

SpaceWire RMAP Protocol

SpaceWire Working Group Meeting

Steve Parkes
University of Dundee

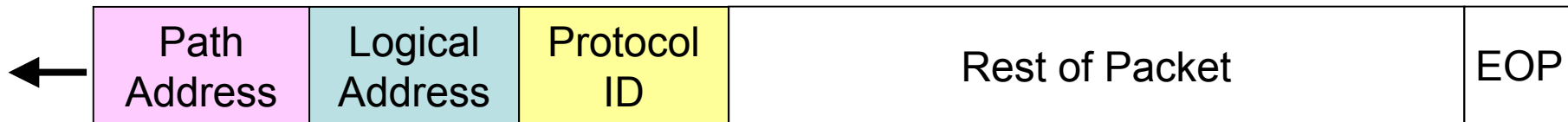
Aims of RMAP

- Remote Memory Access Protocol
- Provide a means of
 - Writing to
 - Reading from
 - Registers or memory on a SpaceWire node
 - Over a SpaceWire network
 - Registers are considered to be memory mapped
- Be simple and effective
- Flexible to encompass diverse applications

SpaceWire Protocol Identifier



Logical Address with Protocol ID



Path Address with Protocol ID

RMAP Commands

- Write
 - With or without acknowledgement
 - Verifying data before writing or writing without verification
- Read
- Read-Modify-Write

Write Commands

- Write non-acknowledged, non-verified
 - Writes zero or more bytes to memory in a destination node
 - Command header is checked using a CRC before the data is written
 - Data is not checked before it is written
 - No acknowledgement given to indicate that the command has been successfully executed
- Used for writing large amounts of data to a destination
 - Where it can be safely assumed that
 - The write operation completed successfully
 - Or that is not critical if it does not succeed
 - E.g. writing camera images to a temporary working buffer

Write Commands

- Write non-acknowledged, verified
 - Writes zero or more bytes to memory in a destination node
 - Command and data are both checked using CRCs before the data is written
 - Limits the amount of data that can be transferred in a single write operation
 - Owing to limited buffer space in destination
 - Erroneous data cannot be written to memory
 - No acknowledgement given to indicate that the command has been successfully executed
- Used for writing command registers and small amounts of data to a destination
 - where it can be safely assumed that
 - the write operation completed successfully
 - or where errors are detected in a different way
 - E.g. writing many commands to different configuration registers in a device and then checking for an error using a status register

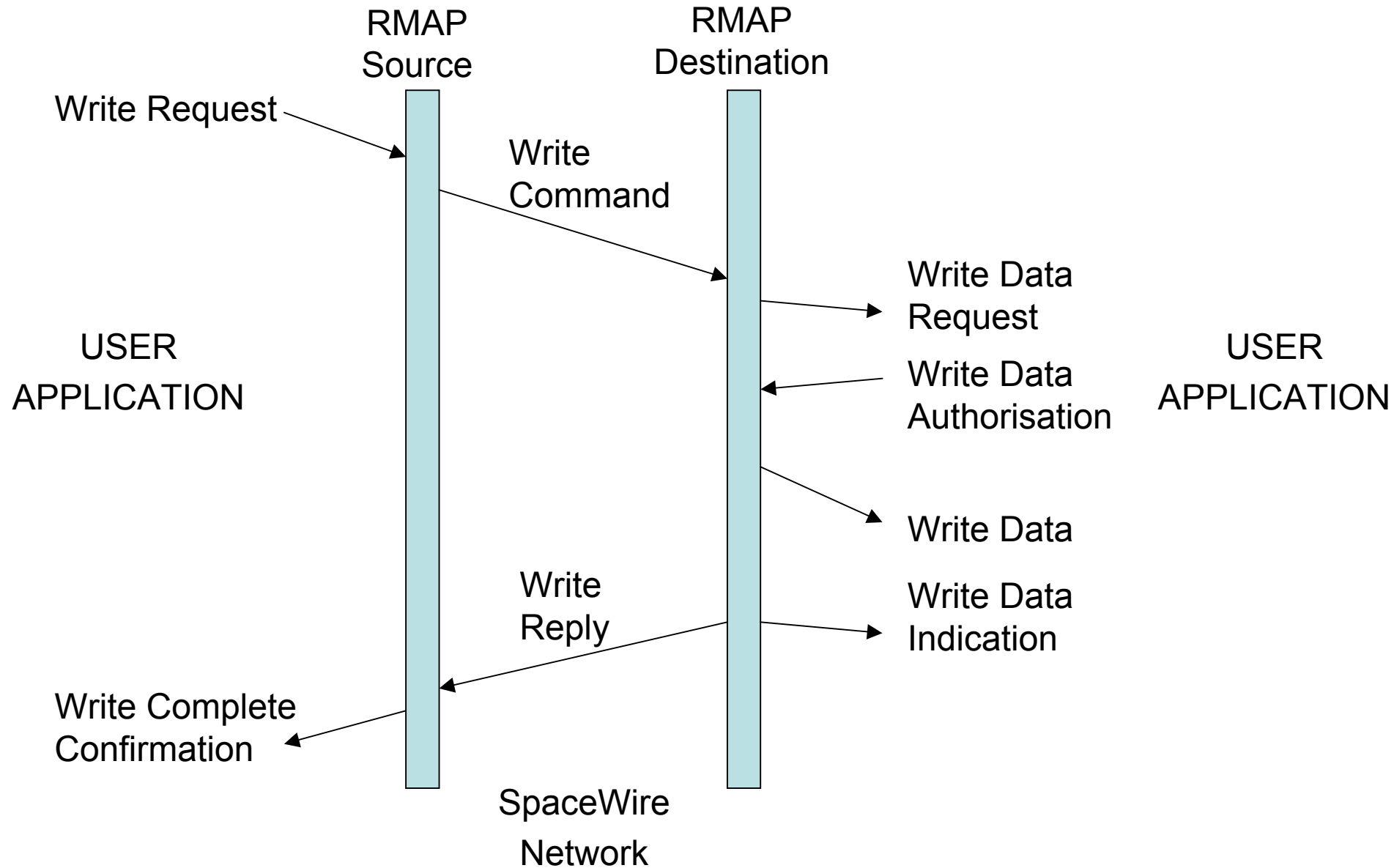
Write Commands

- Write acknowledged, non-verified
 - Writes zero or more bytes to memory in a destination node
 - Command is checked using a CRC before the data is written
 - Data is not checked before it is written
 - Acknowledgement sent to indicate that the command has been successfully executed
- used for writing large amounts of data to a destination
 - where it can be safely assumed that
 - the write operation completed successfully,
 - but an acknowledgement is required.
 - For example writing sensor data to memory.

Write Commands

- Write acknowledged, verified
 - Writes zero or more bytes to memory in a destination node
 - Command and data are both checked using CRCs before the data is written
 - Limits the amount of data that can be transferred in a single write operation
 - Owing to limited buffer space in destination
 - Erroneous data cannot be written to memory
 - Acknowledgement sent to indicate that the command has been successfully executed
- Used for writing small amounts of data to a destination
 - where it is important to have confirmation that
 - the write operation was executed successfully.
 - For example writing to command or configuration registers.

Write Operation



Write Command

Logical Addressing

First byte transmitted

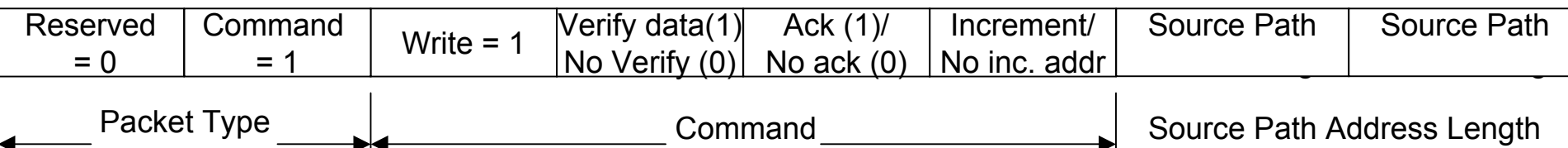
Destination Logical Address	Protocol Identifier	Packet Type, Command, Source Path Addr Len	Destination Key
Source Logical Address	Transaction Identifier	Transaction Identifier	Extended Write Address
Write Address (MS)	Write Address	Write Address	Write Address (LS)
Data Length (MS)	Data Length	Data Length (LS)	Header CRC
Data	Data	Data	Data
Data	Data	Data	Data
Data	Data CRC	EOP	

Last byte transmitted

Bits in Packet Type / Command / Source Path Address Length Byte

MSB

LSB



Write Reply

First byte transmitted

Source Logical Address	Protocol Identifier	Packet Type, Command, Source Path Addr Len	Status
Destination Logical Address	Transaction Identifier	Transaction Identifier	Reply CRC

