SpaceWire-RT Initial Protocol

Steve Parkes
Space Technology Centre
University of Dundee

Space Technology Centre
University of Dundee

Aims and objectives

- Present an initial set of protocols
- Meet SpaceWire-RT requirements

- These protocols will be updated following prototyping

Space
Technology
Centre
University of Dundee

WARNING

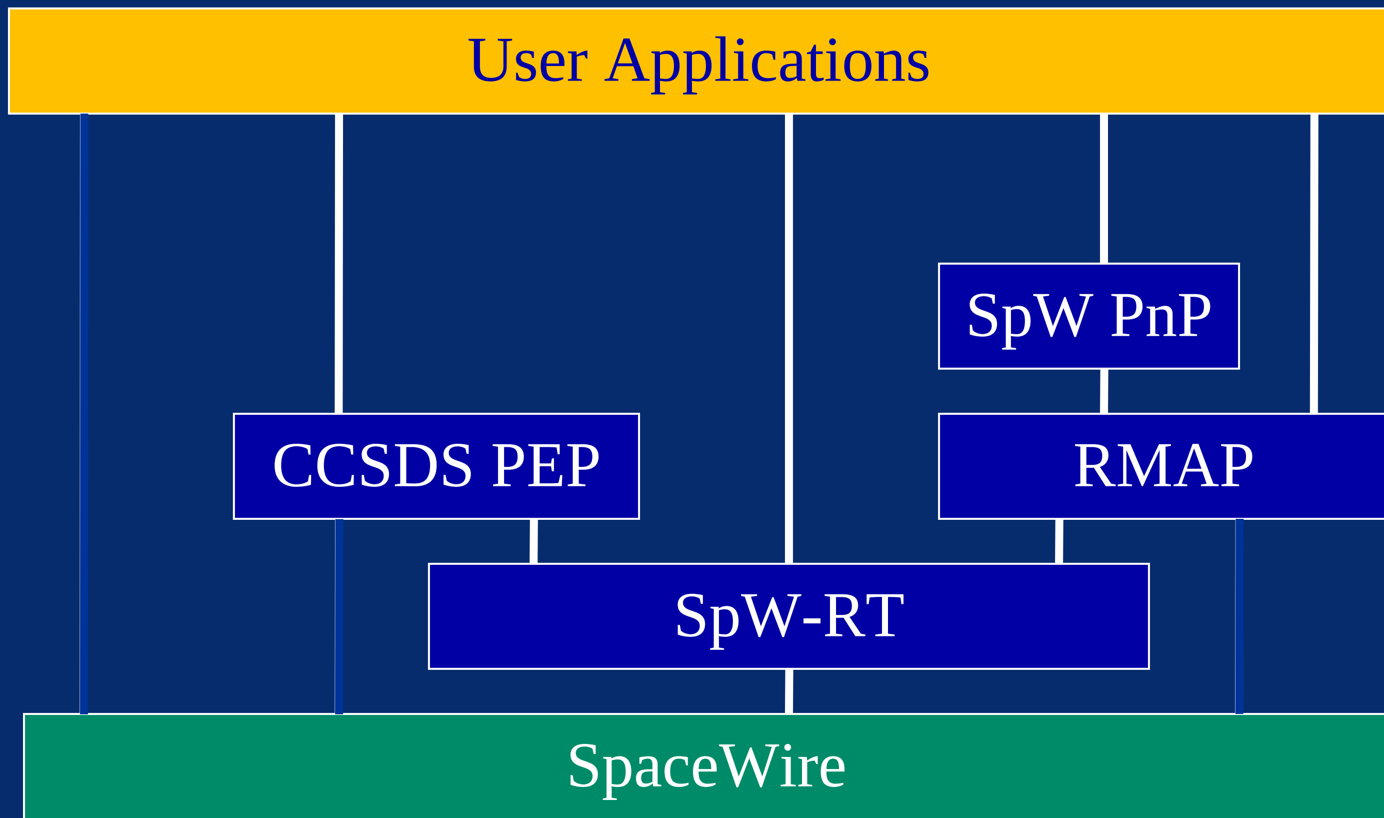
This current document is an early draft of the proposed standard and is for discussion purposes only. It will change after prototyping work has been completed. Applicable documents may also change.

DO NOT USE THIS DOCUMENT TO DESIGN DEVICES OR SYSTEMS!

