Figure 1 shows the ws-reliability protocol roles as five UML object classes:

- Sender – an application which may have a sendsThru contract with a Sending RMP. The sender invokes send operations on the sendingRMP, with a message payload, and must be prepared to accept failureNotification operations invoked by its sendingRMP. Upon failure, the sender is returned the payload of the failed message.
- SendingRMP – a Sending RMP instance acts on behalf of a single Sender instance, and uses a (potentially shared) transport service to send messages and receive responses.
- Transport – this object models the underlying transport. SendingRMPs use it to send messages and receive acks and faults, and ReceivingRMPs use it to receive messges and send acks and faults.
- ReceivingRMP – a Receiving RMP instance acts on behalf of a single receiver instance, and uses a (potentially shared) transport service to receive messages and send responses.
- Receiver – an application which may have a receivesThru contract with a Receiving RMP. The receiver accepts an abstract deliver operation, which conveys the payload of the reliable message, in correct order.

For simplicity, it is assumed that there is only one group at a time (i.e., there are no groupID parameters shown in these examples, just for simplification purposes).

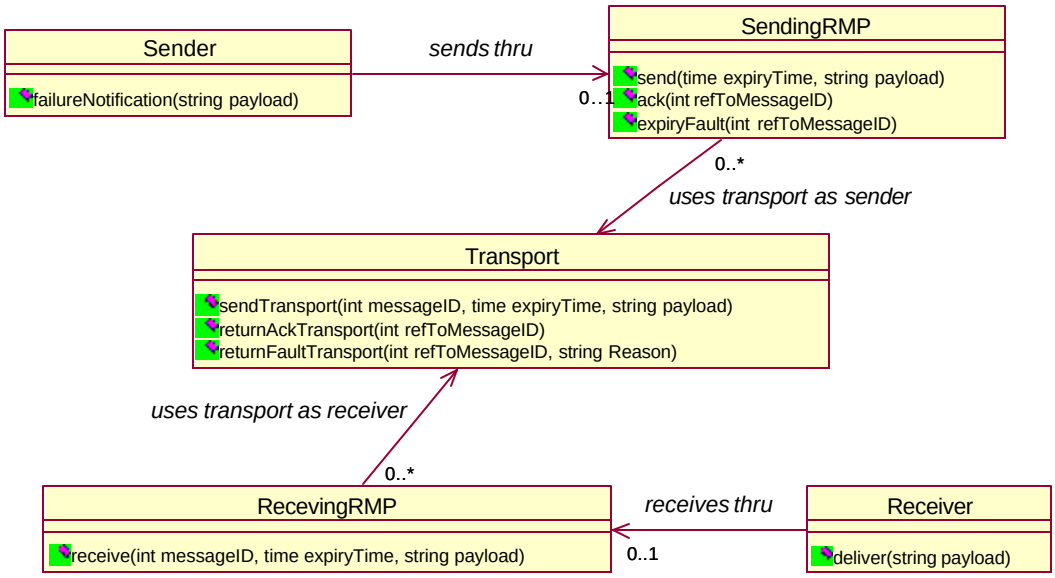

Figure 1: Protocol Roles as Object model (abstract operations and associations shown)

Figure 2 shows an example sequence diagram, which shows an example of ordered delivery, where a lost second message is retransmitted in time to complete the sequence before termination. Note that all three messages are delivered in the proper order, because the retransmitted message 2 was received before the held third message expired.

