

Memory-efficient fragment forwarding with Virtual Reassembly Buffers

draft-bormann-lwig-6lowpan-virtual-reassembly

Carsten Bormann <cabo@tzi.org>

with help from the 6lo Fragmentation design team

Slides mostly provided by Thomas Watteyne — thanks!

Outline

- Review of standards
 - RFC4944
 - RFC6282
- Problem statement
- Reviewing Virtual Reassembly Buffers (VRB)
- Documents and Working Groups

RFC4944 (“6LoWPAN”)

- Link-layer fragmentation only in route-over
→ reassembly at each hop
- Fragment header

- +1 on each new frag
- No initial value specified

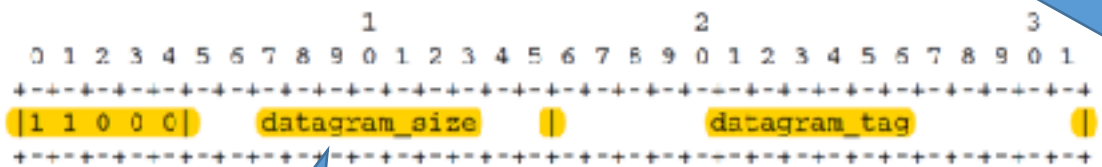


Figure 4: First Fragment

bytes

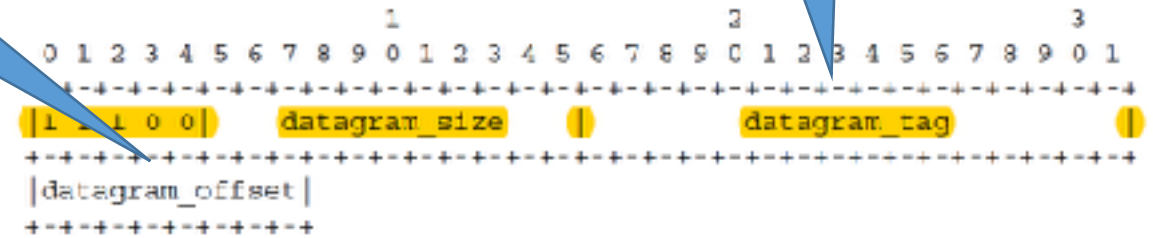


Figure 5: Subsequent Fragments

- Reassembly timer:
 - Starts when node receives first fragment
 - Timeout value MUST be <60s
 - When times out, buffer cleared, packet dropped

Pattern	Header Type	
00	000000	Not a 6LoWPAN frame
01	000001	Uncompressed IPv6 Addresses
01	000010	Compressed IPv6 Addresses
01	000011	Reserved for future use
...	Reserved	Reserved for future use
01	001111	Reserved for future use
01	010000	6LoWPAN_BCP
01	010001	Reserved for future use
...	Reserved	Reserved for future use
01	111110	Reserved for future use
01	111111	ESC
10	000000	Mesh Header
...	Reserved	Reserved for future use
11	000000	Fragmentation Header (initial)
11	001000	Reserved for future use
...	Reserved	Reserved for future use
11	011111	Reserved for future use
11	100000	Fragmentation Header (subsequent)
11	101000	Reserved for future use
...	Reserved	Reserved for future use
11	111111	Reserved for future use

Figure 3: Dispatch Value Bit Pattern

RFC6282 (“6LoWPAN-HC”)

Section 5.3 of [RFC4944] also defines how to fragment compressed IPv6 datagrams that do not fit within a single link frame. Section 5.3 of [RFC4944] defines the fragment header's datagram_size and datagram_offset values as the size and offset of the IPv6 datagram before compression. As a result, all fragment payload outside the first fragment must carry their respective portions of the IPv6 datagram before compression. This document does not change that requirement. When using the fragmentation mechanism described in Section 5.3 of [RFC4944], any header that cannot fit within the first fragment MUST NOT be compressed.