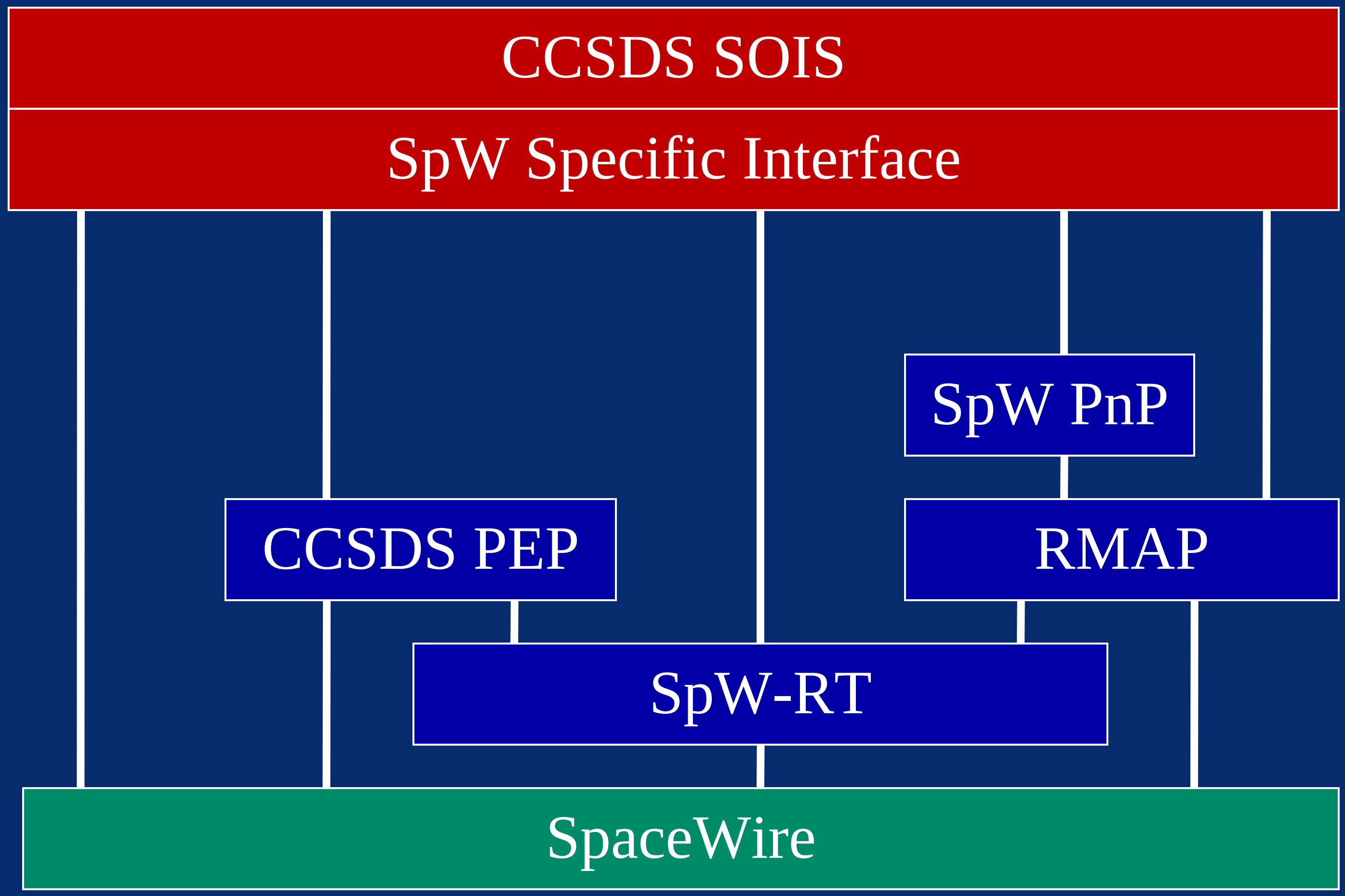
Contents

- Protocol stack
- Overview of SpaceWire-RT protocols
- Summary of progress

SpaceWire Protocol Stack



CCSDS SOIS use of SpW Protocol Stack



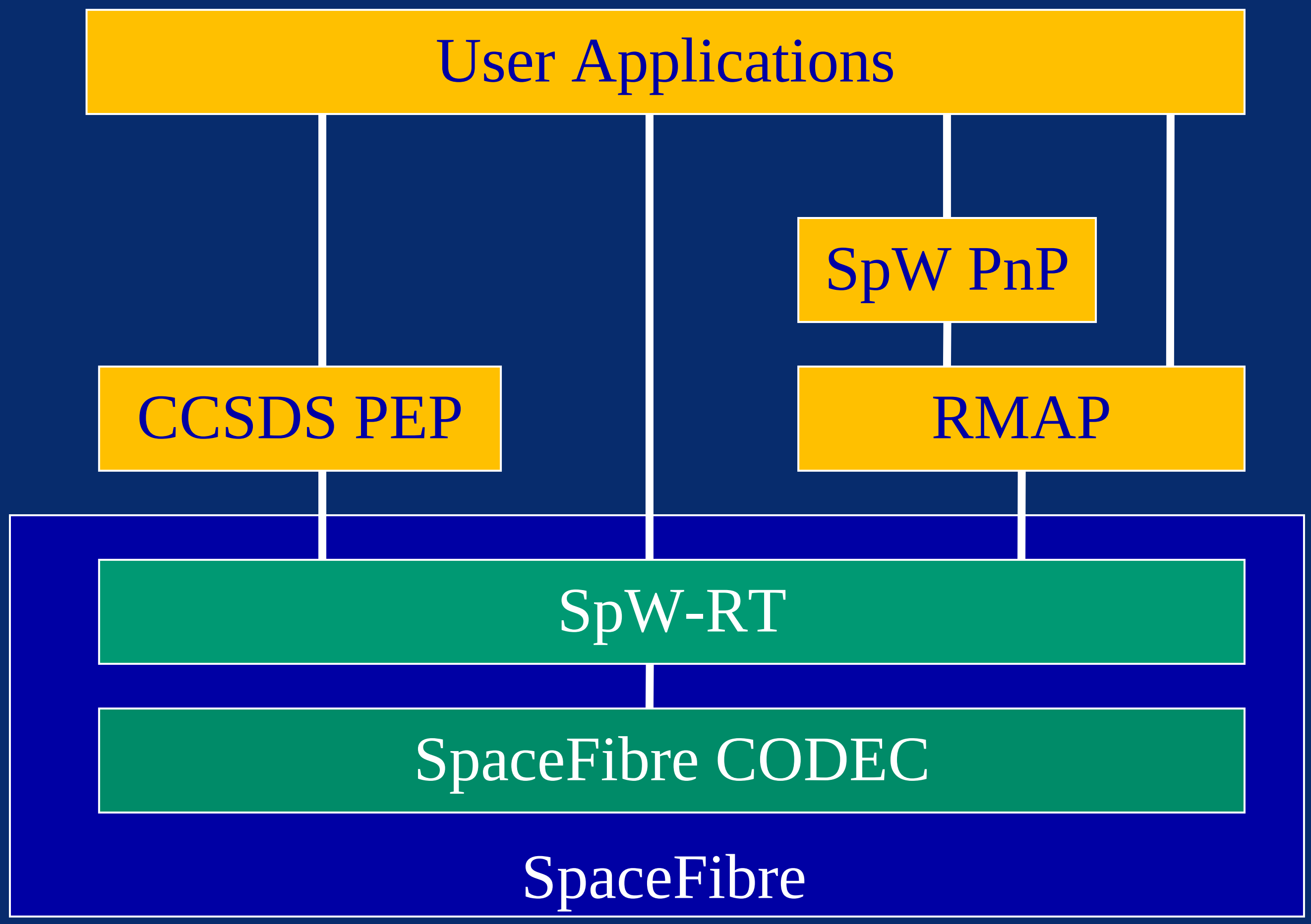
CCSDS SOIS is one possible application for SpaceWire protocol family

Space Technology Centre
University of Dundee

SpaceWire and SOIS

- SpaceWire protocols designed for
  - Efficient implementation
  - High performance
  - Hardware and software implementations
- SOIS designed for
  - Generic implementation
  - Over various buses
  - Hopefully a standard software interface
- Contention
  - Efficiency and hardware implementation
  - Vs
  - Generality
  - Hence mapping of SpaceWire protocols via SOIS software

SpaceFibre Protocol Stack



Space Technology Centre
University of Dundee

SpW-RT Communications Model

- Virtual point-to-point connections
  - Across SpaceWire network
  - Connecting source channel buffer in one node
  - To destination channel buffer in another node
- Two types of system:
- Asynchronous
  - Sending information is asynchronous
  - Priority used to provide timeliness
- Synchronous
  - Network bandwidth split up using time-slots
  - Each source channel assigned one or time-slots
  - When it is allowed to send data
  - Provides deterministic delivery

Communications Model

- One or more source channel buffers in a node
- One or more destination channel buffers
- User application writes data into a source channel buffer
- SpaceWire-RT transfers data to associated destination buffer in destination node
- User application reads data from destination buffer