EOP

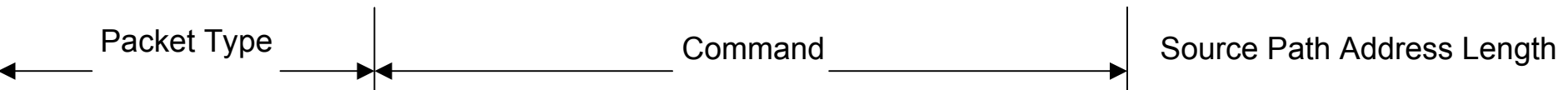
Last byte transmitted

Bits in Packet Type / Command / Source Path Address Length Byte

MSB

LSB

Reserved = 0	Response = 0	Write = 1	Verify data (1) No Verify (0)	Ack = 1	Increment/ No inc. addr	Source Path	Source Path
-----------------	-----------------	-----------	----------------------------------	---------	----------------------------	-------------	-------------



Write Command

First byte transmitted

Path Addressing

	Destination Path Address	Destination Path Address	Destination Path Address
Destination Logical Address	Protocol Identifier	Packet Type, Command, Source Path Addr Len	Destination Key
Source Path Address	Source Path Address	Source Path Address	Source Path Address
Source Logical Address	Transaction Identifier	Transaction Identifier	Extended Write Address
Write Address (MS)	Write Address	Write Address	Write Address (LS)
Data Length (MS)	Data Length	Data Length (LS)	Header CRC
Data	Data	Data	Data
Data	Data	Data	Data
Data	Data CRC	EOP	

Last byte transmitted

Bits in Packet Type / Command / Source Path Address Length Byte

MSB

LSB

Reserved = 0	Command = 1	Write = 1	Verify data(1) No Verify (0)	Ack (1)/ No ack (0)	Increment/ No inc. addr	Source Path	Source Path
--------------	-------------	-----------	---------------------------------	------------------------	----------------------------	-------------	-------------



Write Reply

First byte transmitted

	Source Path Address	Source Path Address	Source Path Address
Source Logical Address	Protocol Identifier	Packet Type, Command, Source Path Addr Len	Status
Destination Logical Address	Transaction Identifier	Transaction Identifier	Reply CRC

EOP

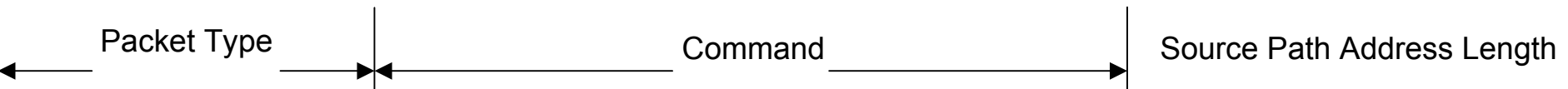
Last byte transmitted

Bits in Packet Type / Command / Source Path Address Length Byte

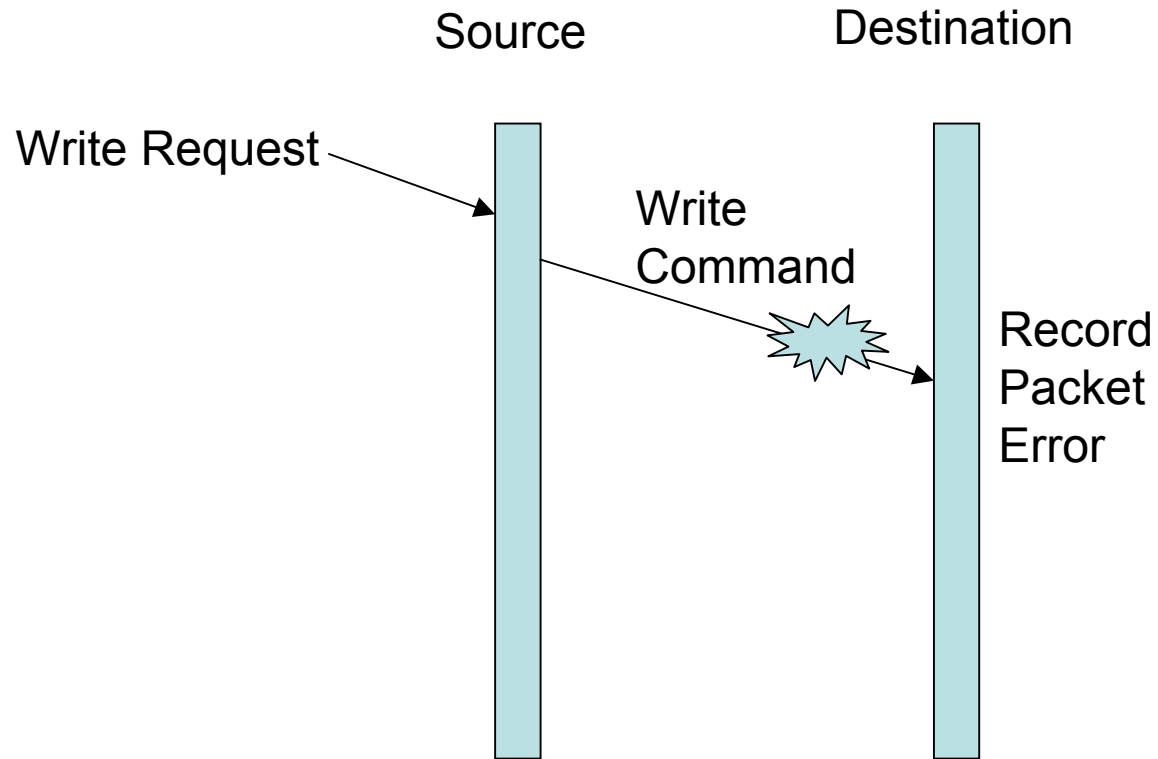
MSB

LSB

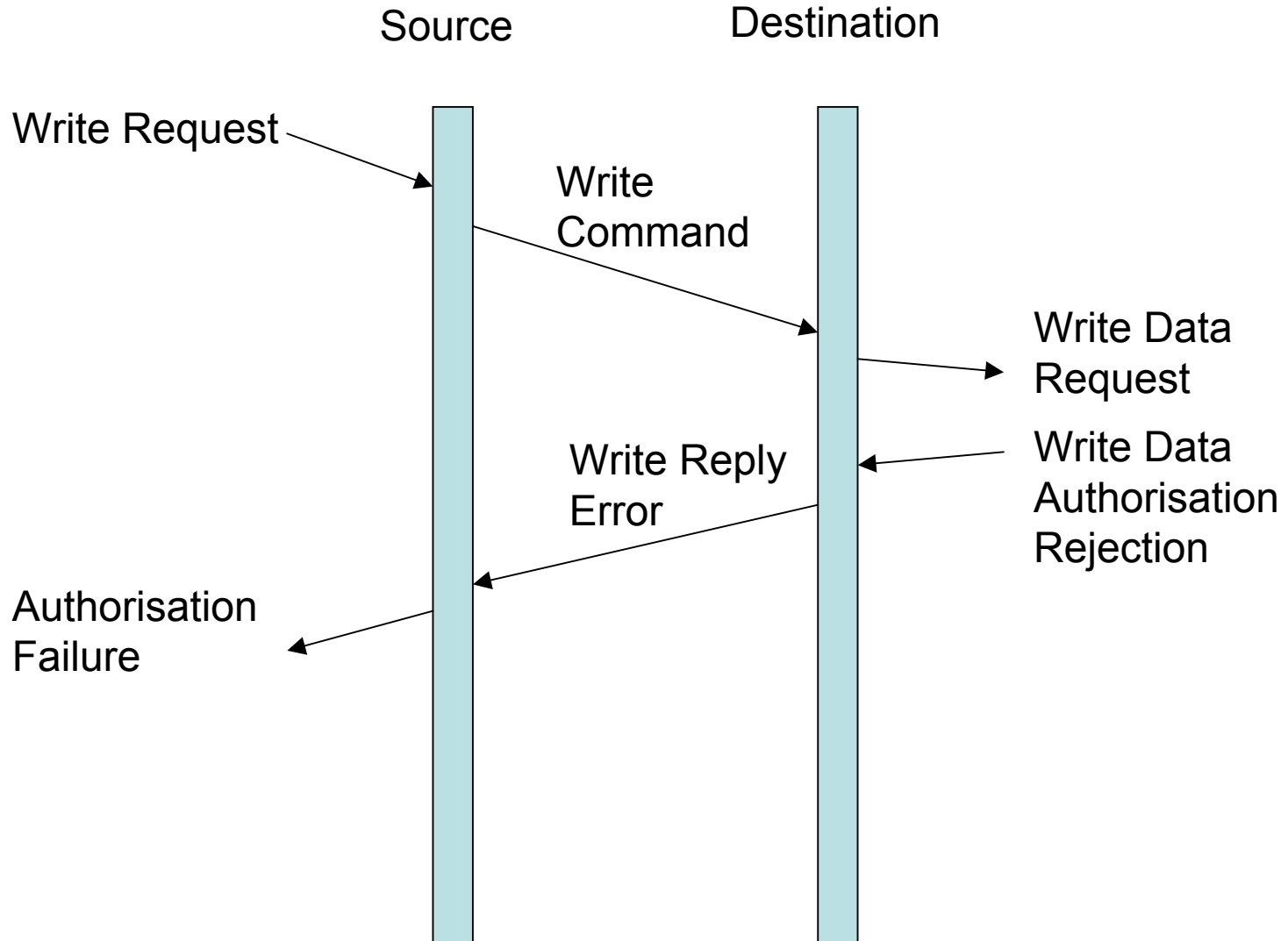
Reserved = 0	Response = 0	Write = 1	Verify data (1) No Verify (0)	Ack = 1	Increment/ No inc. addr	Source Path Address Length	Source Path Address Length
--------------	--------------	-----------	----------------------------------	---------	----------------------------	-------------------------------	-------------------------------