Problem statement

- Per-hop fragmentation and reassembly has 2 issues:
 - Latency:
 - Increases end-to-end latency as you need to wait for each fragment at each hop
 - Reliability:
 - Limited memory → limited number of buffers (1-2?) → packet dropped when new frag received and old not fully reassembled yet
 - No frag recovery: 1 frag loss == packet dropped
- Proposed solution:
 - Fragment forwarding:
 - Source fragments
 - Intermediate nodes relays
 - LBR reassembles

6LoWPAN: The Wireless Embedded Internet

a.k.a. “Zach and Carsten’s book”, Wiley 2009

2.5.2 L3 routing (“Route-Over”)

Layer-3 Route-Over forwarding is illustrated in Figure 2.6. In contrast to layer-2 mesh forwarding, layer-3 Route-Over forwarding does not require any special support from the adaptation layer format. Before the layer-3 forwarding engine sees the packet, the adaptation layer has done its work and decapsulated the packet – at least conceptually (implementations may be able to perform some optimizations by keeping the encapsulated form if they know how to rewrite it into the proper encapsulated form for the next layer-3 hop).

Note that this in particular means that fragmentation and reassembly are performed at each hop in Route-Over forwarding – it is hard to imagine otherwise, as the layer-3 addresses are part of the initial bytes of the IPv6 header, which is present only in the first fragment of a larger packet. Again, implementations may be able to optimize this process by keeping virtual reassembly buffers that remember just the IPv6 header including the relevant addresses (and the contents of any fragments that arrived out of order before the addresses).

Context: typical fragmentation implementation

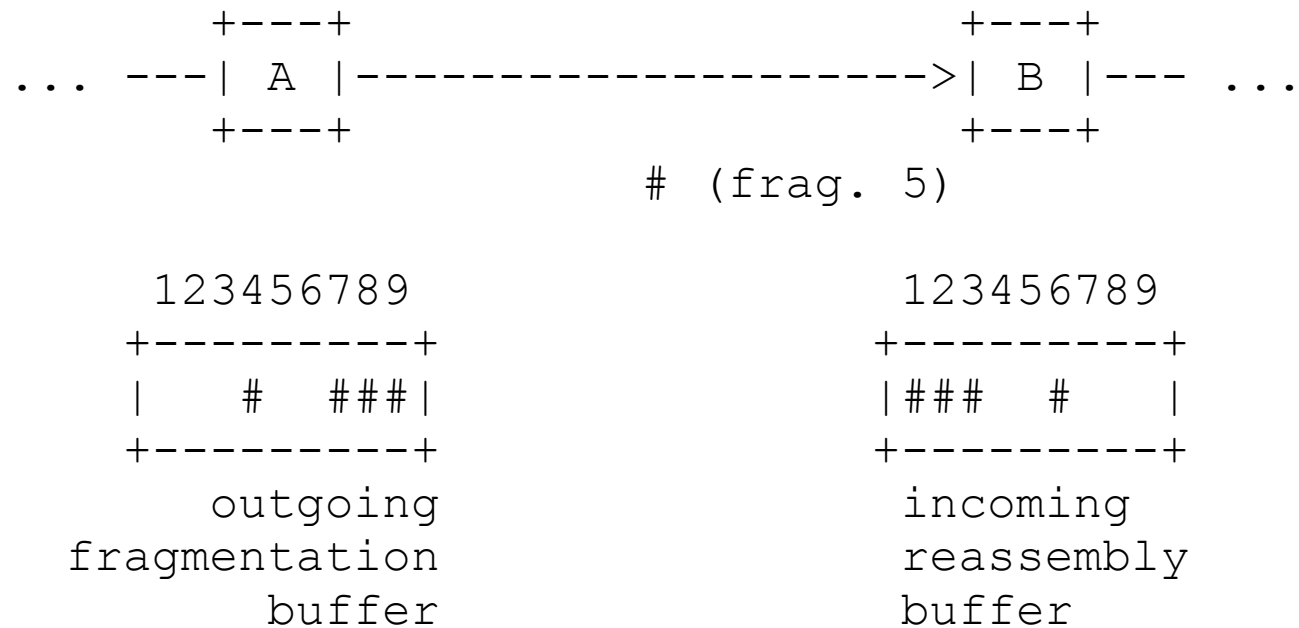
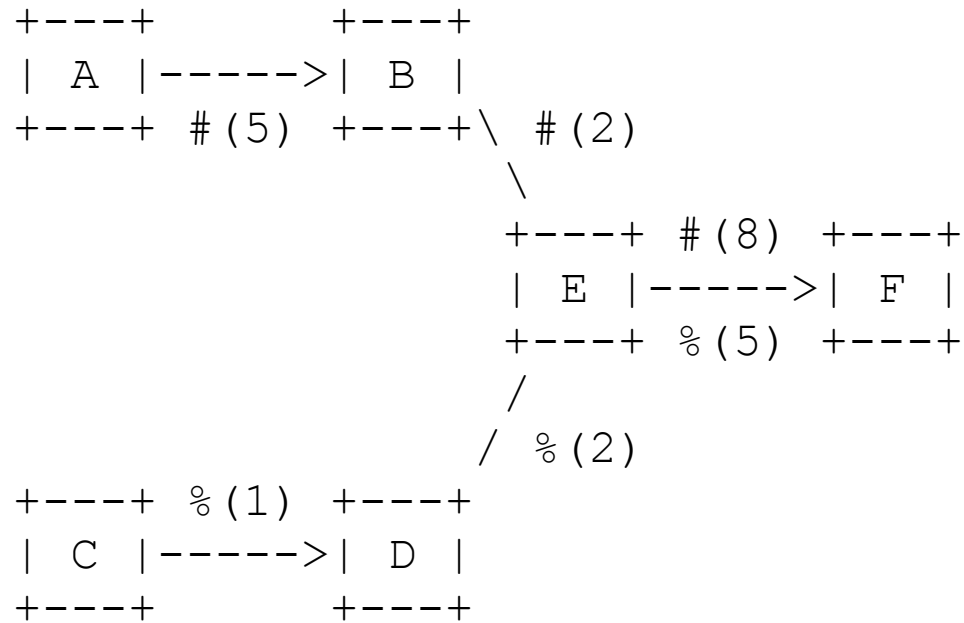