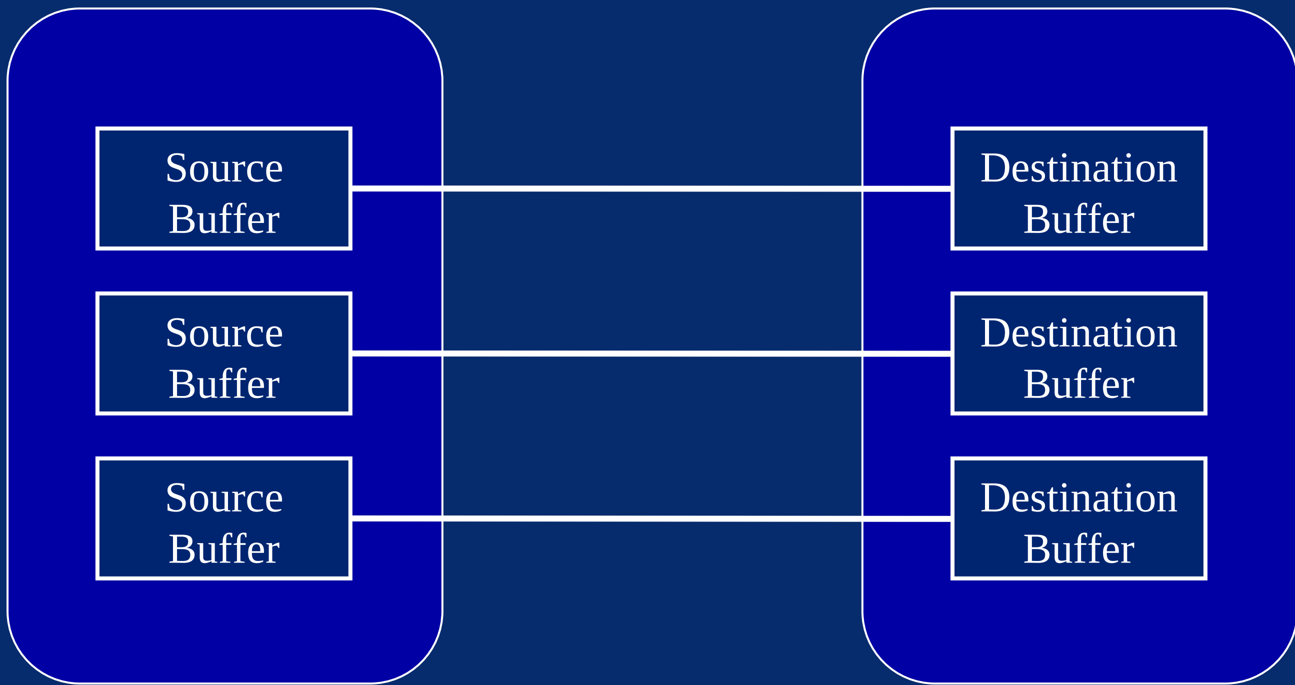
Space Technology Centre
University of Dundee

Channels

- Channel is:
  - Set of network resources
  - Connects source user application
  - To destination user application
- Includes:
  - Source channel buffer
  - SpaceWire links over which PDUs travel
  - Destination channel buffer