Write Command Header Error



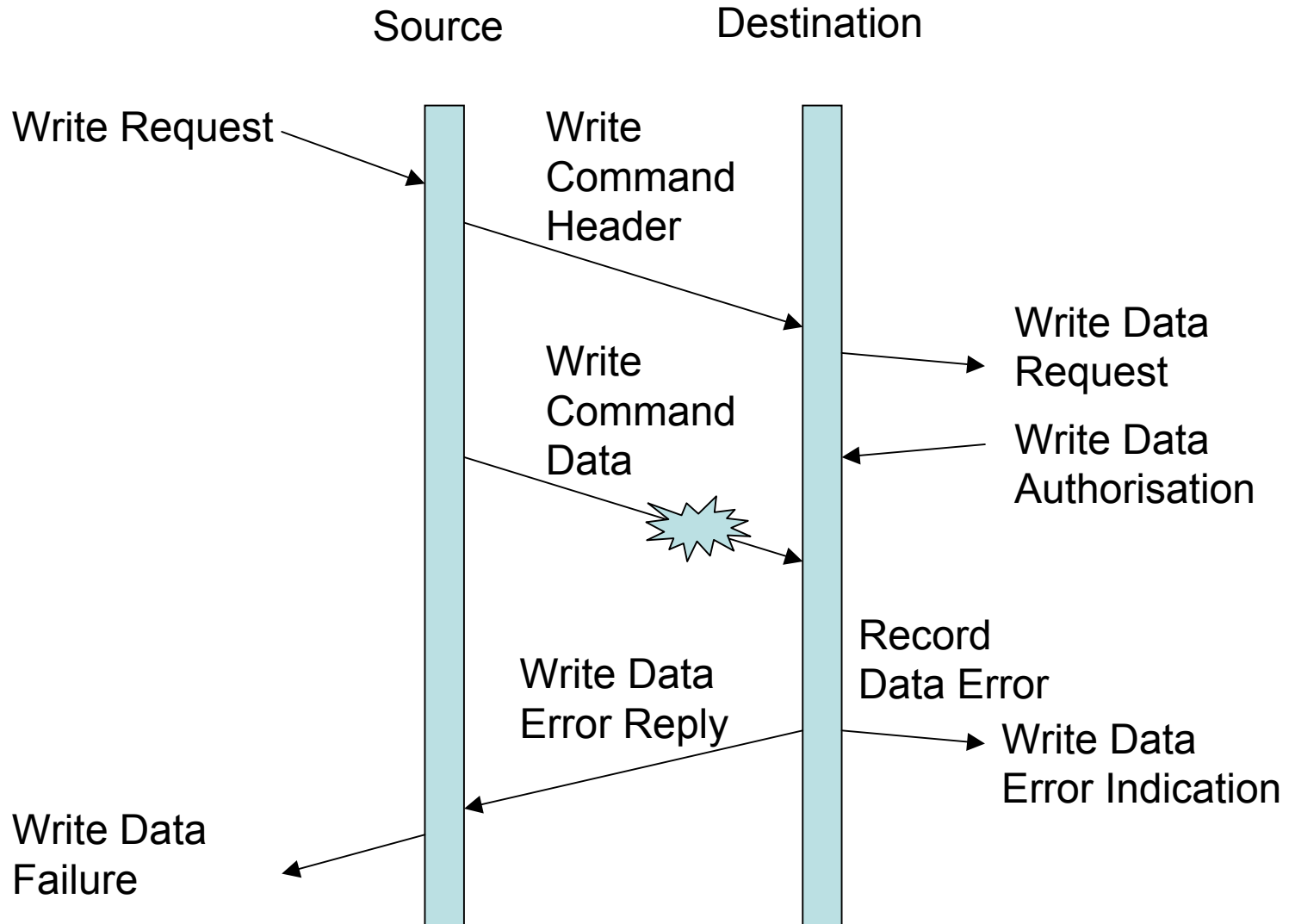
Write Authorisation Rejection



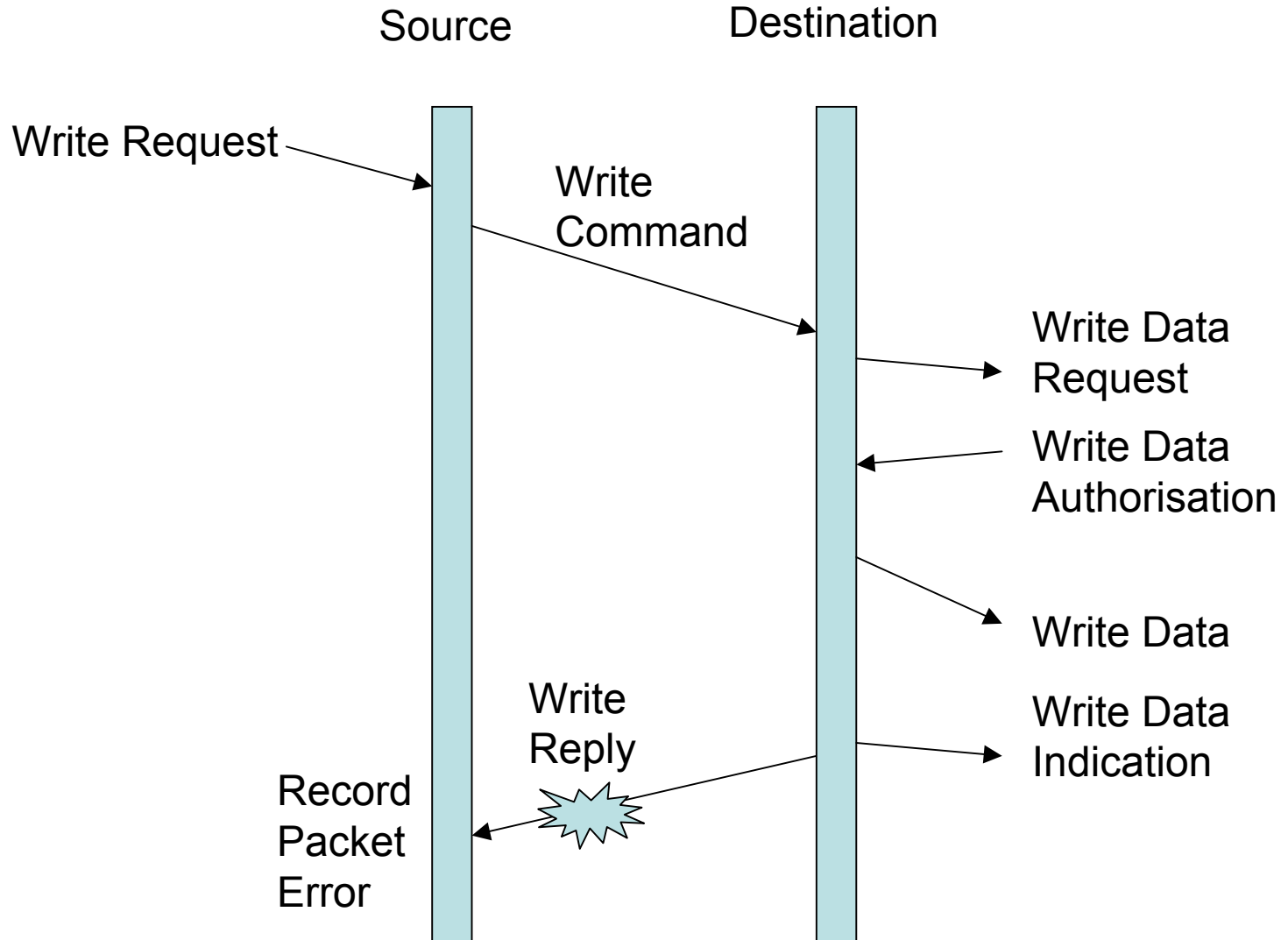
Command Authorisation

- Destination user application
- Can refuse to authorise command for any reason
- E.g.
 - Write address not 32-bit aligned
 - Length not a multiple of four bytes
 - Address range falls partially or completely outside an acceptable region

