPv4 source address and transport layer source port corresponding to the IPv4-related IPv6 source address and source port are derived from the database reflecting current session state in the translator.
- Destination Address:
  - The IPv4 destination address is extracted from the IPv4-derived destination address of the datagram being translated.

Translating ICMPv6 Headers into ICMPv4 Headers

- All ICMP messages that are to be translated require that the ICMP checksum field be updated as part of the translation since ICMPv6 unlike ICMPv4 has a pseudo-header checksum just like UDP and TCP.
- In addition all ICMP packets need to have the Type value translated and for ICMP error messages the included IP header also needs translation.

Translating ICMPv6 Error Messages into ICMPv4

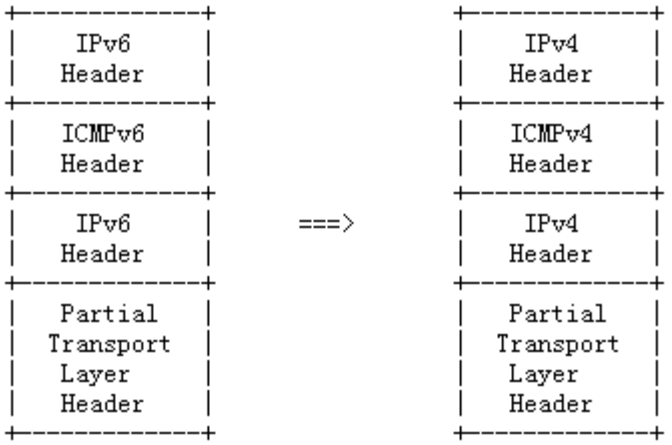

Figure 5: IPv6-to-IPv4 ICMP Error Translation

Discussions from the list

- Fragmentation and MTU
  - Fragmentation
  - TCP MSS
- Higher-layer checksum handling
- Translator sending ICMP error message
- Translating ICMP extensions

Fragmentation and MTU (1)

From IPv4 to IPv6

	IPv4	IPv6
MTU	1500	1500
Fragmentation Handling	DF=1	D. Send ICMP (packet too big) to IPv4 source address if the IPv4 packet exceeds 1480.
	DF=0	E. Fragment the IPv4 packet so that it fits in 1280 byte IPv6 packet (with fragment header extension, generate identification value).
	Fragmented packets	F. Fragmentation header extension copy identification value).
ICMP handling	ICMP fragmentation needed but DF set	G. ICMPv6 (packet too big, the original advertised MTU+20). If the advertised MTU in "packet too big" message is smaller than 1280 bytes, the value put into the translated "packet too big" message is 1280.
TCP MSS handling	First SYN packet	H. Rewrites the MSS (MSS=MSS-20)

Fragmentation and MTU (2)

From IPv6 to IPv4:

	IPv6	IPv4
MTU	1500	1500
Fragmentation Handling	DF=1	A. No fragmentation action. No ICMPv6 action. Set DF=0 if the IPv6 packet is smaller than 1280 and set DF=1 if the IPv6 packet is larger than 1280.
	Fragmentation header extension	B. Fragmented packets (if MF=0 and Offset=0, set DF=0; otherwise set DF=1, copy identification value)
ICMP handling	ICMPv6 (packet too big)	C. ICMP (packet too big, the original advertised MTU-20)

Higher-layer checksum handling

- For arbitrary addresses, the TCP and UDP transport layer checksum must be recalculated based on [RFC0793] and [RFC0768], respectively.
- For arbitrary addresses, the DCCP transport layer checksum must be recalculated based on [RFC4330]. Note that DCCP checksum may covers any application data, part of the application data, or perhaps no application data.
- For unknown transport layer protocols, the translator should pass them along unmodified.
- If an operator chooses an IPv6 address for a host served by a translator where the one's complement addition of the 16-bit words in this IPv6 address and those in the translator prefix equal zero (0x0000 or 0xffff) then, as far as the checksum is concerned, all protocols that use a one's complement checksum will pass through the translator successfully, even if the protocol in question is unknown to the translator.

Translator sending ICMP error message

- If the packet is discarded, then the translator SHOULD be able to send back an ICMP message to the original sender of the packet, unless the discarded packet is itself an ICMP message. The ICMP message, if sent, has a type of 3 (Destination Unreachable) and a code of 13 (Communication Administratively Prohibited).
- The translator device MUST allow to configure whether the ICMP error messages are sent, rate-limited or not sent.

Translating ICMP extensions

- If the ICMP extension [RFC4884] isn't translated, there are two cases for the length field modifications.
  - That the translated packet is created from scratch and the length field never is filled in. Then an ICMP extension will result in that it will be treated as part of the original datagram field.
  - If the IP payload is copied and then modified then the length field will be unmodified while the original datagram field will become longer by the address translation from v4->v6. Thus cutting off the end of the original datagram field for ICMP extension aware receivers.