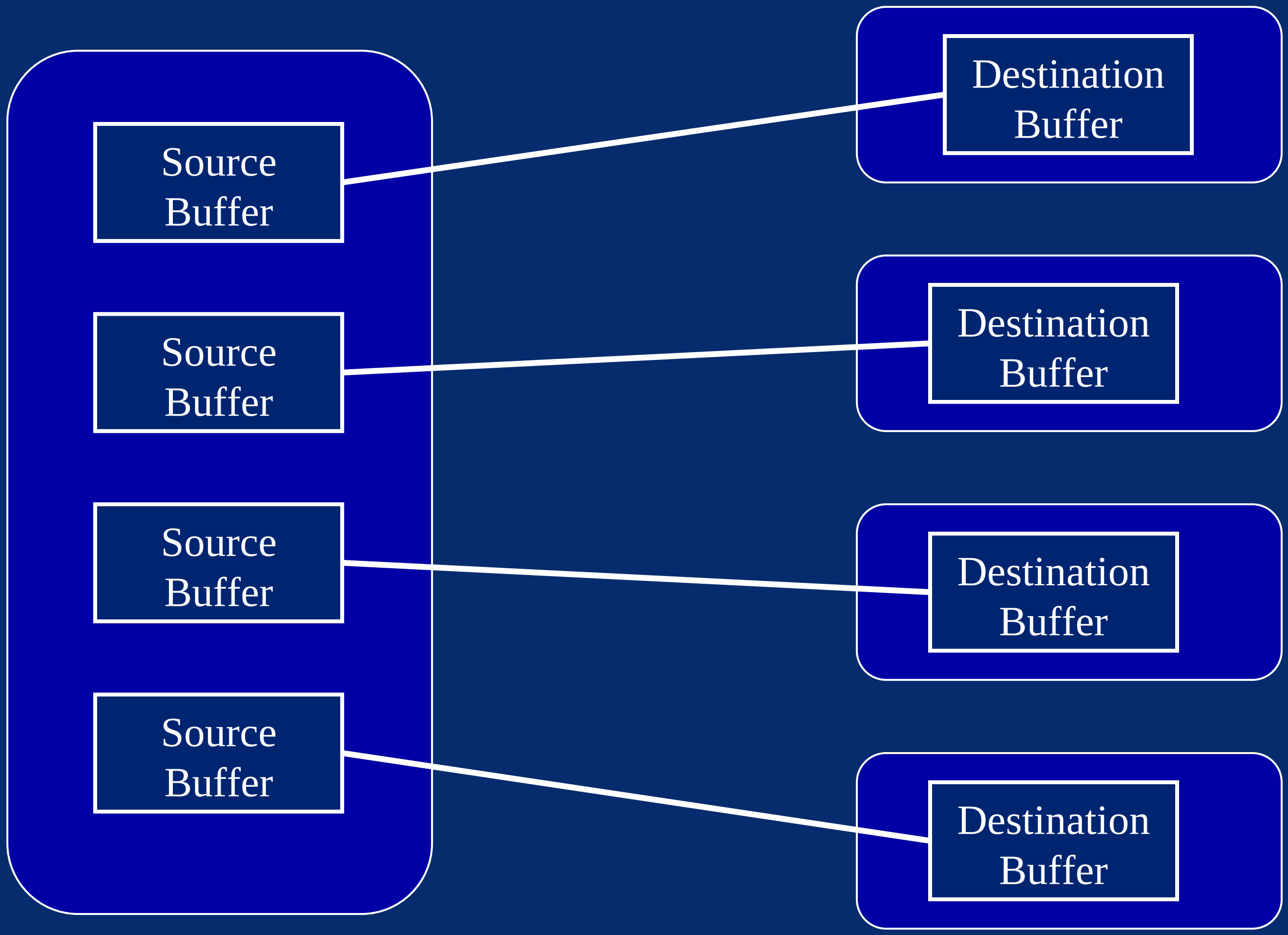
Channels

- One source to one destination



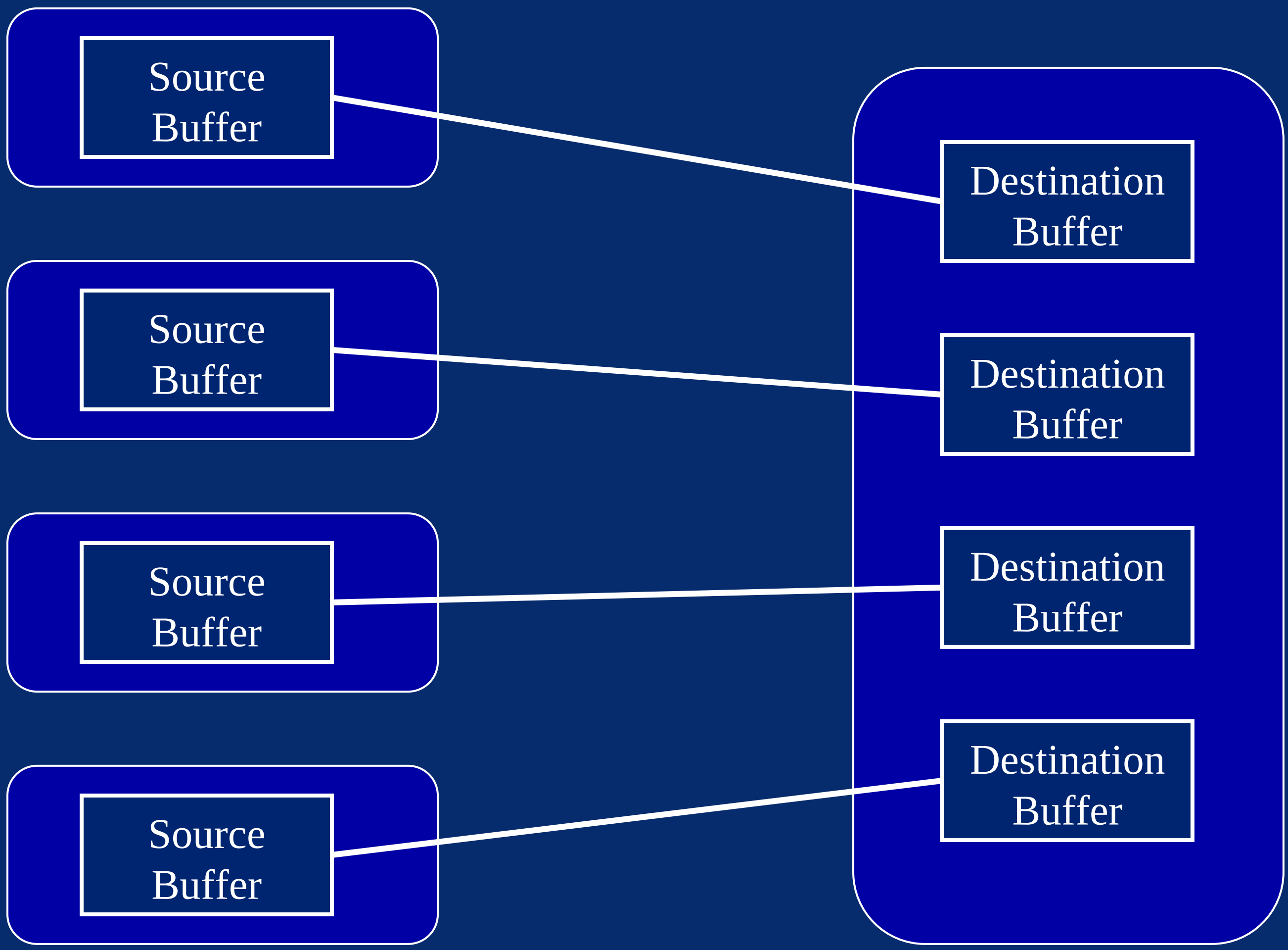
Channels

- One source to many destinations



Channels

- Many sources to one destination



Quality of Service

- Asynchronous:
  - Best Effort
  - Assured (reliable)
  - Priority and architectural constraints provide timeliness
- Synchronous
  - Best Effort
  - Assured (reliable)
  - Resource Reserved (timely)
  - Guaranteed (reliable, timely)

- Asynchronous discussed first
- Synchronous builds on asynchronous concepts

Space Technology Centre
University of Dundee

Space Technology Centre
University of Dundee

Functions

- User application interface
- Address resolution
- Segmentation
- End to end flow control
- Errors
- Redundancy
- Retry
- PDU encapsulation
- Priority
- Resource reservation

User Application Interface

- User writes into channel buffer in source node
- Information is transferred to channel buffer in destination node
- User reads from channel buffer in destination node
- User information:
  – Channel number
    - What channel is to be used to send the data
  – Data
    - Data to be transferred
  – Separator
    - Separates one piece of user information from the next
    - Used by segmentation to send last chunk of information straightaway

Space Technology Centre
University of Dundee

Address Translation

- 223 logical addresses
- Node identification done with logical address
- 223 separate nodes
- Path address may be used to route a packet to its destination

SpaceWire Logical Address	Priority	Prime SpaceWire Address	Redundant SpaceWire Address
120	-	120	120
124	-	1, 6, 5, 2, 124	2, 3, 5, 2, 124
132	low	132	132
132	high	133	133
150	-	1, 132	2, 132

Segmentation

- Size of user information is arbitrary
- Segmentation needed to enable:
  - – Interleaving of different sources of information over SpaceWire network
  - – Timely delivery of information
    - Don't have to wait for a large packet to finish sending
  - – Priority
    - Higher priority information can be sent in the middle of sending a low priority packet
  - – Determinism
    - Using scheduling where limited amount of information is sent in a given time-slot

Segmentation

- What size of segment?
- Compromise between
  - Efficiency
    - Large segment reduces cost of segment header
  - Timeliness
    - Small segments enable finer grained timely delivery
- Maximum segment size 255 bytes
- All segments are maximum size
- Except last one for a piece of user information

