

# Fragmentation Algorithm based on UDP

In the Distributed Test Support System, messages exchanged through the DML may exceed the maximum datagram size of the UDP protocol of 65,515 bytes. To transmit these messages using a UDP socket, a fragmentation algorithm must be implemented by the application.

Based on the results of the investigation that unfragmented UDP datagrams are not reordered in a local network without a switch, the information required for fragmentation and defragmentation can be significantly reduced compared to the IP protocol. A proposal for a fragmentation mechanism is outlined below.

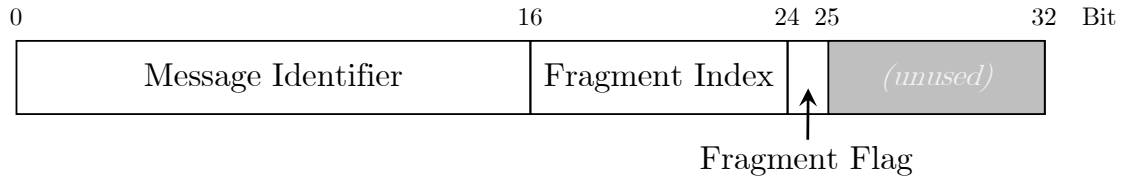


Figure 1: Proposed Header for the Implementation of a Fragmentation Mechanism

Figure 1 shows the proposed header format for the implementation of a fragmentation mechanism. The header is 4 bytes in size and is followed by the payload. The maximum payload size for a message passed to the UDP socket is determined by the MTU, as the IP protocol should not perform any fragmentation. For an MTU of 9000 Bytes, the maximum payload size is 8968 bytes. The Equation below illustrates the calculation.

$$\begin{aligned} \text{Maximum Payload} &= 9000 \text{ Bytes (MTU)} \\ &\quad - 20 \text{ Bytes (IP Header)} \\ &\quad - 8 \text{ Bytes (UDP Header)} \\ &\quad - 4 \text{ Bytes (Fragmentation Header)} \\ &= 8968 \text{ Bytes} \end{aligned} \tag{1}$$

The proposed fragmentation header (see Figure 1) includes the following information:

- **Message Identifier** (*2 Bytes*): The Message Identifier is a sequential number assigned to each message by the sender. All message fragments share the same identifier.
- **Fragment Index** (*1 Byte*): The Fragment Index is used to number the individual fragments of a message.
- **Fragment Flag** (*1 Bit*): The Fragment Flag indicates whether the message is fragmented and more fragments will follow.

The maximum size that can be handled by this proposed fragmentation algorithm is limited by the *Fragment Index*. Since this field has a size of 1 byte, it can represent 256 different values. Therefore, a message can be divided into a maximum of 256 fragments. Each fragment has a maximum payload of 8968 bytes, resulting in a maximum message size of  $256 \times 8968 \text{ Bytes} = 2,295,808 \text{ Bytes}$ .

The *Fragments Flag* is set by the sender to allow the defragmentation of a message. The flag should not be set for unfragmented messages. For fragmented messages, the flag should be set for all fragments except the last one. This allows the recipient to recognize the individual message fragments and reassemble them into a complete message. The absence of the flag on the last fragment indicates that all parts of the message have been received and defragmentation can be initiated.

The proposed mechanism also allows the detection of errors. The loss of a message, whether fragmented or unfragmented, can be detected using the *Message Identifier*. The *Fragment Index* can be used to detect the loss of fragments of a message.

Additionally, the proposed header format includes 7 unused bits that are available for possible future extensions.