Device IO Function Redesign

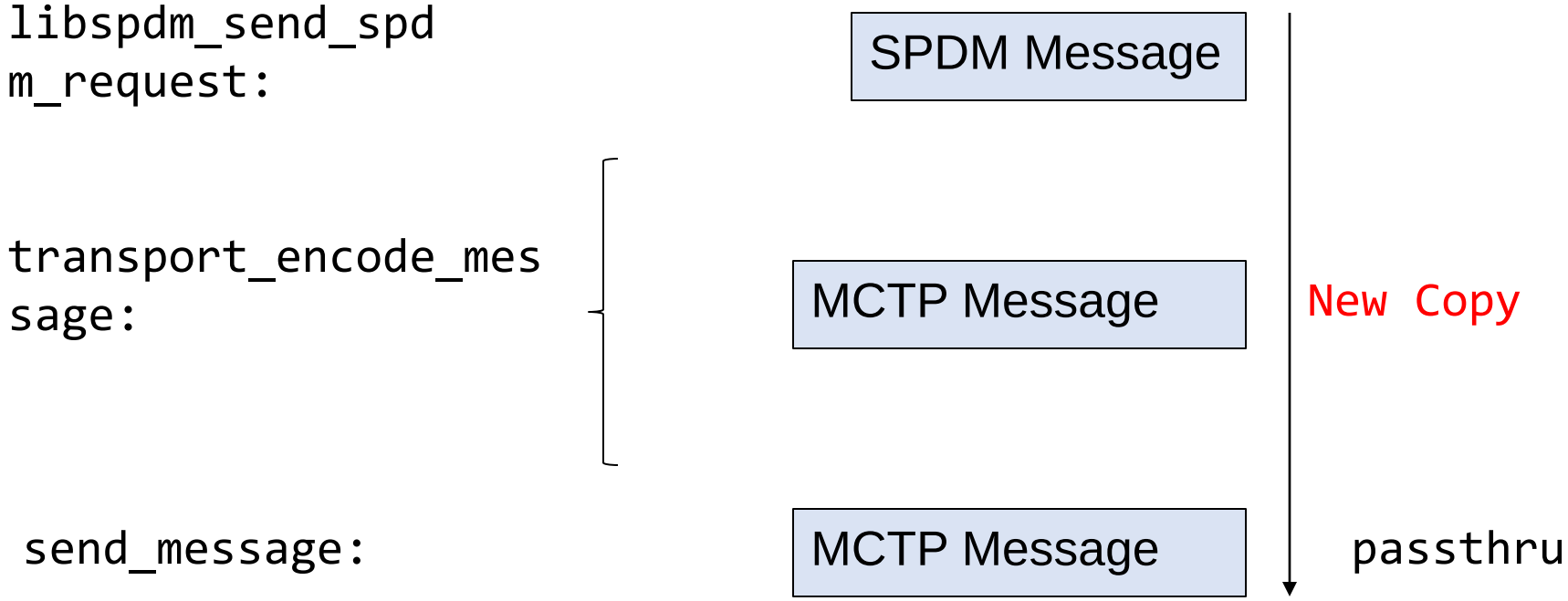
Jiewen Yao, Intel

Agenda

- Current Device IO design 🡐
- Ideal Solution
- API & flow change

- https://github.com/DMTF/libspdm/issues/351

Current Implementation (Send - NoEnc)