Space Technology Centre
University of Dundee

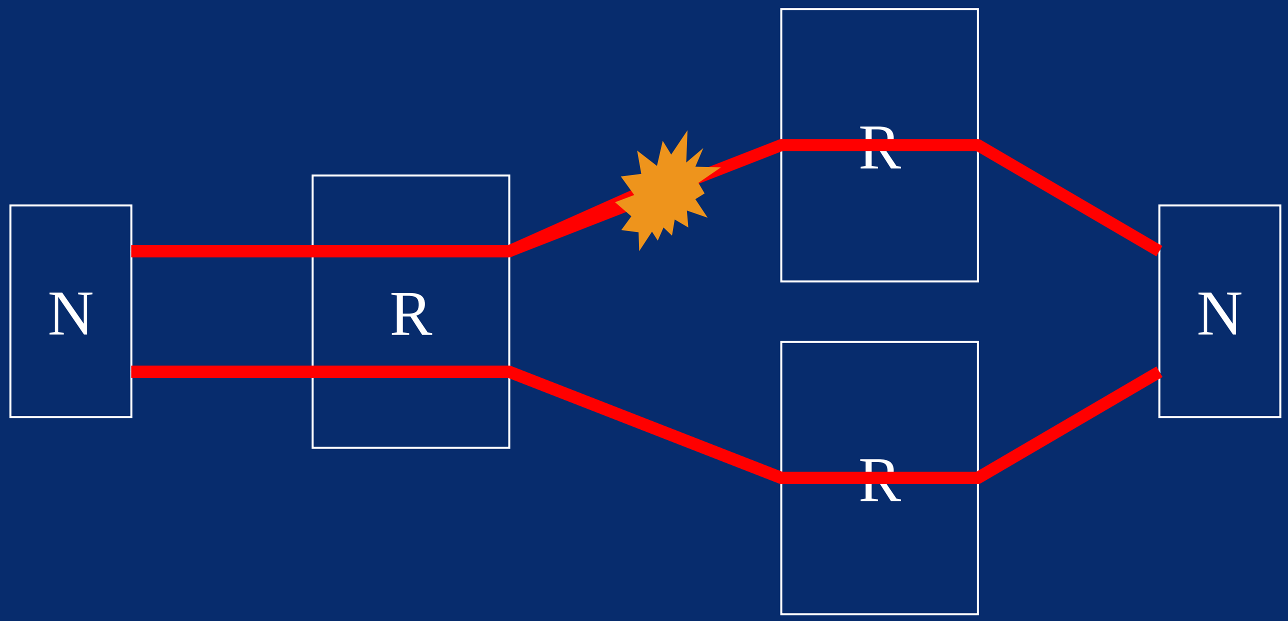
Error Detection

- Six possible types of error:
  - Header error
  - Wrong destination
  - Data error
  - Missing packet
  - Duplicate packet
  - SpaceWire EEP
- Best Effort and Resource Reserved
  - No retry in event of error
  - Errors logged and optionally reported
- Assured and Guaranteed
  - Retry in event of error
  - Persistent errors flagged

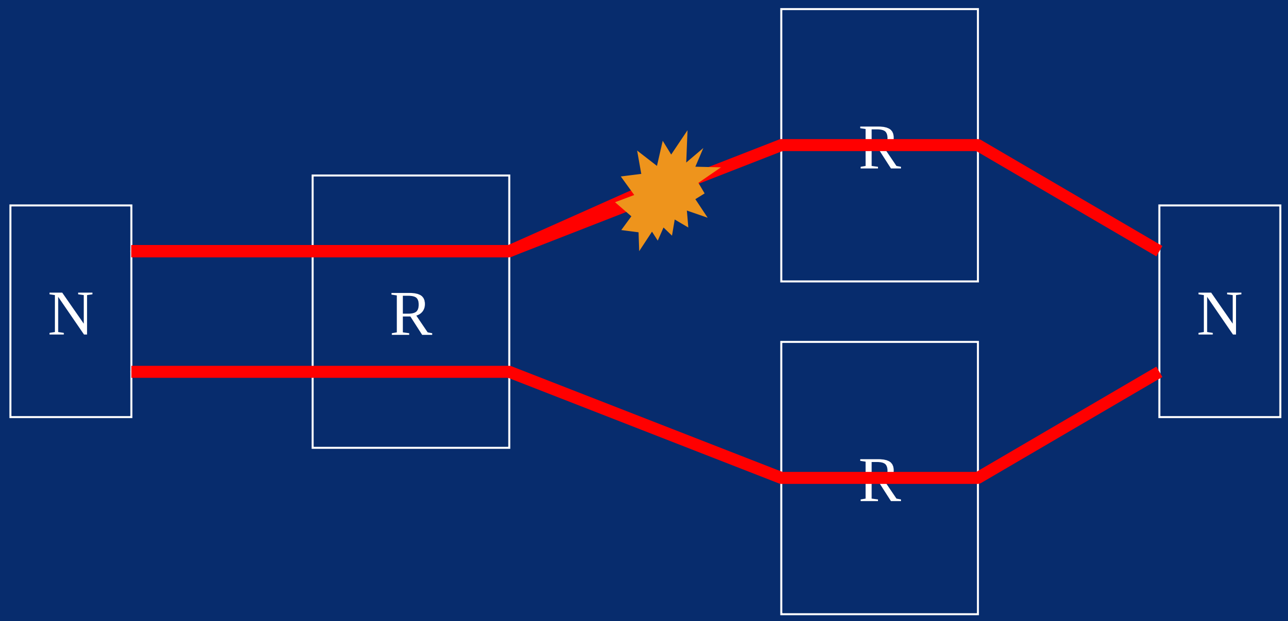
Redundancy

- Redundancy model:
  - Alternative paths from source node to destination node
- Managed and autonomous redundancy switching
- Redundancy supported in three ways:
  - Send over both paths simultaneously
  - Send over prime path then if failure send over redundant path
  - Send over either path then if failure send over remaining path

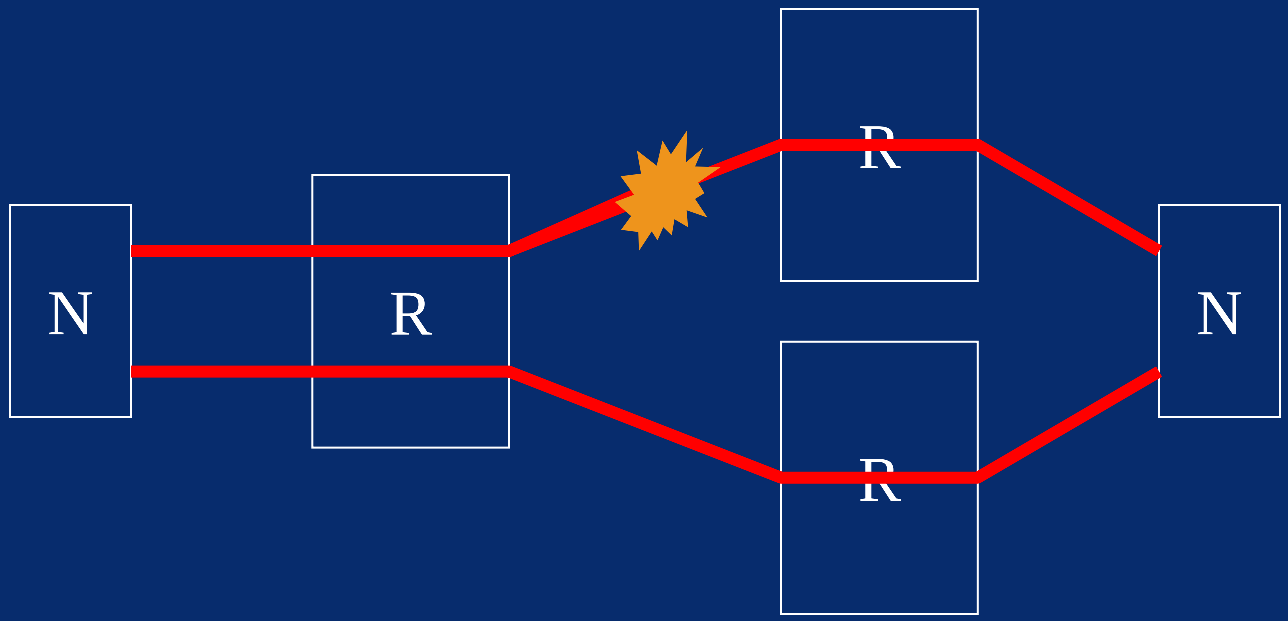
Both paths at same time



One path then alternative



Either path then remaining path



Redundancy Parameters

- Number of attempts on prime path (Np)
- Number of attempts on redundant path (Nr)
- Number of attempts on other alternative paths when appropriate
- Simultaneous retry on/off