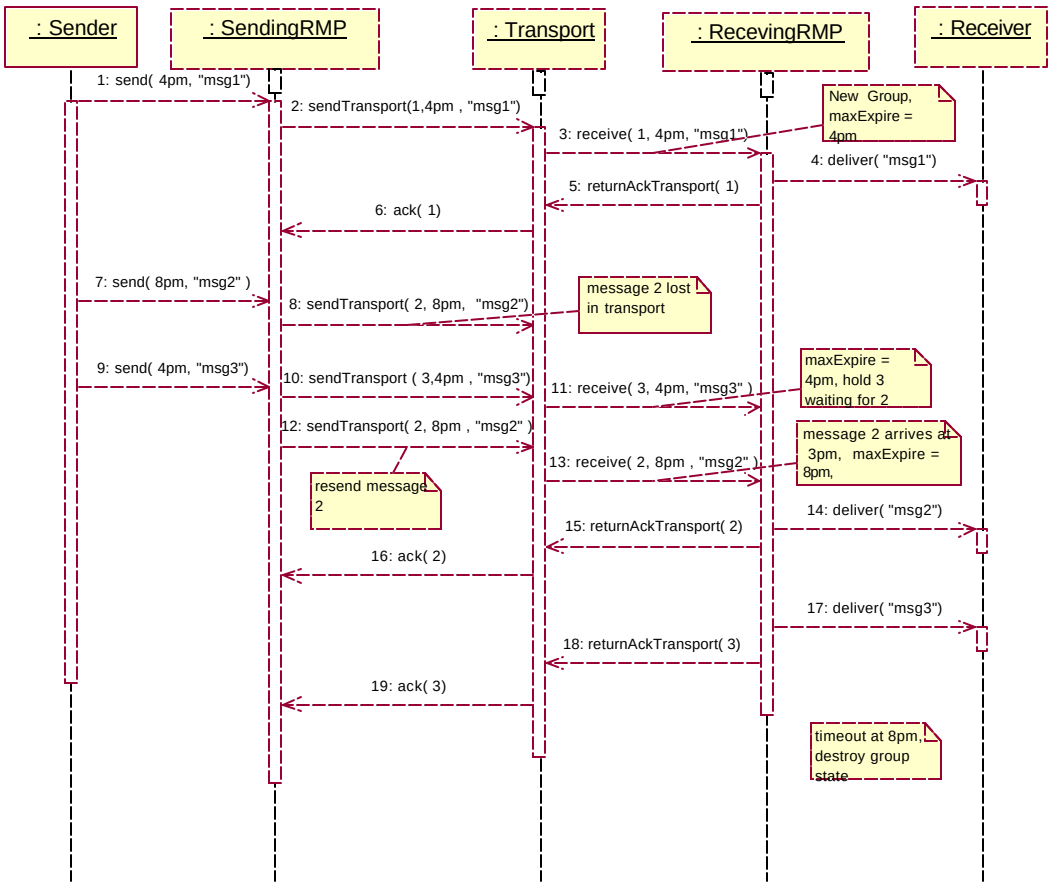

Figure 2: Sequence Diagram Example (how ordered delivery is supposed to work)

Figure 3 shows a more troublesome example, where a delayed message 2 is received after the third message times out. In this case the sender gets a failure Notification for both messages 2 and 3.