Write Data Error



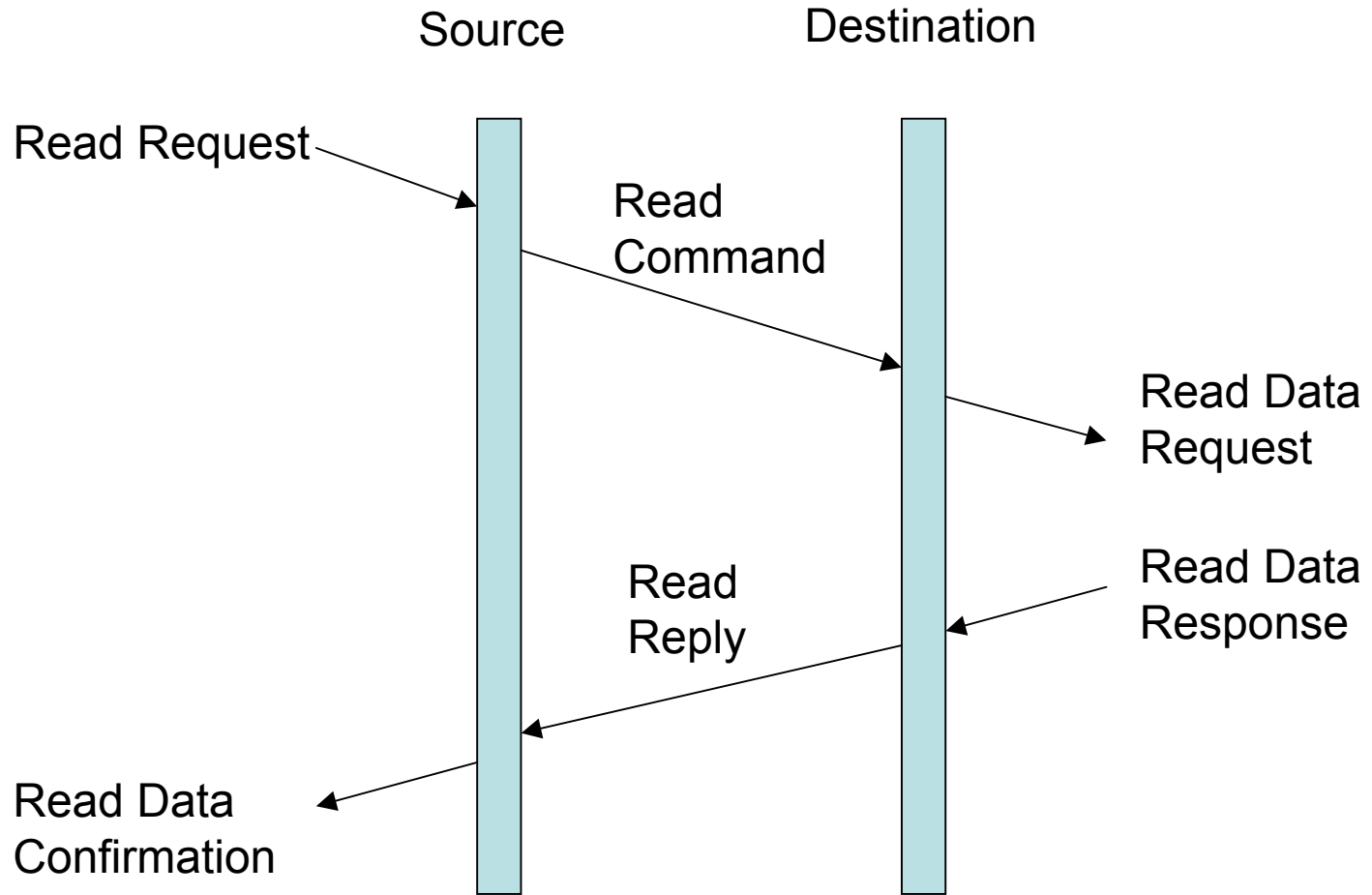
Write Reply Error



Read Command

- Read command
 - Reads one or more bytes of data
 - From specified area of memory in a destination node
 - Data read is returned in a reply packet.

Read Operation



Read Command

First byte transmitted

	Destination Path Address	Destination Path Address	Destination Path Address
Destination Logical Address	Protocol Identifier	Packet Type, Command Source Path Addr Len	Destination Key
Source Path Address	Source Path Address	Source Path Address	Source Path Address
Source Logical Address	Transaction Identifier (MS)	Transaction Identifier (LS)	Extended Read Address
Read Address (MS)	Read Address	Read Address	Read Address (LS)
Data Length (MS)	Data Length	Data Length (LS)	Header CRC

EOP

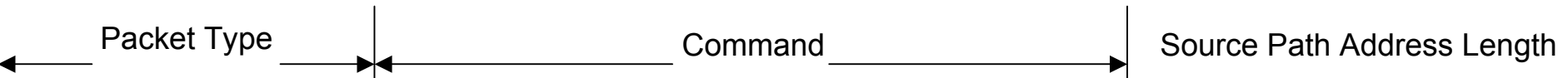
Last byte transmitted

Bits in Packet Type / Command / Source Path Address Length Byte

MSB

LSB

Reserved = 0	Command = 1	Read = 0	Read = 0	Read = 1 (Ack/No_Ack)	Increment/No inc. addr	Source Path Address Length	Source Path Address Length
--------------	-------------	----------	----------	-----------------------	------------------------	----------------------------	----------------------------



Read Reply

First byte transmitted

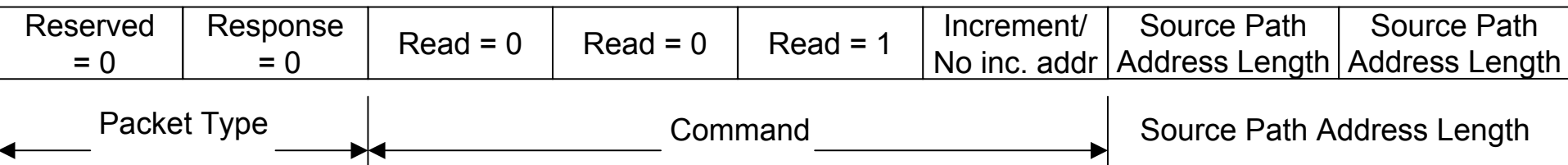
	Source Path Address	Source Path Address	Source Path Address
Source Logical Address	Protocol Identifier	Packet Type, Command, Source Path Addr Len	Status
Destination Logical Address	Transaction Identifier (MS)	Transaction Identifier (LS)	Reserved = 0
Data Length (MS)	Data Length	Data Length (LS)	Header CRC
Data	Data	Data	Data
Data	Data	Data	Data
Data	Data CRC	EOP	

Last byte transmitted

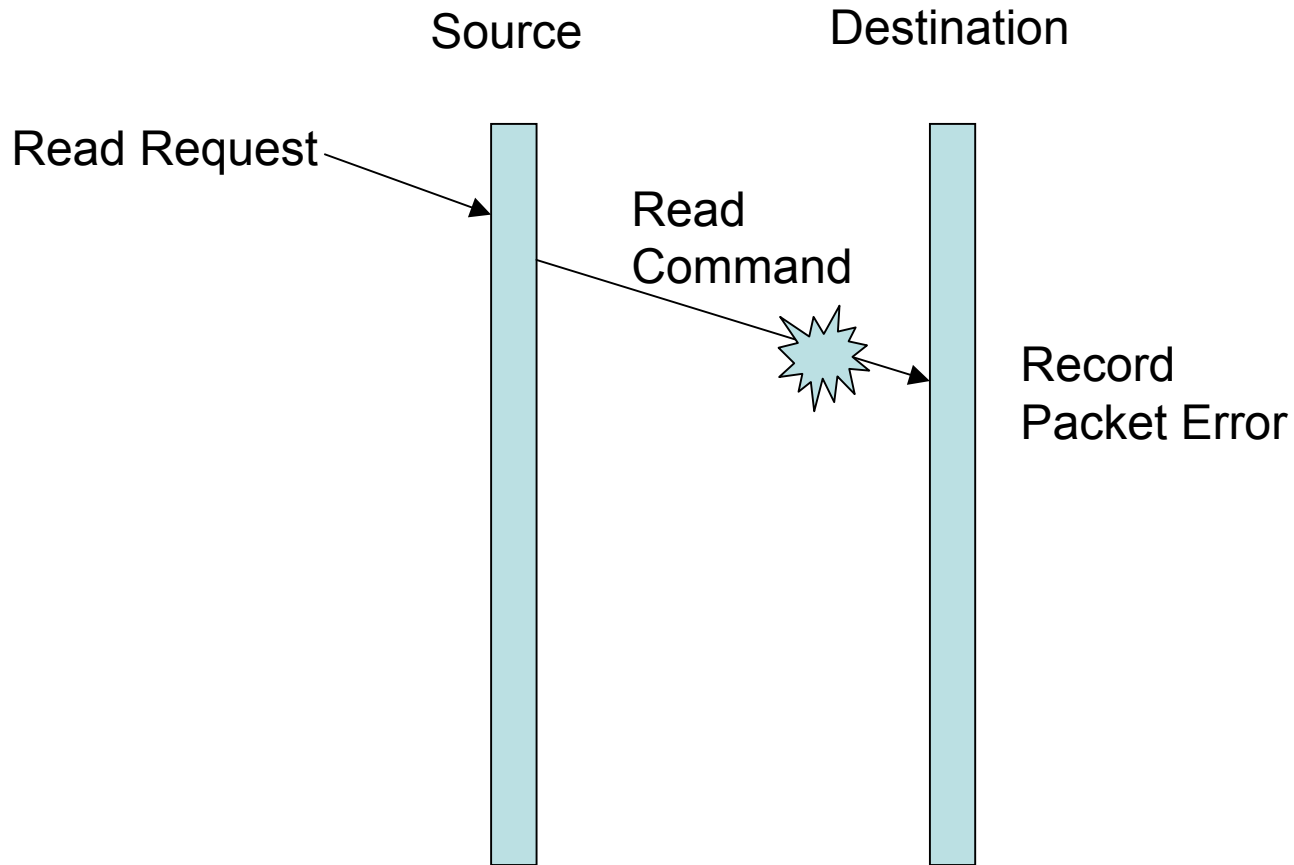
Bits in Packet Type / Command / Source Address Path Length Byte