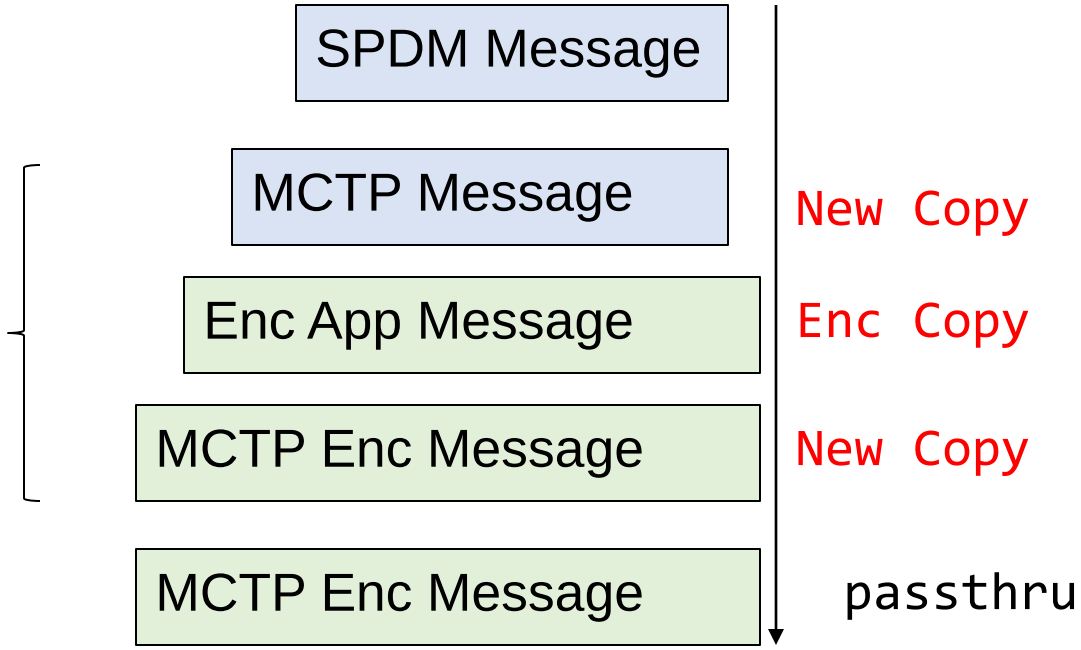


Current Implementation (Send - Enc)

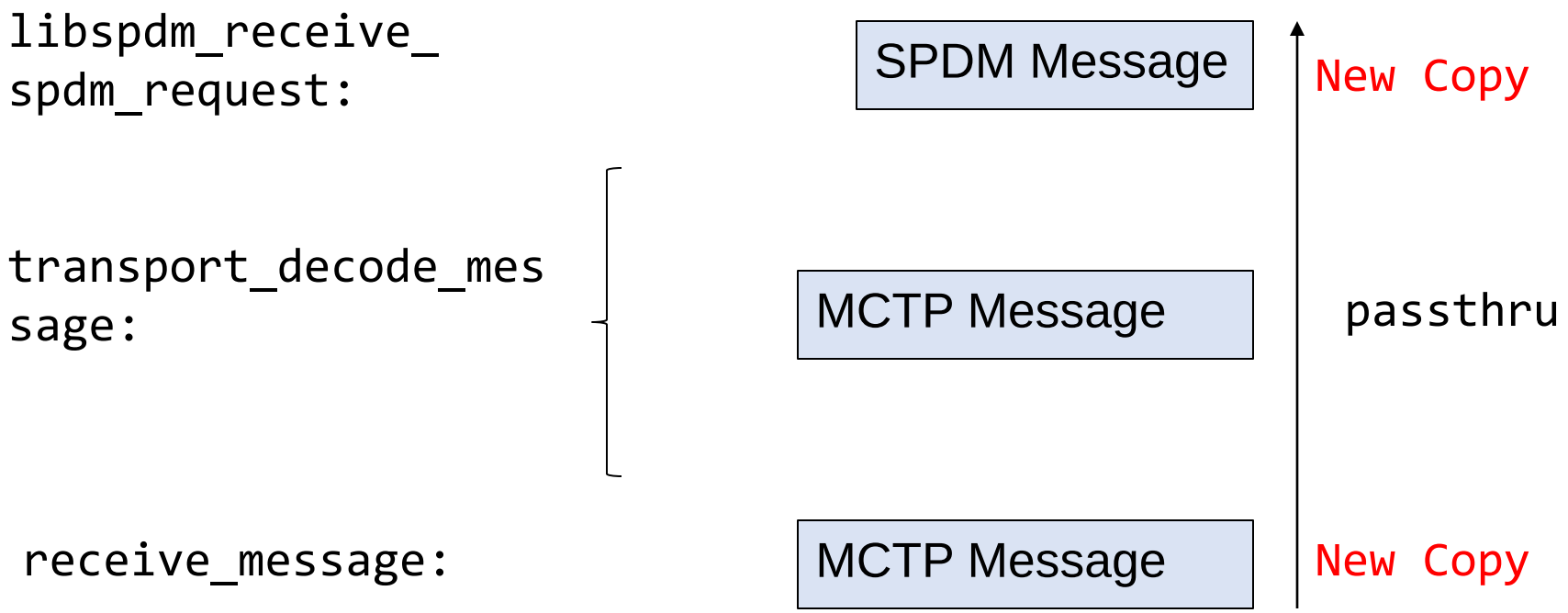
libspdm_send_spdm_request:

transport_encode_message:

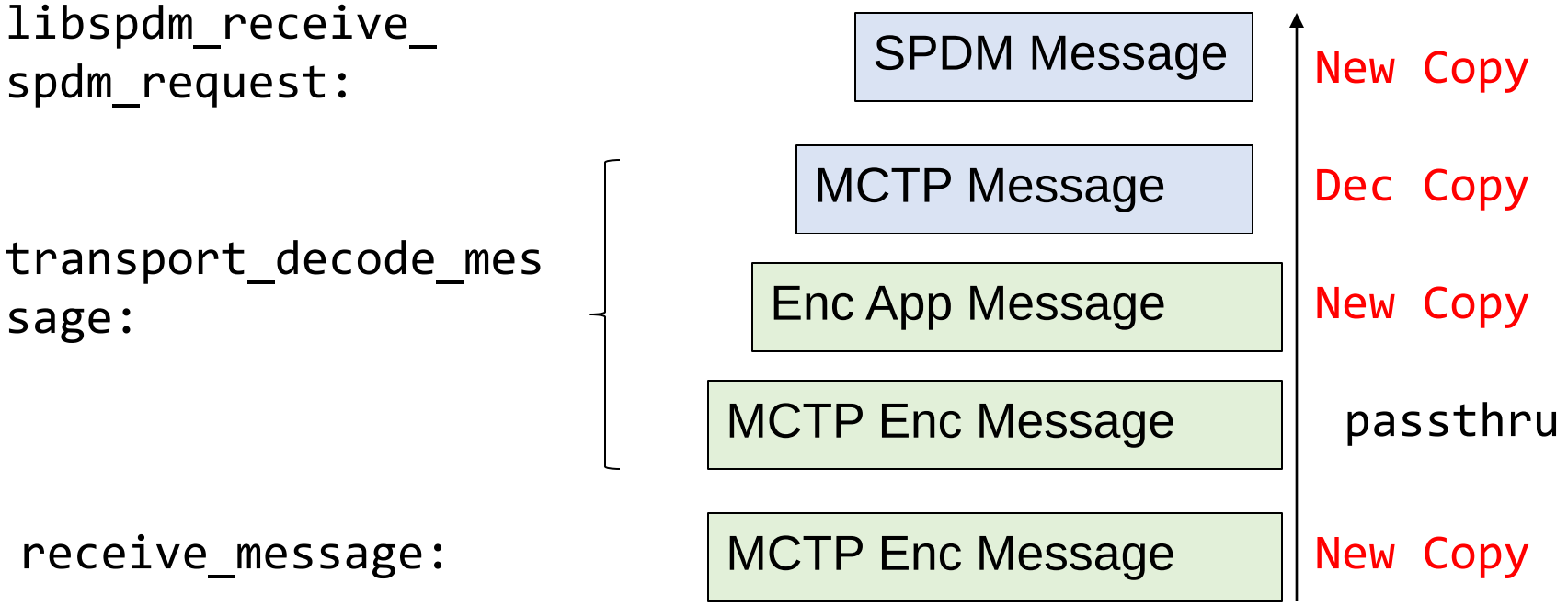
send_message:



Current Implementation (Receive - NoEnc)



Current Implementation (Receive - Enc)



Problem Statement

- Too many copies
  - Impact stack (or heap) usage
  - Impact runtime performance (potentially)

- Ideal solution
  - Minimize the copy
  - Reduce stack/heap usage
  - Integrator may choose memory type for send/receive message.

Agenda

- Current Device IO design
- Ideal Solution 🡸
- API & flow change

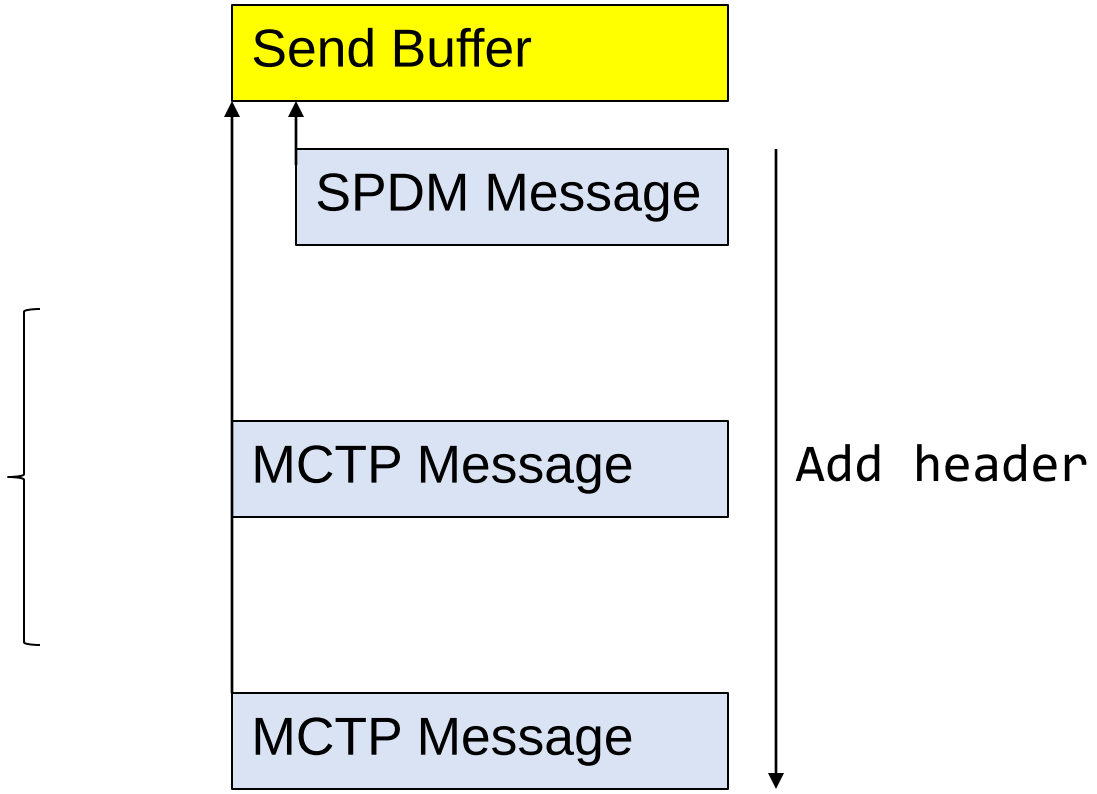
- https://github.com/DMTF/libspdm/issues/351

Ideal Implementation (Send - NoEnc)

libspdm_send_spdm_request:

transport_encode_message:

send_message:

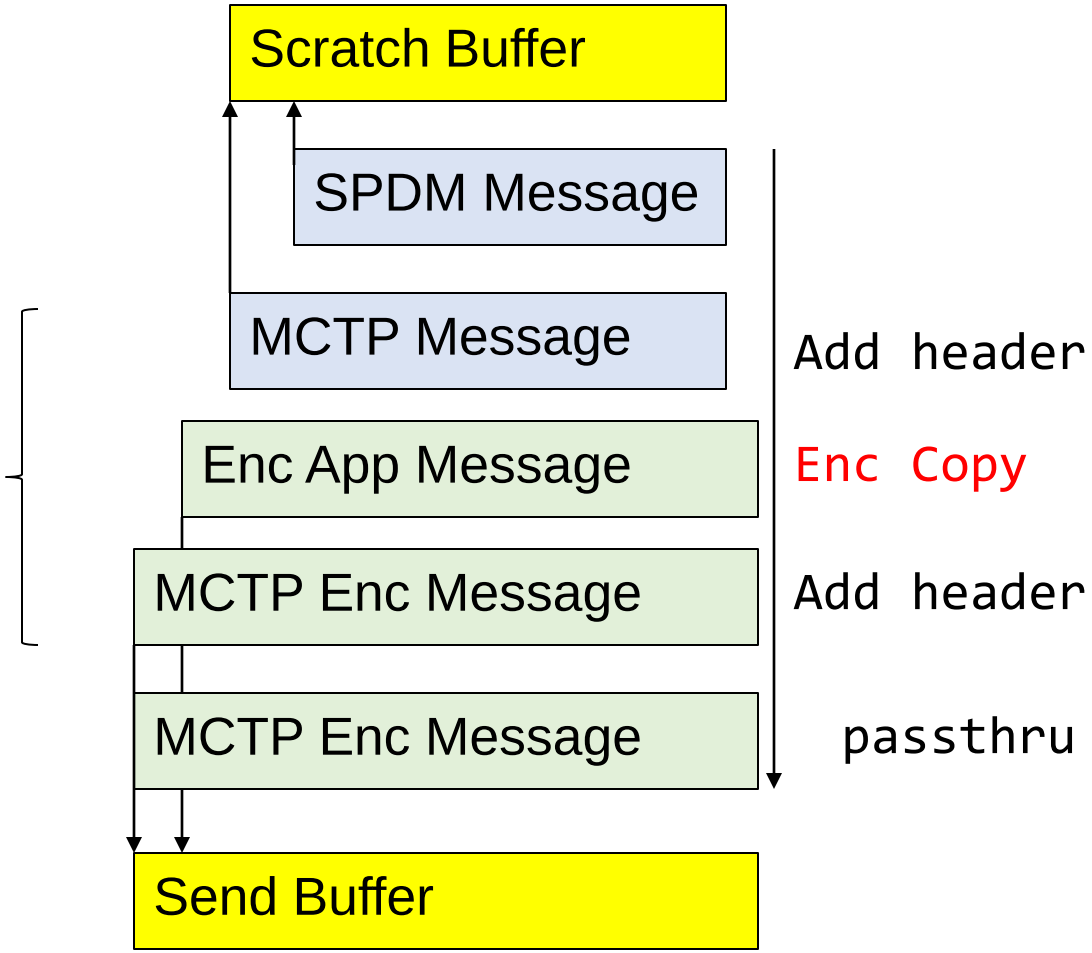


Ideal Implementation (Send - Enc)