Figure 1: Fragmentation at node A, reassembly at node B.

→ Limits: latency, end-to-end reliability (*see preliminary simulation results*)

Virtual Reassembly Buffer (VRB) Implementation



incoming		outgoing	
L2 src	dtag	L2 dest	dtag
B	2	F	8
D	2	F	5
empty			
empty			

Node E's VRB table.

Figure 3: Illustrating VRB. # (5) and % (1) are fragments from packets coming from nodes A and C, with datagram_tag set to 5 and 1, respectively.

VRB: Gotchas and Limits

- Gotchas
 - Long headers (e.g. source routing) → receive multiple fragments before forwarding
 - Out-of-order fragments → receive multiple fragments before forwarding
 - Changing header length → put all slack in the frame sizes into the **first** fragment
 - Security
 - DoS attack still possible by “fragment 1” flood to overflow VRB
 - This is an implementation technique, not a protocol. Limits:
 - Non-zero Packet Drop Probability
 - No Fragment Recovery
 - No Per-Fragment Routing
- If limits are not acceptable you need an actual protocol (such as draft-thubert-6lo-fragment-recovery) — out of scope for LWIG

6lo fragmentation design team

- Produce 2 documents (to be submitted to 6lo WG):
 - informational document
 - summarize fragmentation as standardized now
 - describes Carsten's virtual reassembly buffer implementation
 - discusses its limits
 - standards-track document
 - builds upon the first one
 - adds fragment recovery
- Philosophy
 - keep activity as swift as possible
 - small DT, but regular information to WGs
 - ideally close the DT after London (*to be discussed*)

draft-watteyne-6lo- minimal-fragment-00

Thomas Watteyne

Carsten Bormann

Pascal Thubert

ToC

	1. Overview of 6LoWPAN Fragmentation	2
context	2. Limits of Per-Hop Fragmentation and Reassembly	3
	2.1. Latency	4
	2.2. Memory Management and Reliability	4
VRB	3. Virtual Reassembly Buffer (VRB) Implementation	4
	4. Critique of VRB	6
	5. Security Considerations	7
	6. IANA Considerations	7
	7. Acknowledgments	7
	8. Informative References	8
	Authors' Addresses	8

6lo vs. lwig

- VRB discussion was first published as draft-bormann-lwig-6lowpan-virtual-reassembly
- Original intention was to have everything in draft-watteyne-6lo-minimal-fragment
- Proposal now is to:
 - Keep a focused document just on the VRB implementation technique in LWIG (draft-...-lwig-6lowpan-virtual-reassembly)
 - Put the analysis and overview over new protocol proposals in 6Lo (draft-...-6lo-minimal-fragment)