MSB

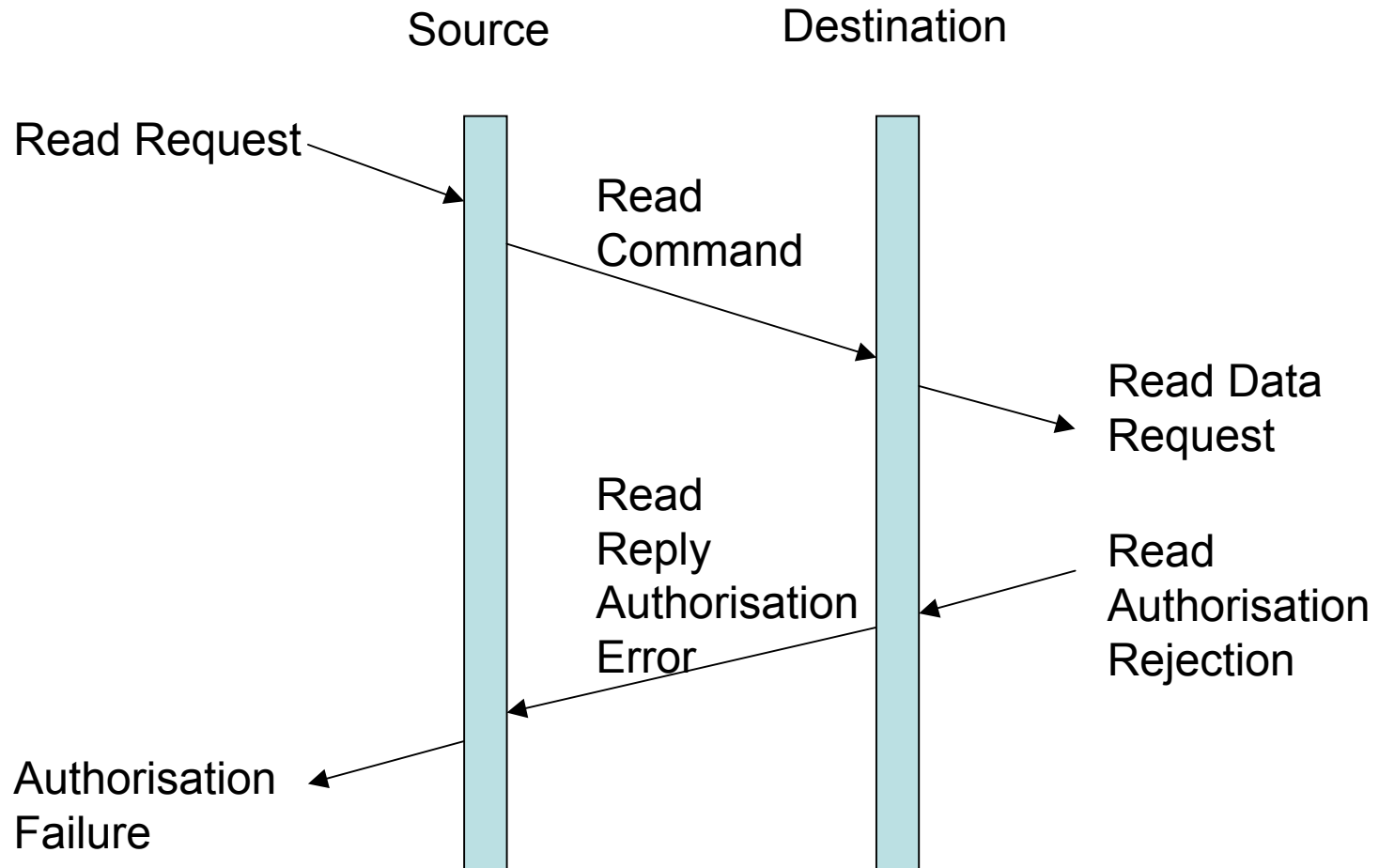
LSB



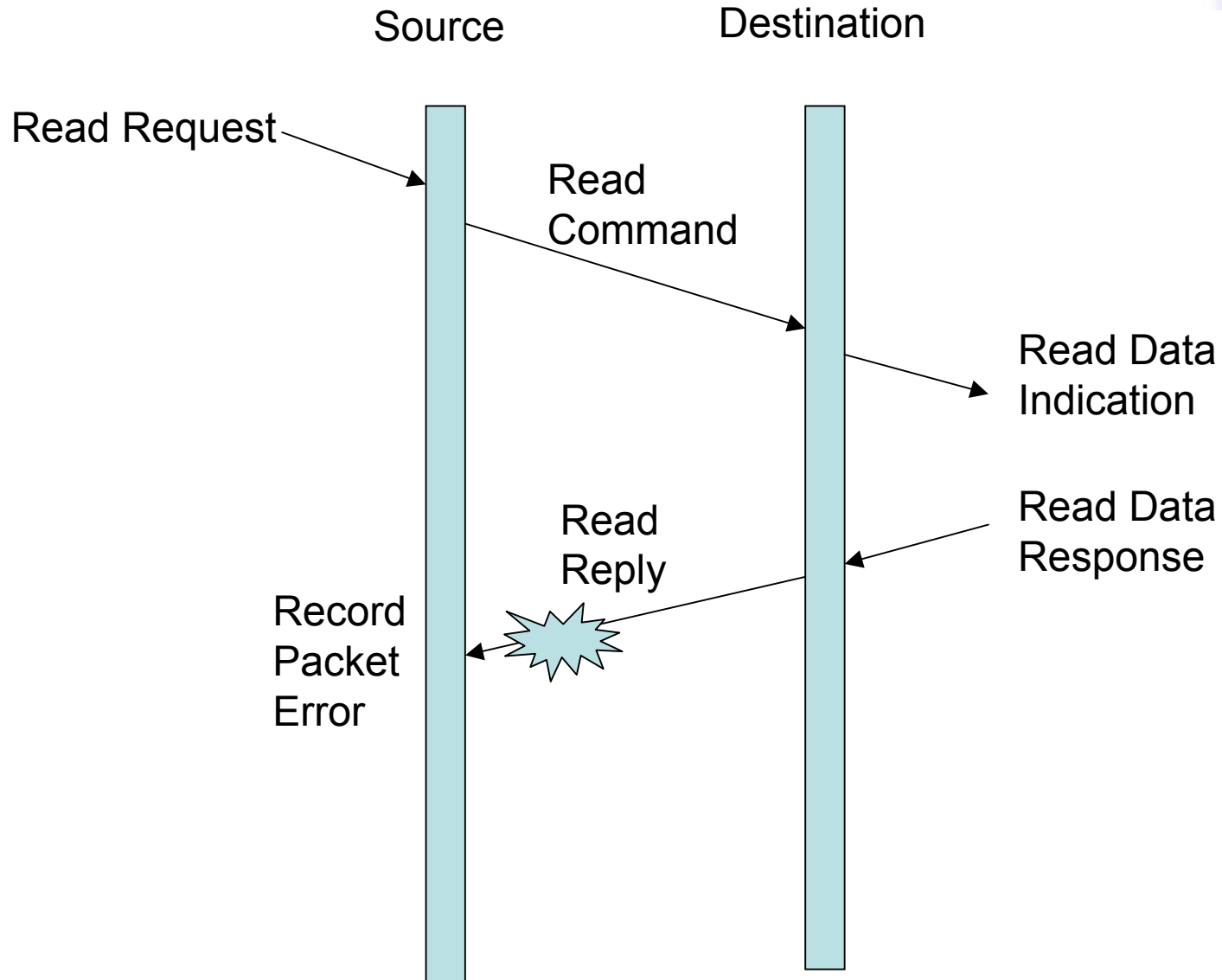
Read Command Error



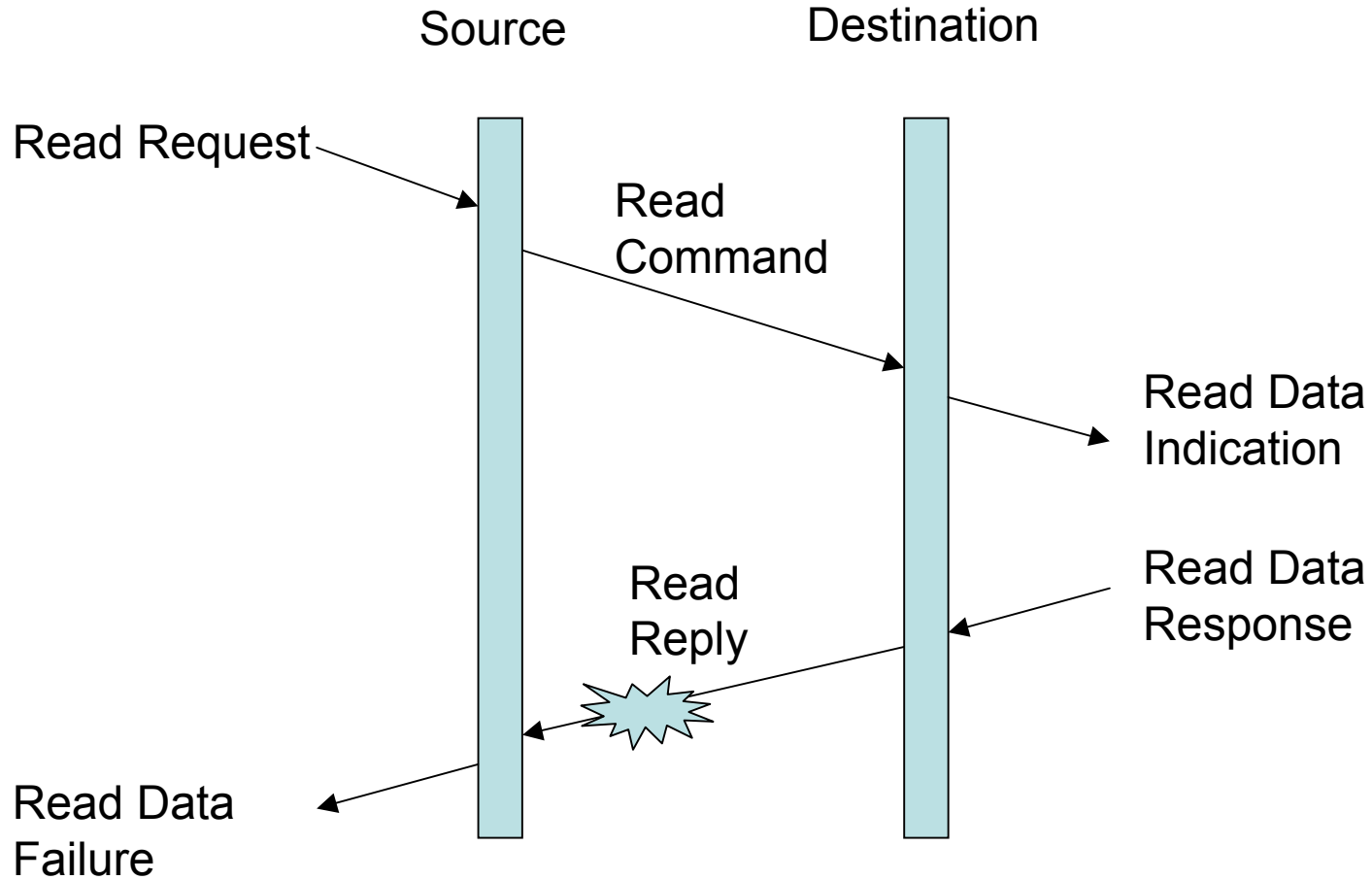
Read Authorisation Rejection



Read Reply Header Error



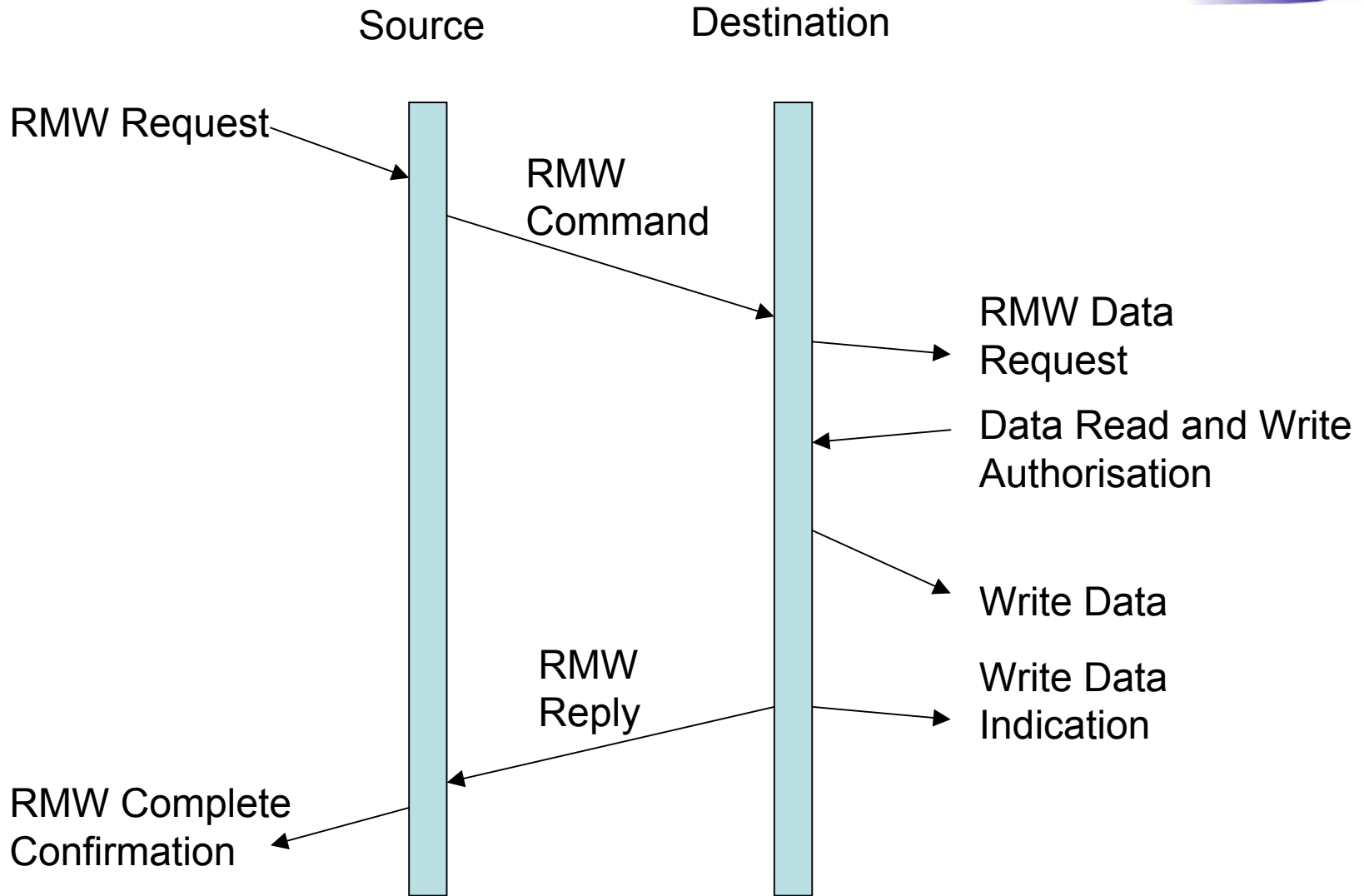
Read Reply Data Error



Read-Modify-Write Command

- Read-modify-write command
 - Reads a register (or memory)
 - Returns its value
 - Writes a new value, specified in the command, to the register.
 - Mask can be included, in the command
 - So that only certain bits of the register are written
- Provides an atomic operation that can be used for semaphores and other handshaking operations.

