



Revision of ebXML Reliable Messaging Specification

November 6, 2000
FUJITSU LIMITED



Status of Reliable Messaging Spec

- ❑ ebXML TRP face-to-face Meeting on 26-28 September
 - Fujitsu updated the RM spec to v0-078 (22 September)
 - Cleared QoS definition
 - Removed RM-Group
 - Defined mapping between MS protocol and Transport protocol
 - Proposed basic idea about reliable routing
 - After the meeting, Fujitsu updated the RM spec to v0-080 (2 October) to include the meeting results
 - MS spec v0-21d included RM spec v0-080.
- ❑ Discussion in the mailing list
 - Some major issues were discussed in mailing list after the meeting
 - Message order
 - Including a business level response in MS Ack, etc.

Issue Status Summary

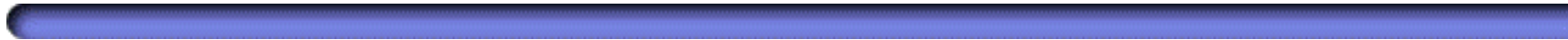
Topic	Issue Raised	Status
Removal of the blocking restriction	Message Order How to guarantee message order?	A possible idea will be proposed in this meeting
	Buffer Size How to expect required buffer size previously?	
Sequence Number	Reset of Sequence Number How to notify reset of Sequence Number?	
Ack Message	Mapping of Acknowledgment How to map between MS Ack and Transport Ack?	Fujitsu proposed a basic idea in previous meeting and RM spec
Error Message	Detailed contents of Error Message and Error Handling What kind of errors should be notified by Error Message? How to handle the errors?	
Routing	Reliability in networks that use routing How is reliability established in routing? What is the From/To scope: point-to-point or router-to-router?	
Other	Transfer of large documents How to split and transfer large documents?	A possible idea will be proposed in this meeting
	Maximum Message Size What is the maximum size of a transferred message? How should the size be defined?	



There is a possible solution



Sliding Window



Problems in Removal of the Blocking Restriction

- ❑ MS spec V0-21d does not have a blocking restriction
 - Sender can send some messages without waiting for acknowledgement messages from previous messages
- ❑ Removing the restriction causes some problems
 - How to guarantee message order?
 - How to set a required maximum buffer size in the Receiver?
- ❑ The Sliding Window algorithm, which is a traditional and widely used protocol, resolves these problems

Sequence Number for Sliding Window

- ❑ Sequence Number is an integer value in the range
0 - 99,999,999
- ❑ The size is represented by symbol “M”
 - $M = 100,000,000$
- ❑ Changed points from previous RM specification
 - Minimum value is “0” (not “1”) so that modulo operation can be used in calculation of sequence number
 - Maximum value is 99,999,999 (not 999,999,999) so that calculation of sequence number can be always executed in the range of a signed 32bit integer

Sending Window

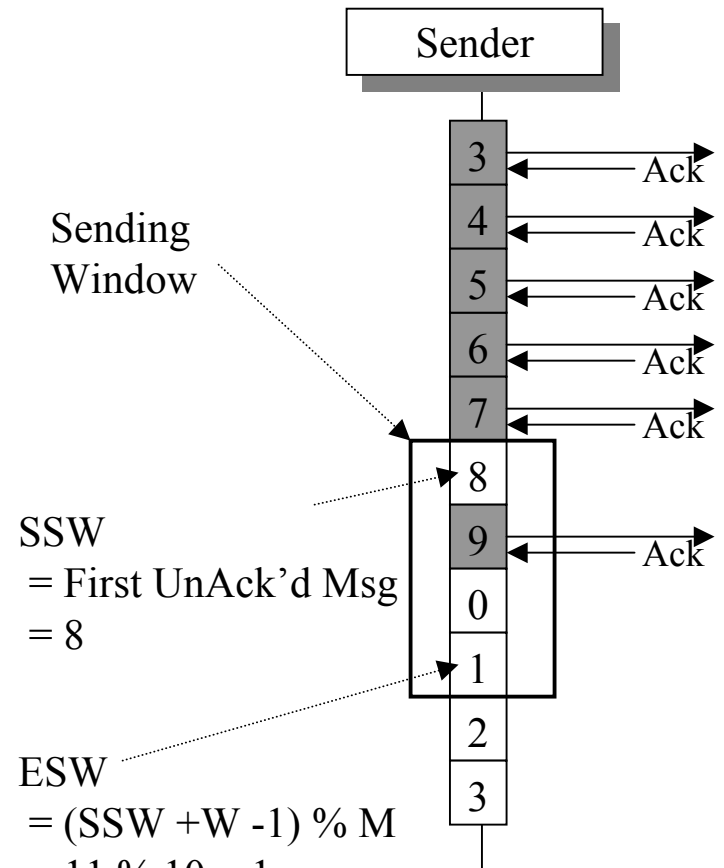
□ Sending Window

- A scope of messages which the Sender MAY send without receiving an Acknowledgement Message
- Starts from the first unacknowledged message

