NAT64

draft-ietf-behave-v6v4-xlate-stateful-01

Marcelo Bagnulo, Iljitsch van Beijnum

BEHAVE WG meeting – IETF75
A couple of open issues to discuss

• Establishing and discarding TCP mappings
• Fragmentation and PMTUD
Managing TCP mappings

- No tracking of the sequence number
- See the next slide
CLOSED

V4 SYN

V4 SYN RCV

6 secs

V6 SYN

V6 SYN RCV

4 min

V4 SYN

V6 SYN

ESTABLISHED

RST RCV

DATA SEG

RST

no packet for 2 hrs.

V4 FIN

V6 FIN

V6 FIN

V6 FIN+

V4 FIN

V4 FIN

V4 FIN

V6 FIN

V6 FIN

NAT64 state machine

4 min

4 min T.O

4 min
Receiving V4 SYN

• If a V4 SYN packet is received
  – silently drop if required by security policy requires, else,
  – If the destination transport address is not in TCP BIB, then the packet is discarded and ICMP error back
  – If the destination transport is in TCP BIB, a new session table entry is created
    • The lifetime of the entry is set to 6 seconds as per [RFC5382].
    • The packet is discarded.
MTU and fragmentation

- IPv4-to-IPv6 path MTU discovery
- PMTUD on the IPv6 side
- The IPv4 identification field value
- Fragmentation handling
v4v6 PMTUD classic

• Issue: < 1280 MTU on the IPv4 side
• RFCs 2765 and 2460: IPv6 host sends 1280-byte packets with fragment header
• Translate v6 pkts with frag header to DF=0
• Can create a PMTUD blackhole if:
  – IPv6 host disables PMTUD by setting MTU=1280 and filtering "too big" msgs
v4v6 PMTUD in draft

- The fragment header provides no useful function, so:
  - IPv4 "too big" msgs with < 1280 are translated into IPv6 "too big" with = 1280
  - All IPv6 packets ≤ 1280 are translated to IPv4 with DF=0
  - > 1280 translated to IPv4 with DF=1
Middle ground

• Stick to RFC 2765 / 2460 behavior by leaving too bigs intact
• But translate IPv6 packets ≤ 1280 DF=0
• > 1280 to DF=1
• This avoids the potential black hole
IPv6-to-IPv4 PMTUD

- Our draft: handle this locally in the NAT64:
  - NAT64 knows an IPv6 host has MTU x
  - IPv4 packets are translated, if larger than x the NAT64 fragments

- Other option: translate the "too big" msgs
  - but: many PMTUD black holes in IPv4
IPv4 identification

• Stateful translator can't copy (lower bits of) IPv6 identification field (if present):
  • Multiple IPv6 host may use the same identification values
  • So NAT64 must locally generate IPv4 identification values for ALL IPv4 packets
Translating fragments

- Reassemble and translate vs translating the fragments
- Latter is more efficient: no packet buffering in the (common) in-order case
- In IPv6-to-IPv4 direction fragments can be translated without matching session state if ID values are kept consistent (= per-packet translation state but no buffering)