IPv6 Fragment Retransmission

IETF112 6MAN Working Group (11/09/2021)
Fred L. Templin (Boeing) (fltemplin@acm.org)
draft-templin-6man-fragrep
Motivation

• Some applications see greater performance sending IPv6 packets larger than path MTU:
  • NFS/UDP, LTP/UDP, QUIC/UDP (?), IPv6 tunnels
  • Readily demonstrated with iperf3

• Source fragments IPv6 packets larger than path MTU

• Problem: loss unit (individual fragment) smaller than retransmission unit (entire packet)

• Objective: when possible, retransmit individual fragments to make loss unit equal retransmission unit
IPv6 Fragment Retransmission

• IPv6 fragment header includes 8-bit Reserved field (set to 0 on transmission; ignored on reception)
• Source uses Reserved field to encode fragment Ordinal value plus retransmission supported indication
• Destination maintains per-packet Ordinal checklist; requests retransmission of any missing Ordinals
• Destination sends ICMPv6 Fragmentation Report (FRAGREP); source retransmits missing fragments if possible
• Link-layer Automatic Repeat Request (ARQ) [RFC3366]: (fast and efficient; often avoids slow expensive end-to-end retransmission)
IPv6 Fragment Header

• RFC8200 IPv6 Fragment Header:

```
+---------------------------+-------------------+
<table>
<thead>
<tr>
<th>Next Header</th>
<th>Reserved</th>
</tr>
</thead>
</table>
+---------------------------+-------------------+
|     | Fragment Offset |   Res|M|     |
+---------------------------+-------------------+
| Identification   |
+---------------------------+
```

• Updated IPv6 Fragment Header:

```
+---------------------------+-------------------+
|   Next Header   |    Ordinal   | R|A| |
|-----------------|-------------|---|---|
+---------------------------+-------------------+
|     | Fragment Offset |   Res|M|     |
+---------------------------+-------------------+
| Identification   |
+---------------------------+
```

• A=1 for “ARQ Supported” (R-Reserved)

• When A=1, Ordinal encodes value between 0 – 63
ICMPv6 Fragmentation Report (FRAGREP)

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>Code</td>
<td>Checksum</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identification (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ordinal Bitmap (0) (0-31)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ordinal Bitmap (0) (32-63)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identification (1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ordinal Bitmap (1) (0-31)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ordinal Bitmap (1) (32-63)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- New ICMPv6 message (FRAGREP)
- Includes N (Identification, Ordinal Bitmap) list entries (12 octets each)
- Number of entries limited by 1280 (~102 list entries max per ICMPv6 message)
- Send multiple FRAGREPs if more needed
FRAGREP (2)

• Destination includes N FRAGREP list entries with 32-bit Identification followed by 64 bit Ordinal Bitmap

• For each Identification, examine each Ordinal Bitmap bit ‘i’:
  • i=1 means the fragment with Ordinal value ‘i’ received
  • i=0 means NO fragment with Ordinal value ‘i’ received

• When source receives FRAGREP, retransmits each per-Identification fragment for Ordinal Bitmap(i)=0 if fragment still in cache

• Source discards FRAGREP after all list entries processed
Additional Considerations

• 6-bit Ordinal plus 64-bit Bitmap limits ARQ to only first 64 fragments (any additional fragments sent best-effort same as current practice)

• Largest IPv6 packet that can undergo fragmentation is 64K and minMTU is 1280 (i.e., 64 normally more than enough)

• However, IPv6 fragments sometimes traverse IPv4 networks (e.g., via tunneling) with smaller minMTU (i.e., more than 64 may be needed)

• Source needs means to provide “soft” Packet Too Big (PTB) feedback meaning packet was sent but originator should reduce size of future packets - especially important when source is IPv6 tunnel ingress
Additional Considerations (2)

- ICMPv6 PTB “Soft Error” indicated by setting Code field to non-zero
- IPv6 tunnel ingress both forwards packet using IPv6 fragmentation and returns an ICMPv6 PTB “Soft Error”
- Original source receives soft error and reduces size of future packets, while original packet will likely arrive at final destination
- ICMPv6 PTB “Soft Errors” provide dynamic feedback so original source can tune packet sizes to ensure optimum performance with little or no loss
Additional Considerations (3)

• Is an additional integrity check necessary?
  • Each IPv6 fragment will undergo link-layer CRCs on the path, as well as transport-layer checksums at final destination
  • For pure IPv6 paths, this provides sufficient integrity assurance since IPv6 fragmentation includes 32bit Identification and other reassembly safeguards
  • For IPv4 paths (or mixed IPv6/IPv4 paths), IPv4 fragmentation only includes a 16bit Identification and no safeguards meaning reassembly errors possible
  • For this reason, IPv6 fragments that may be transported over IPv4 networks require an additional integrity check inserted by the IPv6 tunnel ingress and verified by the IPv6 tunnel egress
Draft Status

• ‘draft-templin-6man-fragrep’ (currently personal draft)
• 6MAN Working Group Item?