□ Calculation for Sending Window

- The window size is represented by symbol “W”
- Start of Sending Window
 $SSW = \text{the first unAck'd msg}$
- End of Sending Window
 $ESW = (SSW + W - 1) \% M$

Ex. $M = 10$, $W = 4$, First UnAck'd Msg = 8



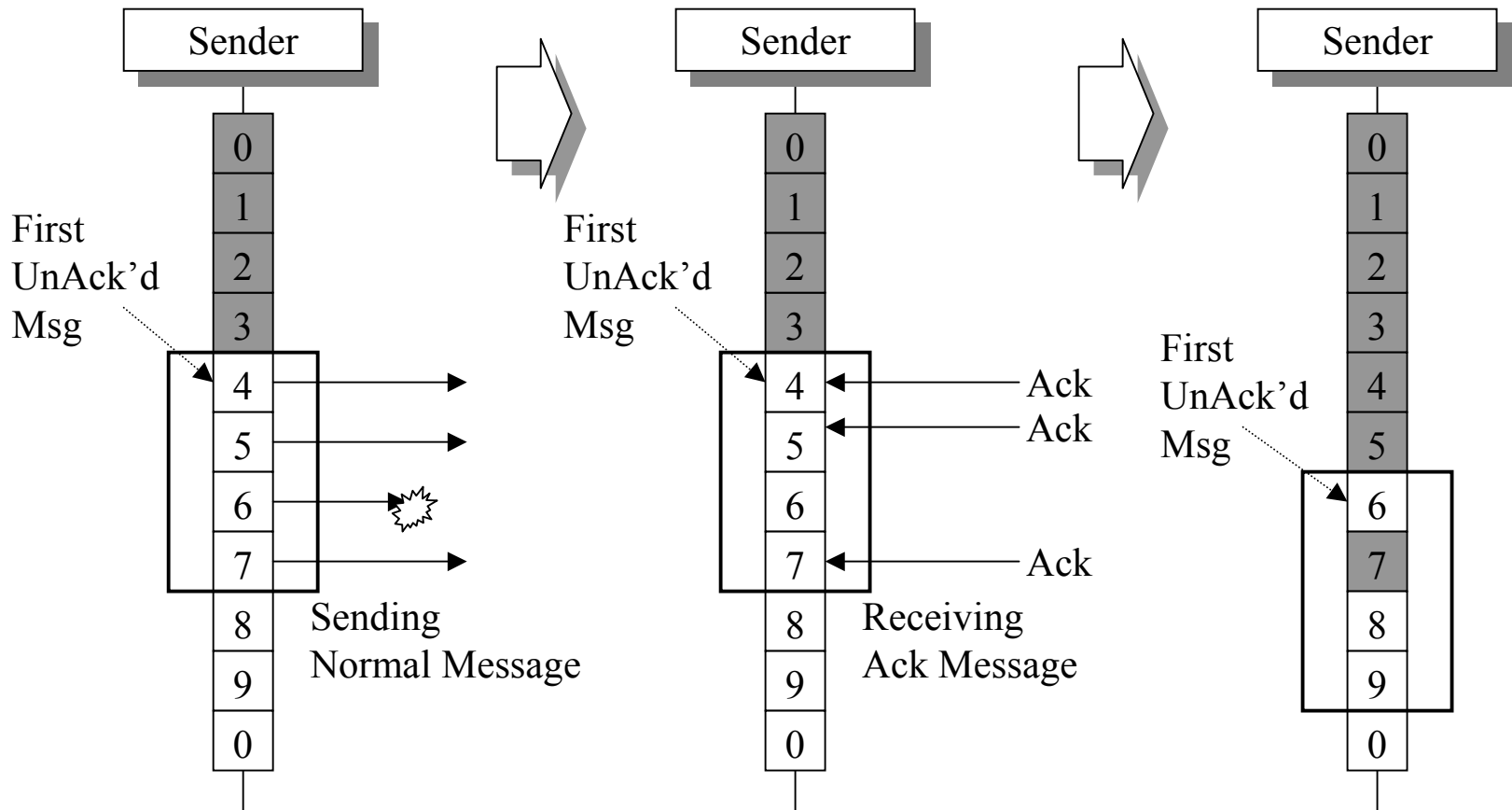
■ : Sent and Acknowledged

□ : Not Sent, or Sent but not acknowledged yet

Sliding of Sending Window

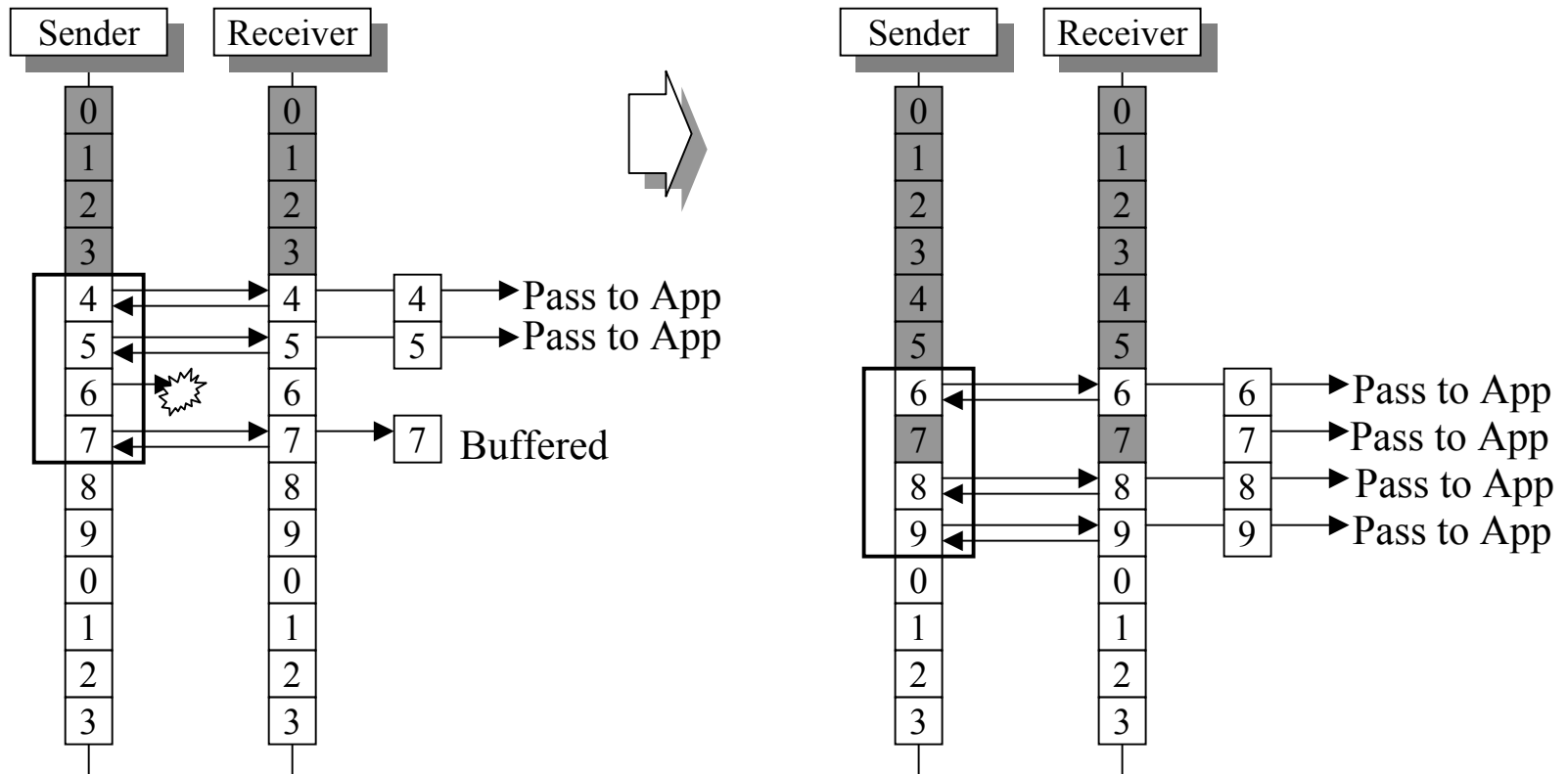
- Sending Window is slid by acknowledgement in width from 1 to W