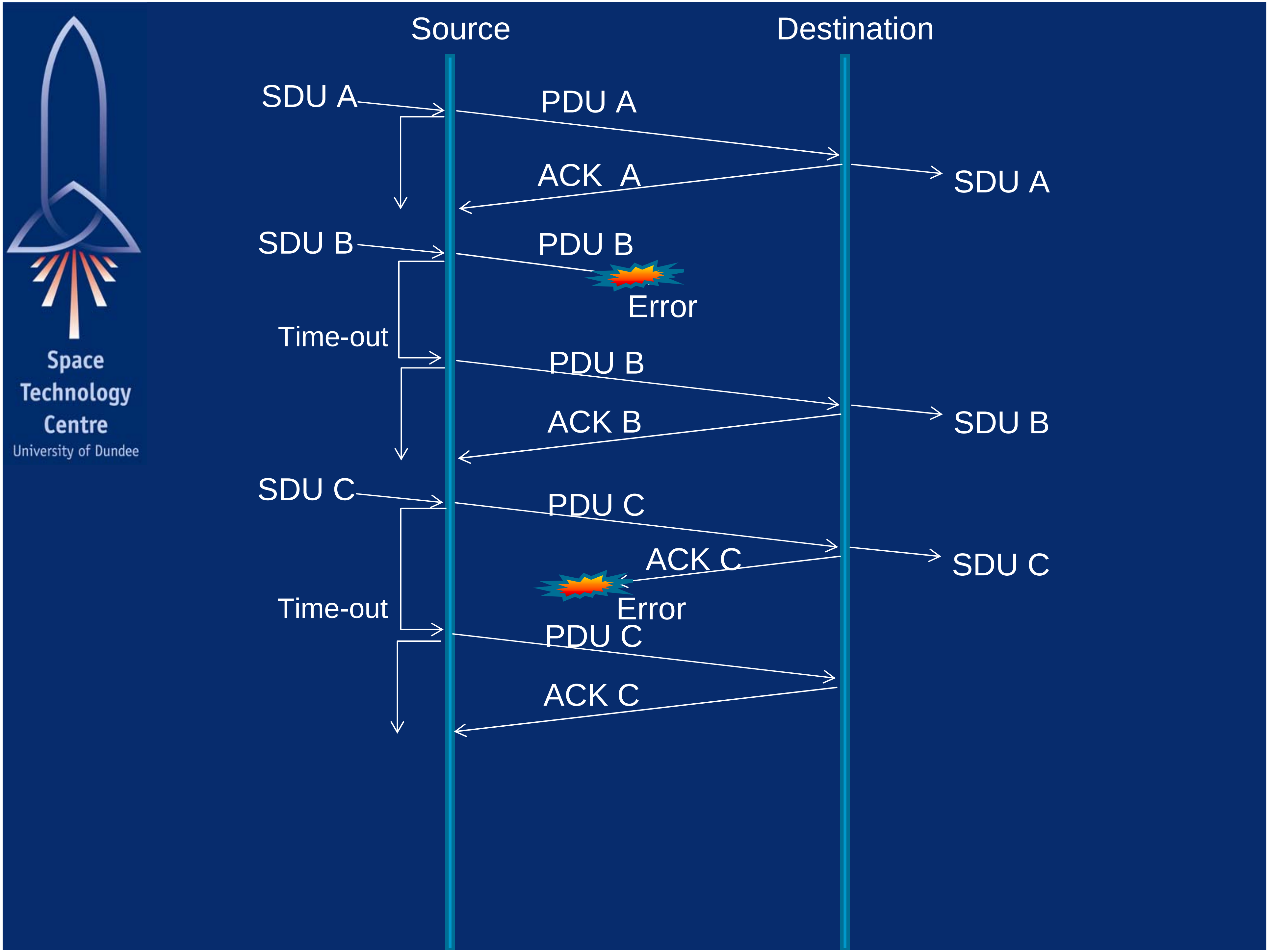
Space Technology Centre
University of Dundee

Redundancy

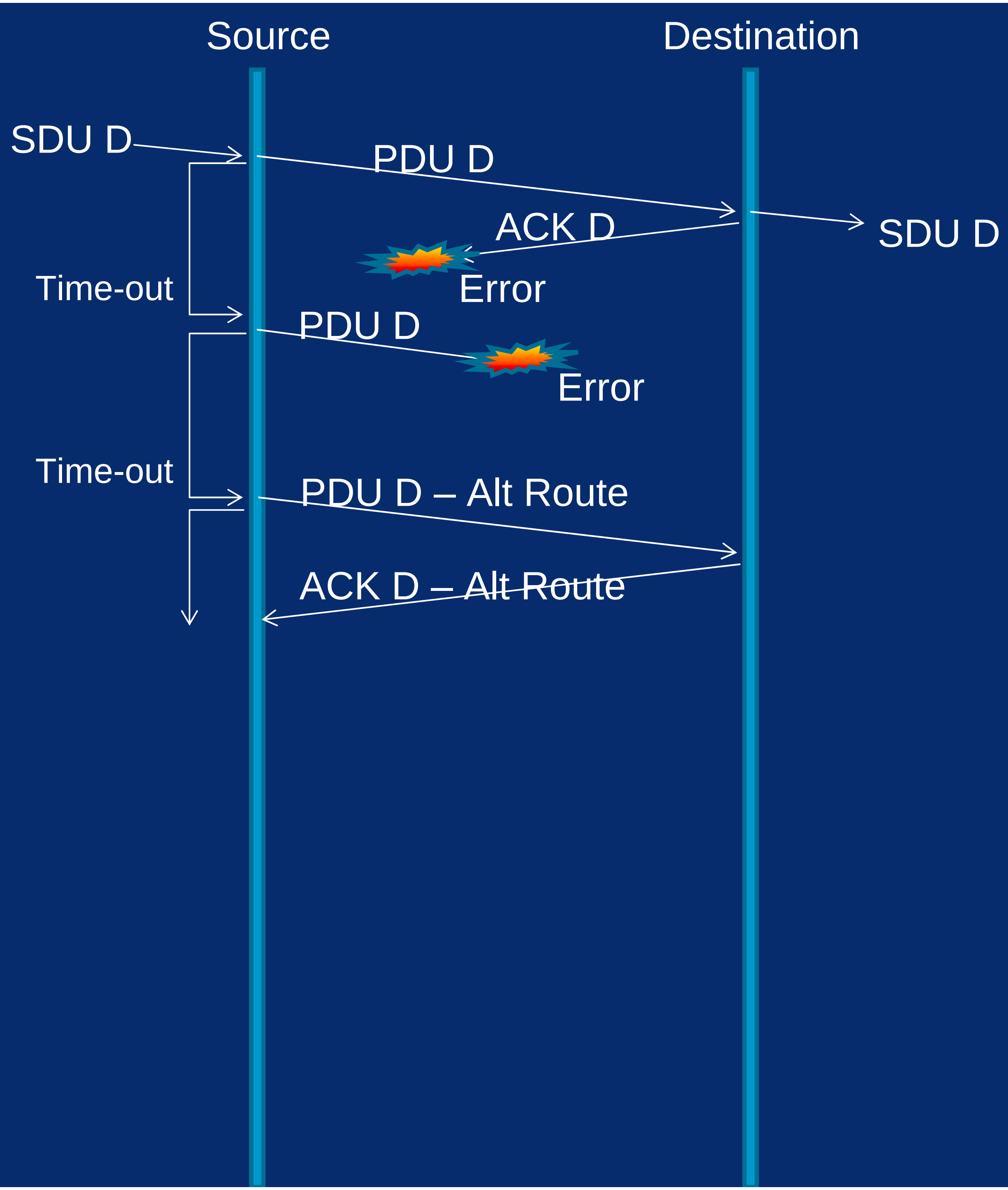
- Autonomous switching between equivalent data links supported.
- Controlled using management parameters.
- System management policy might dictate a uniform redundancy policy which applications must use.

