

UDP Options

IETF 105 – Jul 2019

Joe Touch
strayalpha.com

Pending updates

- **Wordsmithing**

- EDIT: Experimental -> standards track; Check section nums (6 should be 5.11, 6.1 should be 5.12); NOP section
- ADD: Cite cap-letter 2119 (8174); definitions for terminology
- Check DPLPMTUD interdependence
- Extend Sec 9 to discuss how user apps decide what's needed/allowed

- **OCS clarifications**

- 2 byte checksum, includes length correction, cite IP checksum RFC
- OCS field as mandatory, at front
- Calculate without TLV processing of entire options area (fill after EOL MUST be zero)
- OCS use as SHOULD send as non-zero (MUST set when UDP CS !=0, receiver checks OCS only if CS!=0?)
- Receiver MAY disable OCS check when UDP CS !=0 and enable OCS check when UDP CS==0, i.e., override

- **Options cover the entire surplus area**

- No “post-option” area

- **Long options**

- LEN = 255 as flag to use subsequent 2-byte length

Email threads on other issues

- Option-area structure
 - Option-wide version field
 - Whether/how to align OCS
 - Must-support flag (“drop X if...”)
- Specific options:
 - LITE format
 - FRAG format
 - PADN, etc.

Some caveats

- Sec 7, 8 provide rules for new options AND processing
 - The email proposals have not followed them
 - Changes in those sections need to happen first
 - 8 needs clarification – intent is that options don't depend on each other's contents or modify the UDP user data area
- Proposed changes are not atomic
 - There are dependencies and relationships that need to be considered as a whole

My position

- **Versioning**
 - Already supported by KIND
 - (define a new KIND, define how to parse that KIND separately), which saves space until needed
- **Must-support flag**
 - OK if it says “drop all options (i.e., the entire surplus area is ignored)”, but NOT “drop this packet” (even zero-len packets)
 - Need an viable example vs. confirm via soft-state before use
- **LITE and FRAG**
 - Need to decide what we want/need to support
 - Seems to be support to drop UDP-LITE partial coverage
 - May be useful to retain the potential for both pre and post reassembly options
- **PAD, PADN variants**
 - Easy to add PADN; impossible to match both TCP *and* IPv6 codepoints (pick one or neither)

Design goals

These are highly interdependent

1. support options
2. allow at least some options to be silently ignored by legacy receivers (to enable “optionally enhanced” exchanges)
3. allow at least some options to be “support or drop all options”
4. allow the options themselves to be protected
5. support for fragmentation/reassembly
 - With RDMA-like direct placement (swaps)
 - Without duplicate checksum coverage
6. support for MTU discovery
7. support middlebox checksum/payload length bug traversal
8. support partial checksum coverage

and...

#5 frag is critical to our driving use cases, e.g., the basis of needing options, i.e., #1 opt

#5 frag depends on #8 LITE to satisfy #2 legacy

#5 frag depends on #6 MTU to determine how to frag

#7 is important - at least for the short term - to get around middleboxes checksum computation bugs

#4 is important whenever UDP CS != 0

#3 is important to the use of some options that either cannot (eg., those that can't do #2 legacy) or should not (e.g., ACS, AE) be ignored

i.e., 1 depends on 4, 5 (which depends on 2 (which depends on 3), 6, and 8) and 7

(review) Design principles

(per inclusion/processing rules)

- Who decides what?
 - Senders decide what to add
 - Receivers decide what to require
 - Soft-state helps coordinate safe use of content-modifying options, e.g., AE
 - Ensures default is consistent with legacy
- These are options, not encapsulations
 - Add to body, but not modify or depend on it
 - Exceptions include only LITE, AE, ACS
 - Do not depend on each other
 - Exceptions include only OCS, AE
 - Cannot declare order
 - Process in the order they occur
 - EXCEPT:
 - LITE always first if present
 - Then OCS (if present)

FRAG and LITE – a deeper view

*Given changes to OCS use and coverage (previous slides),
what do new proposals offer vs. what do they give up?*

- **Benefit of existing approaches**

- LITE data is consecutive with the UDP user payload (or head of UDP user area even if partial coverage isn't needed/supported), e.g., for RDDP / zero-copy
- Supports both pre and post-reassembly options
- Supports LITE without FRAG (i.e., no overhead to hide user data, e.g., for compression)

- **Key question**

- Do we need to keep true LITE (protocol 136) behavior, i.e., partial checksum coverage

- **Non-Issues**

- Worries about FRAG without LITE
(we can force use as only FRAG+LITE and LITE, i.e., never FRAG alone)
- LITE not being covered by OCS (it is now)