Ex. $M = 10$, $W = 4$, Sliding width = 2



Guarantee of Message Order

- ❑ Window Size means required maximum buffer size in the Receiver
- ❑ The Receiver can correct invalid message order in the buffer using sequence numbers



Duplication Check Window

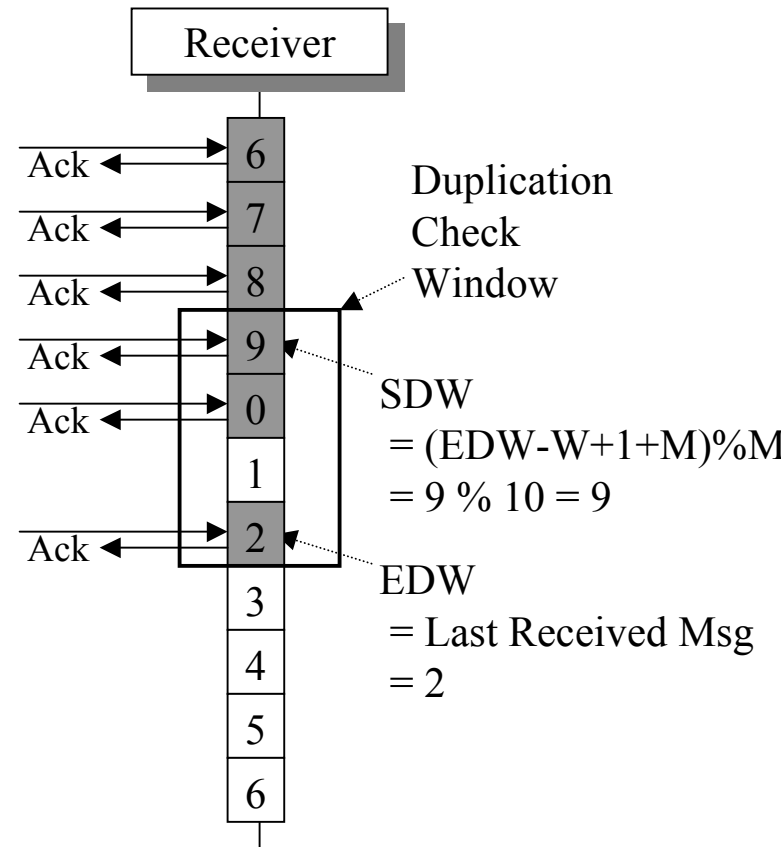
❑ Duplication Check Window

- A scope of messages which the Receiver SHALL use to check for duplication when messages arrive
- Terminates with last received message

❑ Calculation for Duplication Check Window

- The size is W (same as Sending Window)
- Start of Duplication check Window
 $SDW = (EDW - W + 1 + M) \% M$
- End of Duplication check Window
 $EDW =$ the last received message