Retry

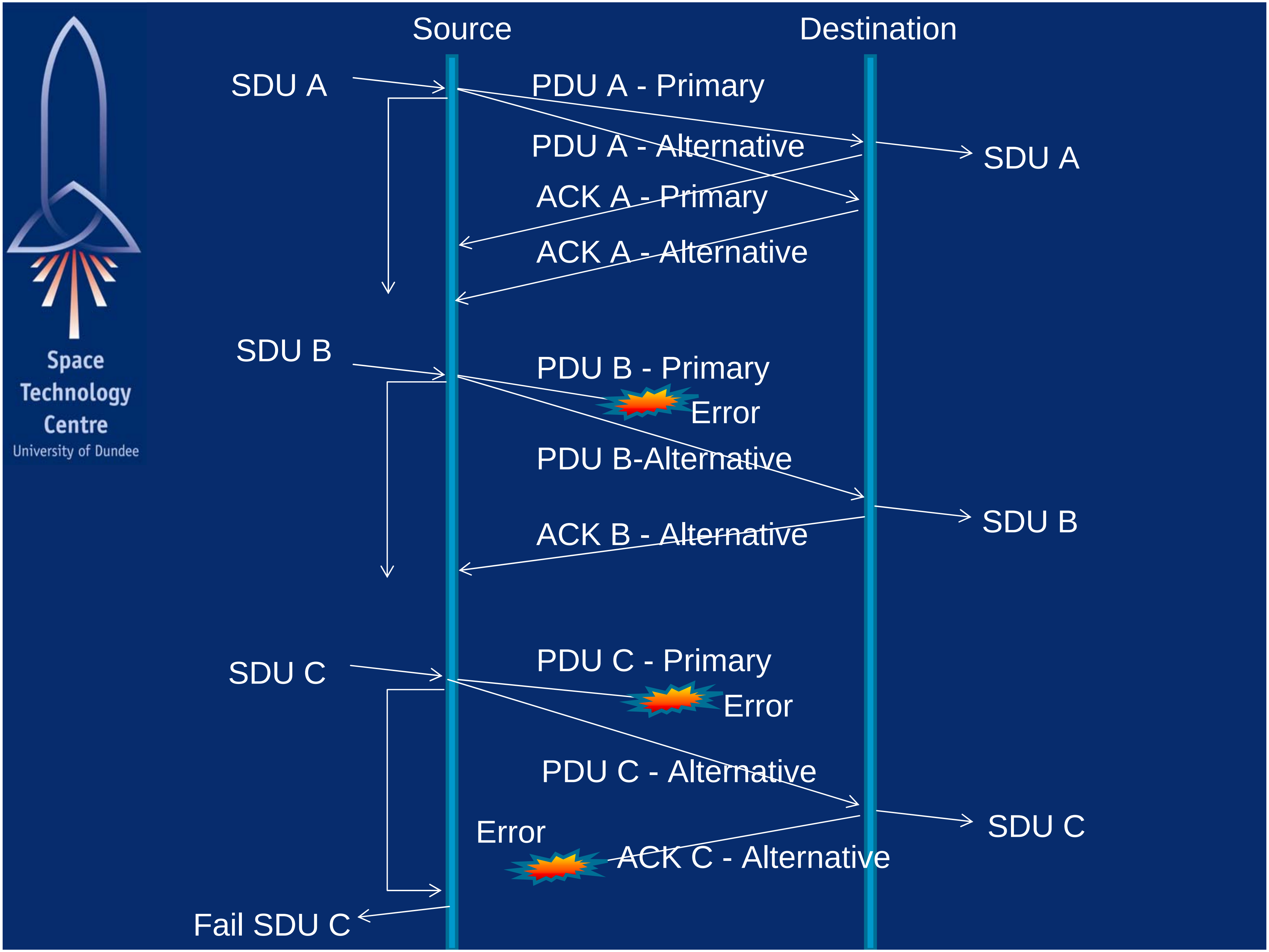
- Retry function necessary for reliability
  - In conjunction with redundancy
- Resends any segment
  - that goes missing
  - or that arrives in error
- Means that applications do not have to worry about this
- Delivery is ensured
- Simplifies application development
- Efficient implementation
- Flexible



Space
Technology
Centre
University of Dundee
Source
Destination
SDU A
PDU A
ACK A
SDU A
SDU B
PDU B
Error
Time-out
PDU B
ACK B
SDU B
SDU C
PDU C
ACK C
SDU C
Error
Time-out
PDU C
ACK C