Read-Modify-Write Operation



Example Read-Modify-Write Operation

1	0	0	0	1	0	0	0
---	---	---	---	---	---	---	---

Data in command (Data)

1	0	0	0	1	1	1	0
---	---	---	---	---	---	---	---

Mask in command (Mask)

1	1	1	0	0	0	1	1
---	---	---	---	---	---	---	---

Data read from destination memory and returned to source (Read)

1	1	1	0	1	0	0	1
---	---	---	---	---	---	---	---

Data written to destination memory
= (Mask AND Data) OR (/Mask.Read)

Read-Modify-Write Command

First byte transmitted

	Destination Path Address	Destination Path Address	Destination Path Address
Destination Logical Address	Protocol Identifier	Packet Type, Command Source Path Addr Len	Destination Key
Source Path Address	Source Path Address	Source Path Address	Source Path Address
Source Logical Address	Transaction Identifier	Transaction Identifier	Extended RMW Address
RMW Address (MS)	RMW Address	RMW Address	RMW Address (LS)
Data +Mask Length (MS) = 00h	Data + Mask Length = 00h	Data + Mask Length (LS) = 00h, 02h, 04h, 06h or 08h	Header CRC
Data (MS)	Data	Data	Data (LS)
Mask (MS)	Mask	Mask	Mask (LS)
Data/Mask CRC	EOP		

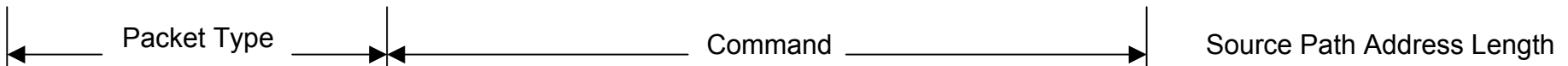
Last byte transmitted

Bits in Packet Type / Command / Source Address Path Length Byte

MSB

LSB

Reserved = 0	Command = 1	Read = 0	Verify Data Before WR = 1	Ack/No_Ack = 1	Incr. address = 1	Source Path Address Length	Source Path Address Length
--------------	-------------	----------	------------------------------	-------------------	----------------------	-------------------------------	-------------------------------



Read-Modify-Write Reply

First byte transmitted

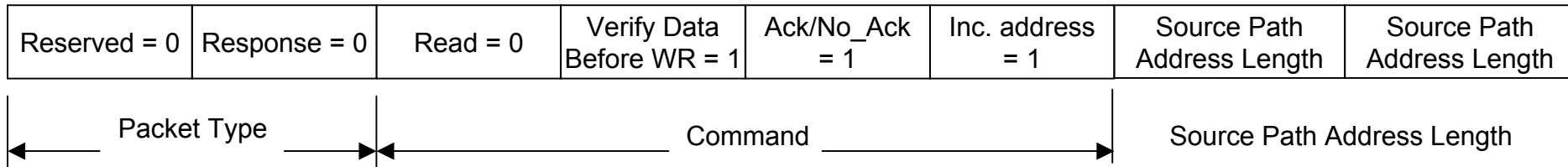
	Source Path Address	Source Path Address	Source Path Address
Source Logical Address	Protocol Identifier	Packet Type, Command, Source Path Addr Len	Status
Destination Logical Address	Transaction Identifier (MS)	Transaction Identifier (LS)	Reserved = 0
Data Length (MS) = 0	Data Length = 0	Data Length (LS) = 01h, 02h, 03h or 04h	Header CRC
Data	Data	Data	Data
Data CRC	EOP		

Last byte transmitted

Bits in Packet Type / Command / Source Path Address Length Byte

MSB

LSB



Data length zero

- If data length is zero no data will be read or written
- Data length of zero can be used for testing whether a command is acceptable by a unit.

Partial Implementation of RMAP

- Partial implementations are permitted
- For example:
 - Support of write and read but not RMW commands
 - Support of 32-bit data lengths only
- If destination receives command it does not support
- Or command with options not supported
- It refuses to authorise the command
- Command is not executed
- If reply requested then it will contain Authorisation Failure error code

Posted Operations

- All read and write operations defined in the RMAP protocol are posted operations
- I.e. source does not wait for acknowledgement or reply to be received
- Many reads and writes can be outstanding at any time
- Means that no timeout mechanism implemented in RMAP for missing acknowledgements or replies
- If acknowledgement or reply timeout mechanism is required it must be implemented in the source user application.

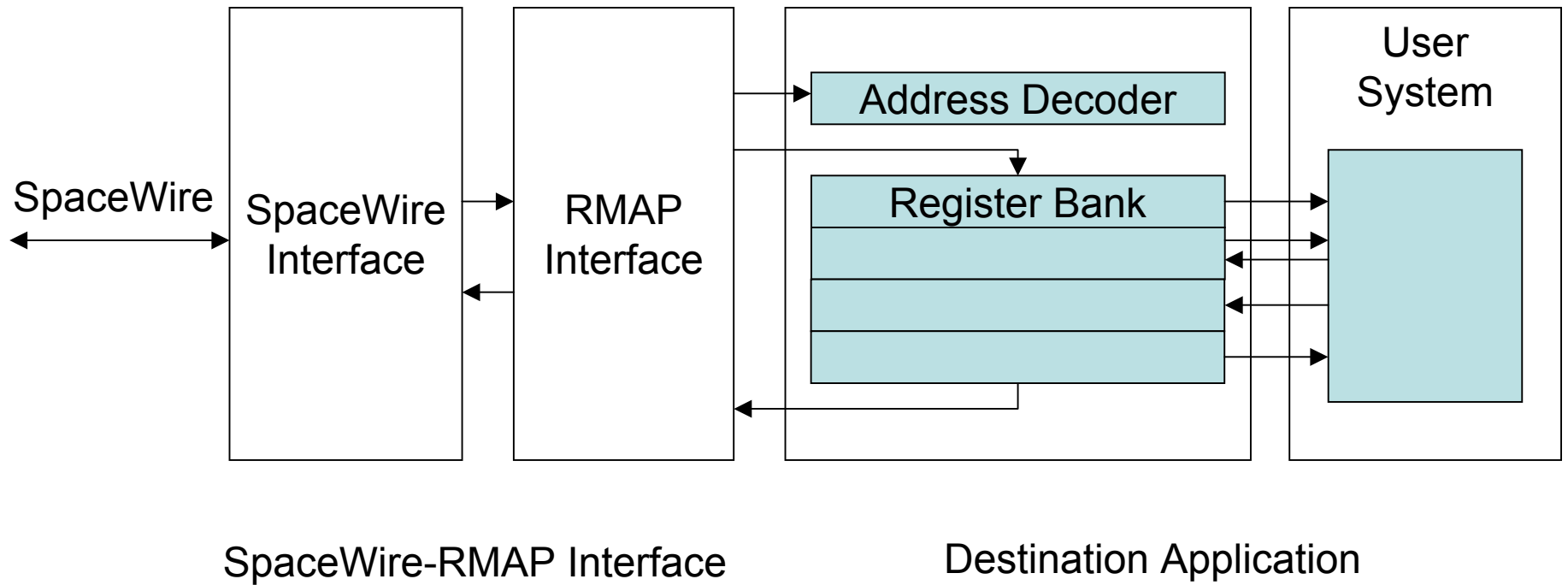
Receive Buffers

- Note RMAP does not handle the user application receive buffers
- otherwise it would have to maintain at least a pointer for every outstanding read request
- It is up to user application to handle any receive buffers.
- Appropriate receive buffer for a read reply may be identified in user application by transaction identifier in read reply.

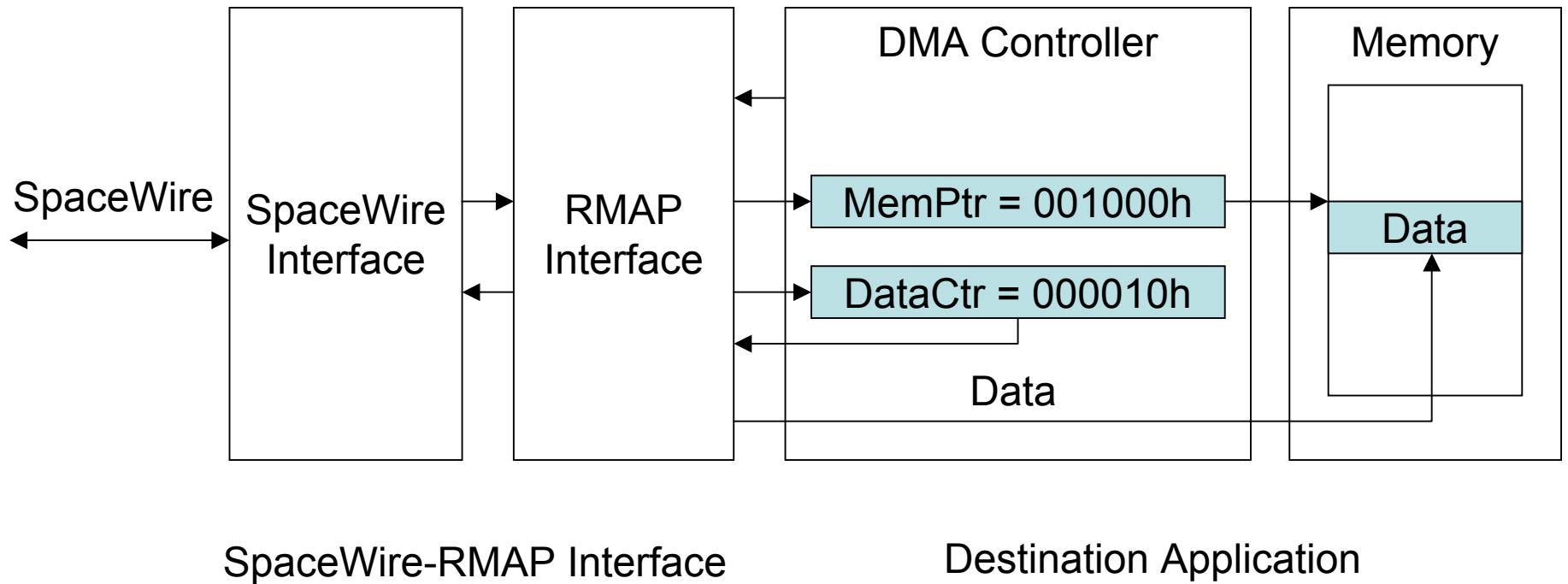
CRC Checks

- *Header CRC*
 - *8-bit CRC*
 - *Fairly simple to compute*
 - *Provides reasonable protection for short header*
- *Data CRC*
 - *Same 8-bit CRC*
 - *May be computed using same hardware/software as Header CRC*
 - *Provides reasonable protection for short data lengths*
 - *For long packets of data additional protection may be necessary*
 - *Which must be supplied by the user application*
- *Galois version of CRC used*
 - $X^8 + X^2 + X^1 + 1$
 - *Simple to implement in hardware*