Ex. $M = 10$, $W = 4$, Last Received Msg = 2



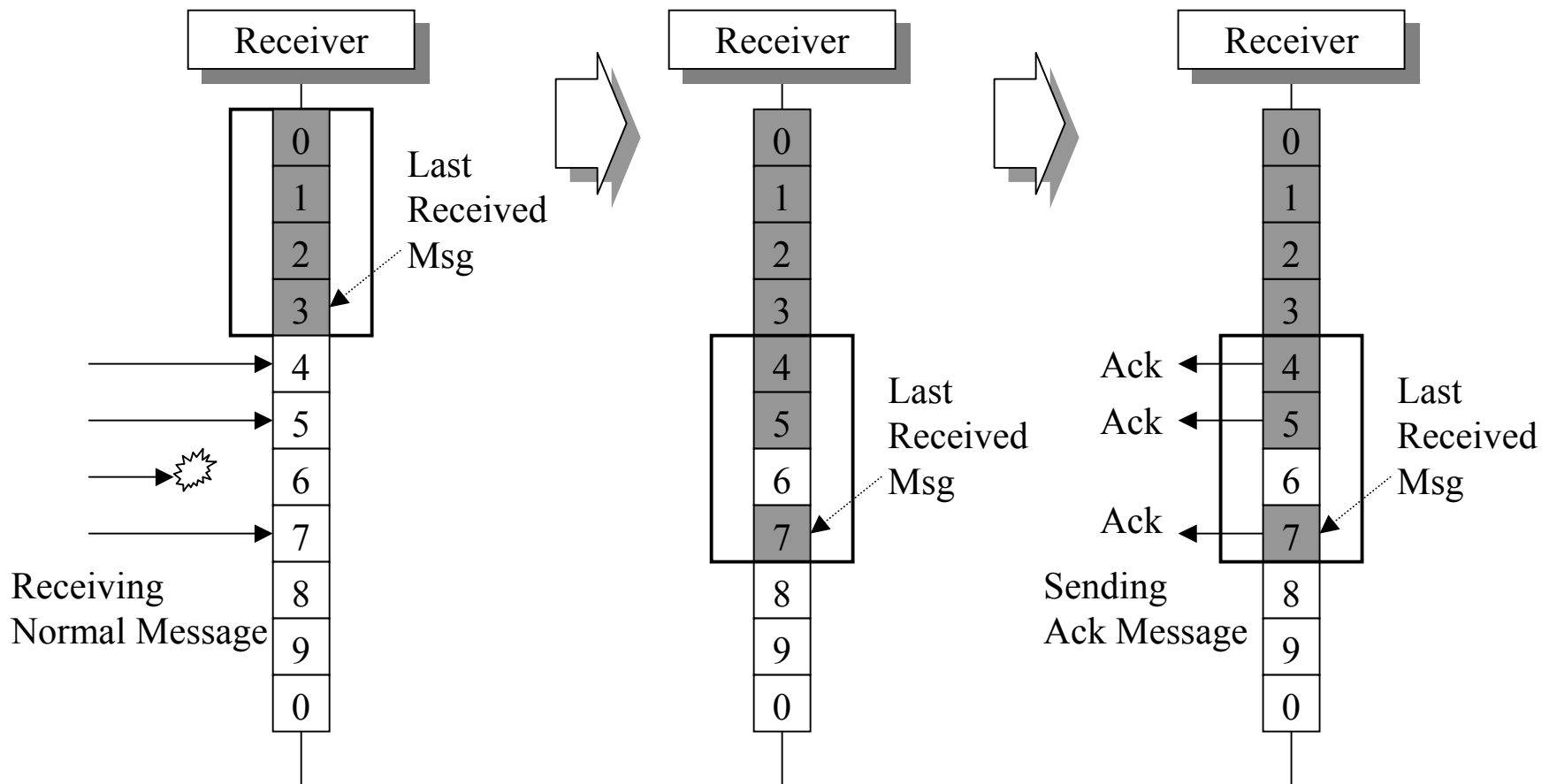
■ : Received and Ack'd, or Received but not Ack'd yet

□ : Not Received

Sliding of Duplication Check Window

- ❑ Duplication Check Window is slid by received message in width from 1 to W

Ex. $M = 10$, $W = 4$, Sliding width = 2



Advantages of Sliding Window and Sequence Number

❑ The Sliding Window provides

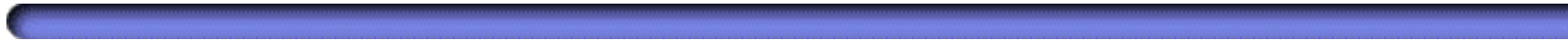
- Expectation of required maximum buffer size in the Sender
- High performance

❑ The Sequence Number in the Sliding Window provides

- Guarantee of Message Order
 - The Receiver can correct invalid order of received messages in the buffer using Sequence Numbers and then can pass the messages to higher level in correct order
- Effective Duplication Check
 - The Receiver can use Duplication Check Window which has a list of Sequence Numbers should be checked