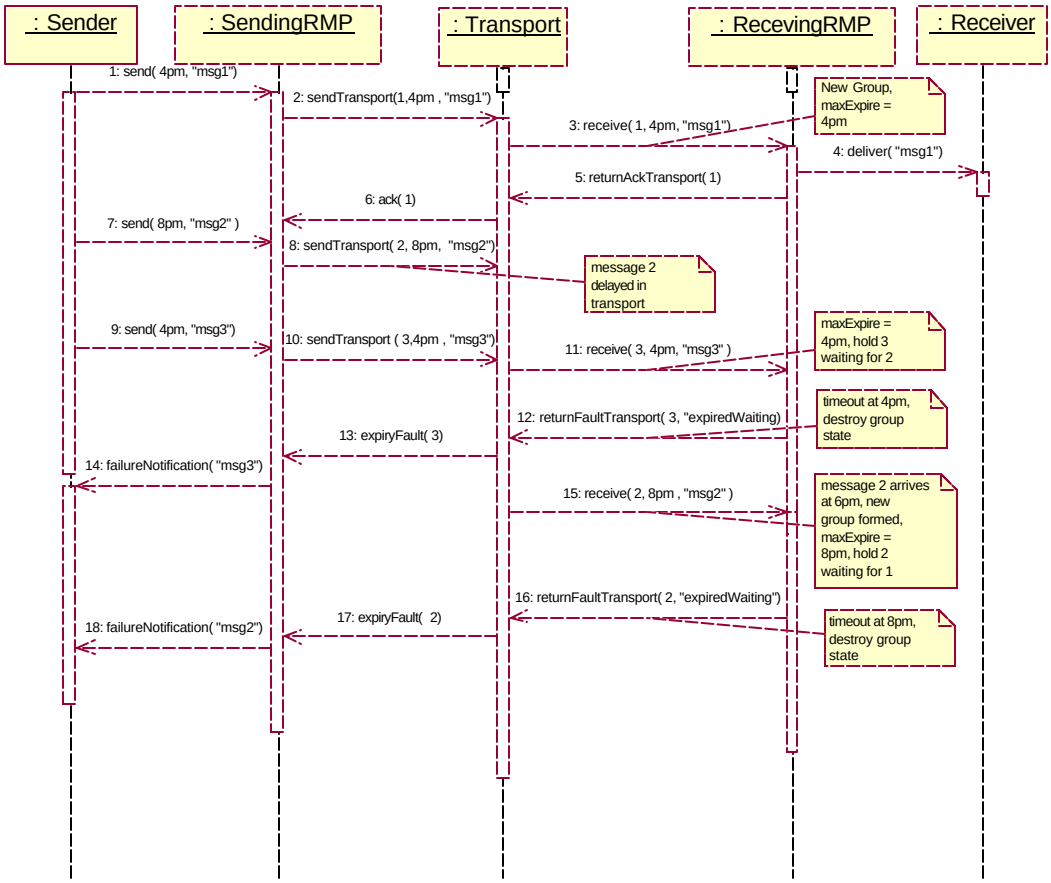

Figure 3: Sequence Diagram Example (Extensively Delayed receipt of message 2)

Figure 4 is another troublesome example, where the message 2 is retransmitted by the sending rmp after the third held message expires. However, the sender is made aware of the failure to deliver, since it receives failure notifications for both failed messages 2 and 3.

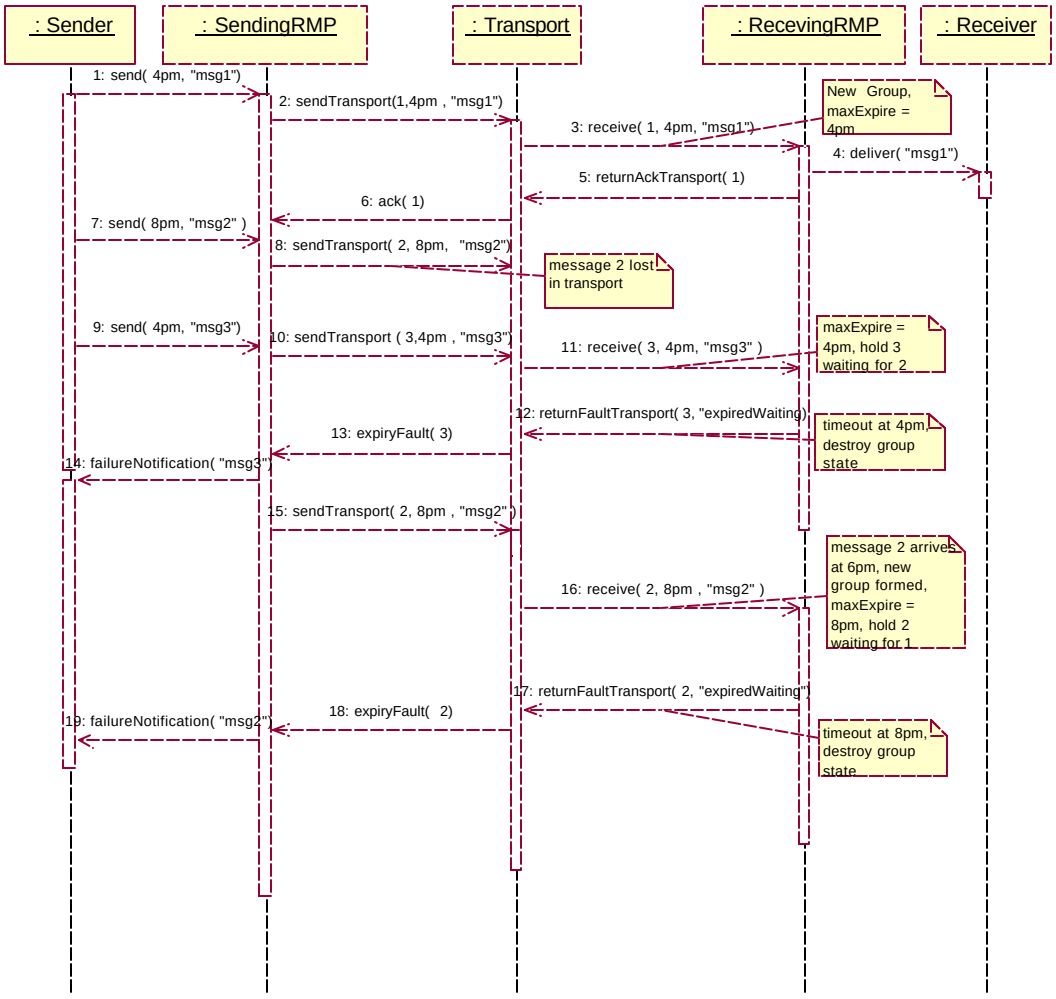

Figure 4: (Extensively Delayed Retransmission of message 2)