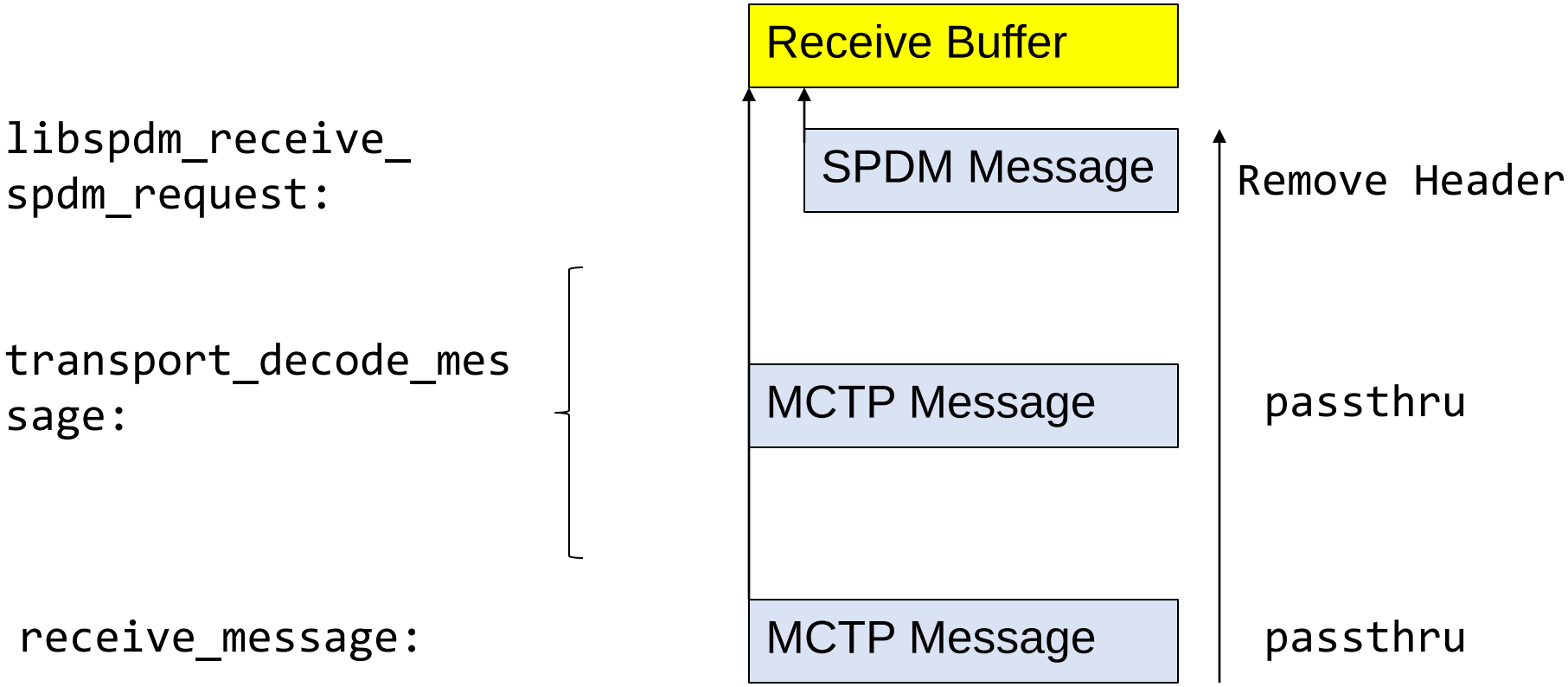
libspdm_send_spdm_request:

transport_encode_message:

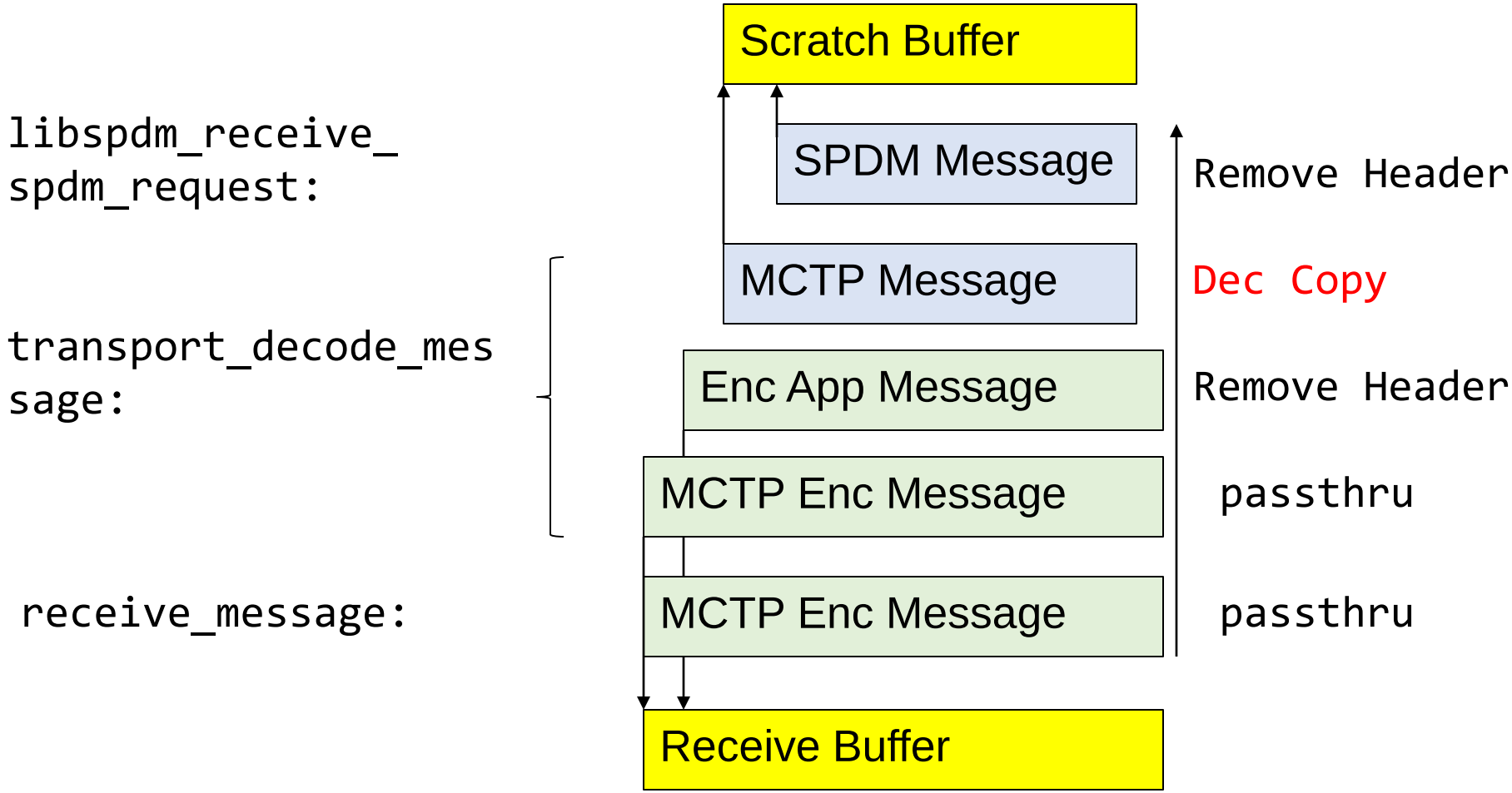
send_message:



Ideal Implementation (Receive - NoEnc)



Ideal Implementation (Receive - Enc)



Extra Copy Required

- 1) Support transcript
  - Need cache last request (append after response is returned)
  - Need cache the full transcript for LIBSPDM_RECORD_TRANSCRIPT_DATA_SUPPORT
- 2) Support retry
  - Need cache last request and last response.
- 3) Support chunking
  - Need cache the full SPDM message (larger than transport buffer).