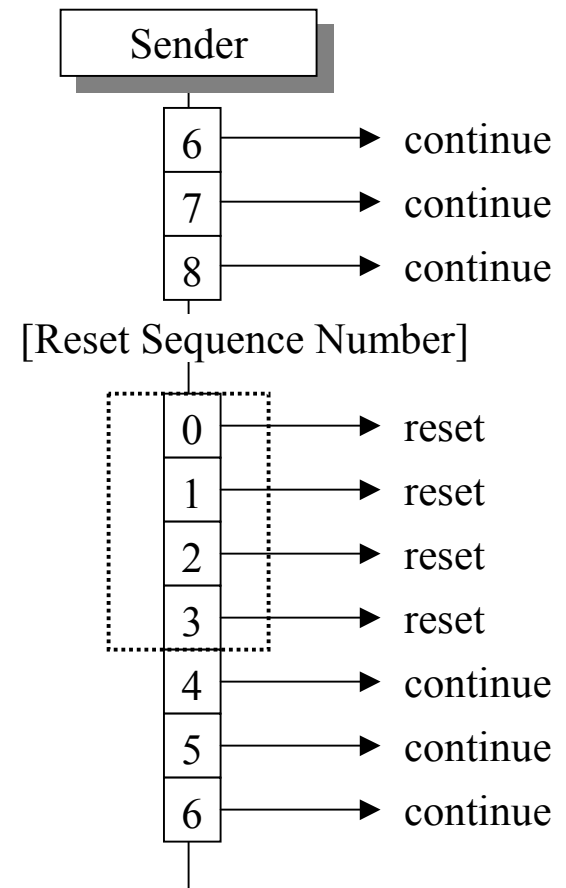


Other Issues



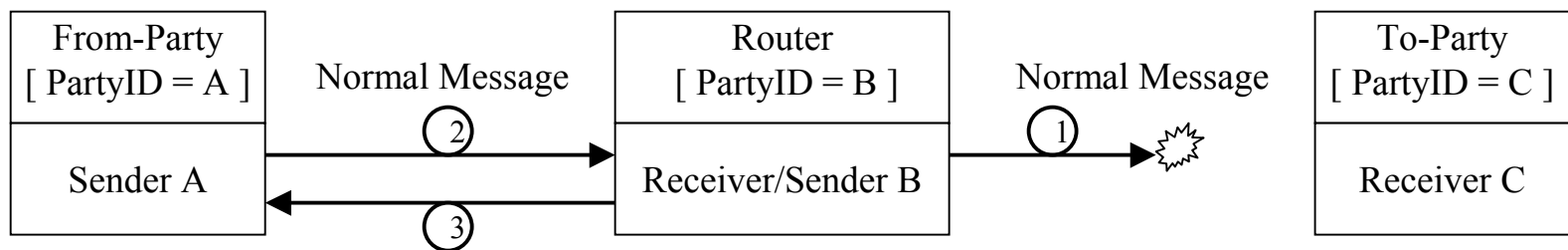
Notification of Sequence Number Reset

- ❑ Add “Status” attribute to SequenceNumber element which takes one of following values:
 - Status = “Reset”
 - It is first message series from the Sender to the Receiver
 - The Sender reset the SequenceNumber
 - Status = “Continue”
 - Any other case (including wraparound)
- ❑ When the Sender resets the SequenceNumber, all the Sequence Numbers’ Status attributes in the first Window take value “Reset” so that the Receiver can know the reset immediately even if the messages in the first window reach in invalid order



Error Handling in Routing

- ❑ When the Router can't forward received messages, the Router MAY return an Error Message (Transient Error) instead of an Acknowledgement Message so that the Sender can suspend sending messages
 - Add new Error Message (Transient Error) to MS spec to suspend sending message in the Sender
 - When the Sender receives the Error Message (Transient Error), the Sender suspend sending message for the period specified in the MinRetrySecs field in the Error Message
 - If the MinRetrySecs field does not exist, RetryInterval specified in TPA is used as the suspending time



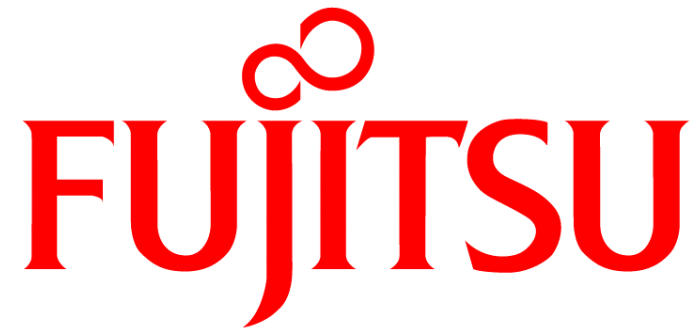
Error Message (Transient Error)

Transfer of Large Documents

- ❑ Two Messaging Service parameters are specified for transfer a large document
 - MaxSize - maximum payload size which the Sender can send
 - CompressEncoding - encoding to compress the Payload Document
- ❑ After the compression, if the size of message is still larger than MaxSize, the Sender can split the Document into some parts and can send each as individual Payload Documents in separate Normal Messages. Additional elements used for splitting are:

- SplitId : a common unique id for the group
- SplitNumber : total number and sequential number

```
<MessageData>  
  <MessageId>UUID-A</MessageId>  
  ...  
  <SplitId>UUID-B</SplitId>  
  <SplitNumber Total="5">3</SplitNumber>  
</MessageData>
```

T H E P O S S I B I L I T I E S A R E I N F I N I T E