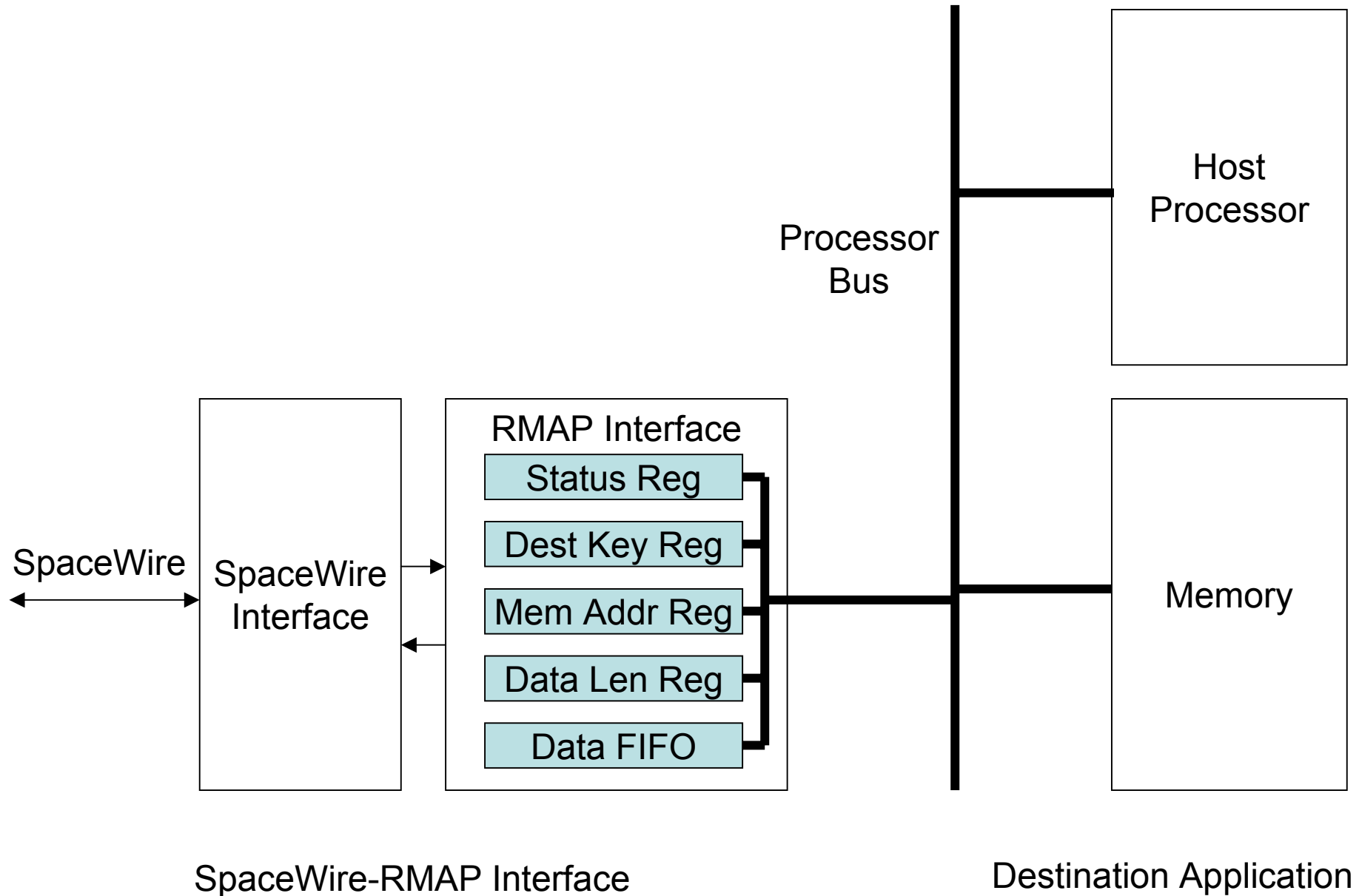
Access Registers with RMAP



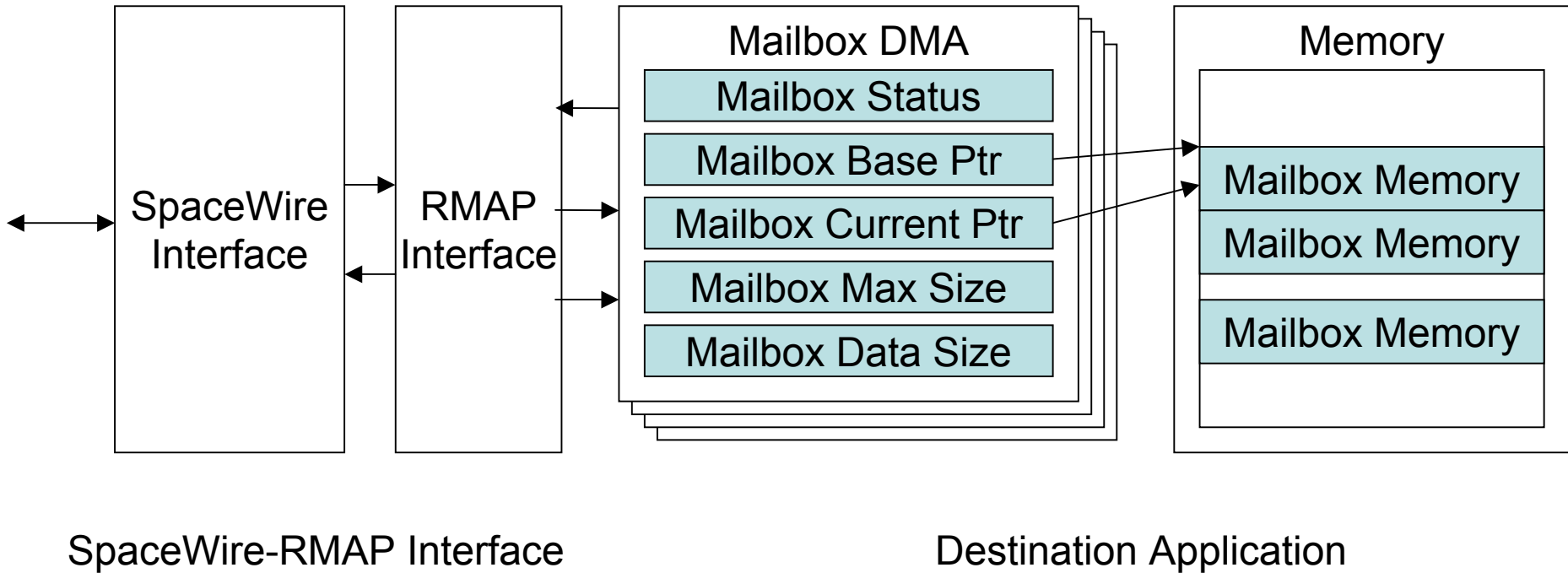
Write to Memory with DMA



Read from Memory with Software



Write to Mailbox with DMA



Changes from Draft C to Draft D

- Galois version of CRC specified
- Additional error code added
 - Error code 12
 - Invalid destination logical address
 - Informs the source that the destination logical address was not the value expected by the destination.

Implementation in Router

- We do not allow zeros in the source path address
- Except leading zeros
- E.g. for a single word source path address
0 0 0 1 and 0 2 3 4 are accepted source path address
0 0 0 0 , 2 3 0 1 and 2 3 4 0 are not accepted
- This is because 0 routes to a configuration port
- A configuration port should not be configuring other configuration ports.
- Hence reply/acknowledgement should not be routed to port 0

Summary

- RMAP provides:
 - A consistent means of reading and writing
 - Registers and memory
 - Over SpaceWire network
 - Is very flexible
 - Partial implementation permitted
 - Is being implemented in the ESA radiation tolerant Router ASIC
- Draft C of standard currently being reviewed by working group
- Will be included in ECSS-E50-12 Part 2