Agenda

- Current Device IO design
- Ideal Solution
- API & flow change 🡸

- https://github.com/DMTF/libspdm/issues/351

Current Flow – Requester

- libspdm_get_xxx()
  - **libspdm_send_spdm_request** (uintn req_size, const void *request)
    - **transport_encode_message** (uintn req_size, const void *request,
      uintn *msg_size, void *message)
    - **send_message** (uintn msg_size, const void *message)
  - **libspdm_receive_spdm_response** (uintn *rsp_size, void *response)
    - **receive_message** (uintn *msg_size, void *message)
    - **transport_decode_message** (uintn msg_size, const void *message,
      uintn *rsp_size, void *response)

Changed Flow – Requester

NOTE: Final sender msg_ptr might be inside of acquired msg_buf.

1) The length of transport header might be unpredictable.
2) There is difference between normal SPDM message and secured SPDM message.

- libspdm_get_xxx()
  - **acquire_sender_buffer** (uintn *max_msg_size, void **msg_buf_ptr)
  - **libspdm_send_spdm_request** (uintn req_size, const void *req_buf_ptr)
    - **transport_encode_message** (uintn req_size, const void *req_buf_ptr, uintn *msg_size, void **msg_buf_ptr_1)
    - **send_message** (uintn msg_size, const void *msg_buf_ptr_1)
  - **release_sender_buffer** (const void *msg_buf_ptr)

  - **acquire_receiver_buffer** (uintn *max_msg_size, void **msg_buf_ptr)
  - **libspdm_receive_spdm_response** (uintn *rsp_size, void **response)
    - **receive_message** (uintn *msg_size, void **msg_buf_ptr)
    - **transport_decode_message** (uintn msg_size, const void *msg_buf_ptr, uintn *rsp_size, void **response)
  - **release_receiver_buffer** (const void *msg_buf_ptr)

Current Flow – Responder

- libspdm_responder_dispatch_message ()
  - **receive_message** (uintn *msg_size, void *message)
  - **libspdm_process_message** (uintn msg_size, const void *message, uintn *rsp_msg_size, void *rsp_message)
    - **libspdm_process_request** (uintn msg_size, const void *message)
      - **transport_decode_message** (uintn msg_size, const void *message, uintn *req_size, void *request)
      - // copy to last_spdm_request
    - **libspdm_build_response** (uintn *rsp_msg_size, void *rsp_message)
      - // call get_response_func (last_spdm_request, rsp_message)
      - **transport_encode_message** (uintn rsp_size, const void *rsponse, uintn *msg_size, void *message)
  - **send_message** (uintn msg_size, const void *message)

Changed Flow – Responder

- libspdm_responder_dispatch_message ()
  - **acquire_receiver_buffer** (uintn *max_msg_size, void **msg_buf_ptr)
  - **receive_message** (uintn *msg_size, void **msg_buf_ptr)
  - **libspdm_process_request** (uintn msg_size, const void *msg_buf_ptr)
    - **transport_decode_message** (uintn msg_size, const void *message, uintn *req_size, void **request)
    - // copy to last_spdm_request
  - **release_receiver_buffer** (const void *msg_buf_ptr)

  - **acquire_sender_buffer** (uintn *max_msg_size, void **msg_buf_ptr)
  - **libspdm_build_response** (uintn *msg_size, void **msg_buf_ptr_1)
    - // call get_response_func (last_spdm_request, rsp_msg_buf_ptr)
    - **transport_encode_message** (uintn rsp_size, const void *rsp_buf_ptr, uintn *msg_size, void **msg_buf_ptr_1)
  - **send_message** (uintn msg_size, const void *msg_buf_ptr_1)
  - **release_sender_buffer** (const void *rsp_msg_buf_ptr)

NOTE: Final sender msg_ptr might be inside of acquired msg_buf.

Open

- Can we assume sender_buffer and receiver_buffer are different?
  - NO. We cannot.

- Can we fix the location of sender_buffer and receiver_buffer at init?
  - NO need.