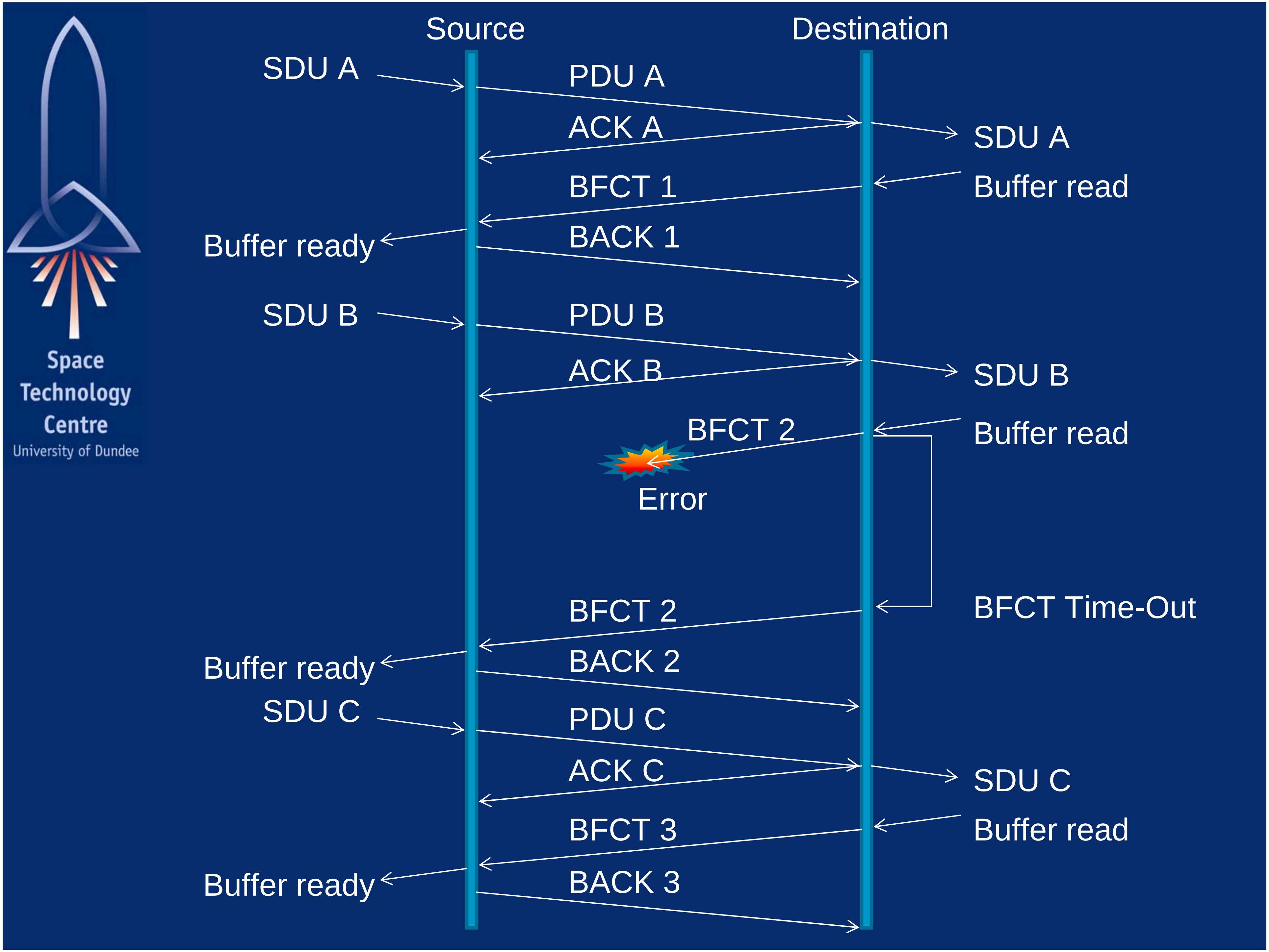


Source
Destination
SDU D
PDU D
SDU D
ACK D
Error
Time-out
PDU D
Error
Time-out
PDU D – Alt Route
ACK D – Alt Route

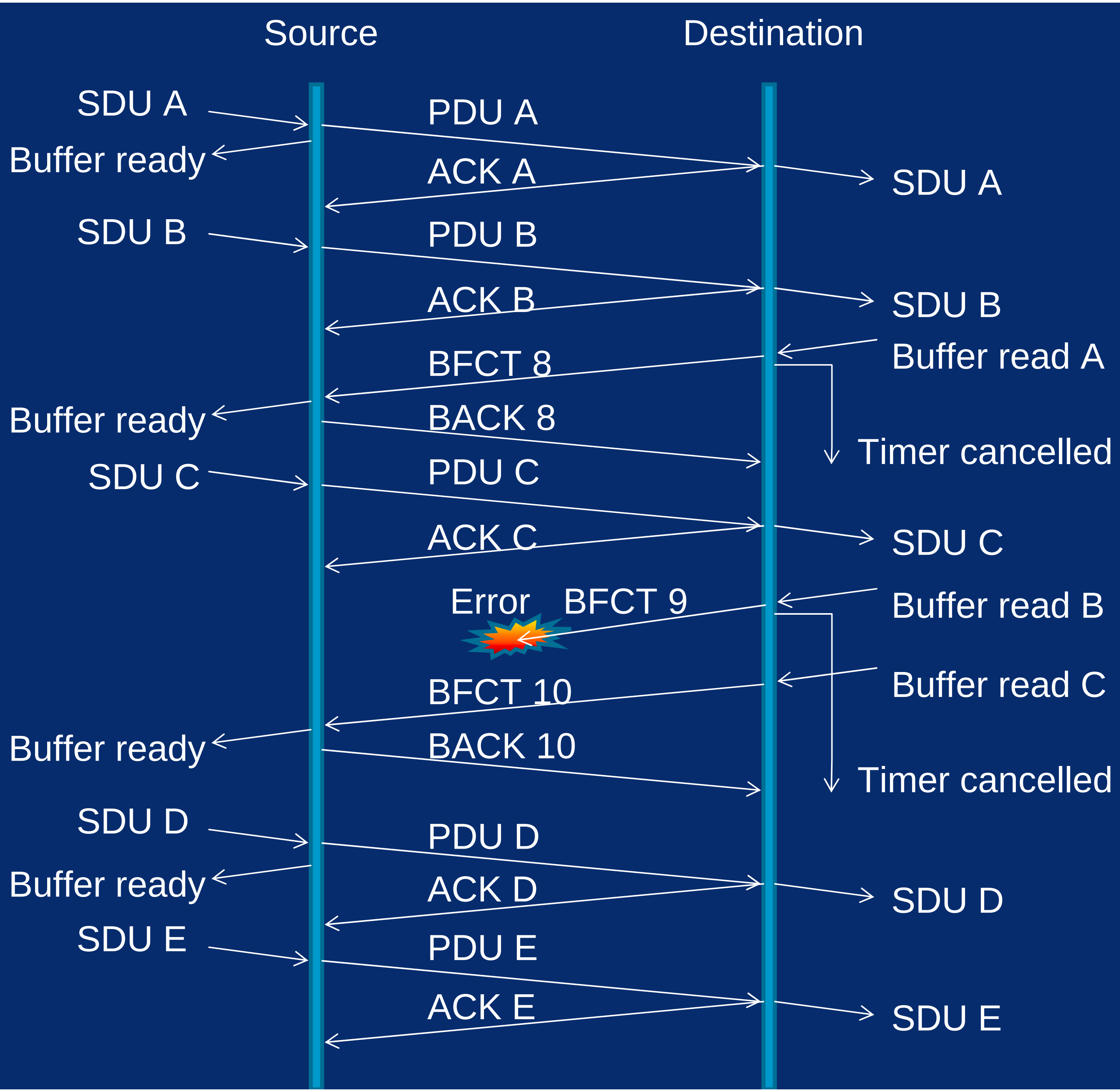

Space
Technology
Centre
University of Dundee
Source
Destination
SDU A
PDU A - Primary
PDU A - Alternative
SDU A
ACK A - Primary
ACK A - Alternative
SDU B
PDU B - Primary
Error
PDU B-Alternative
SDU B
ACK B - Alternative
SDU C
PDU C - Primary
Error
PDU C - Alternative
SDU C
Error
ACK C - Alternative
Fail SDU C

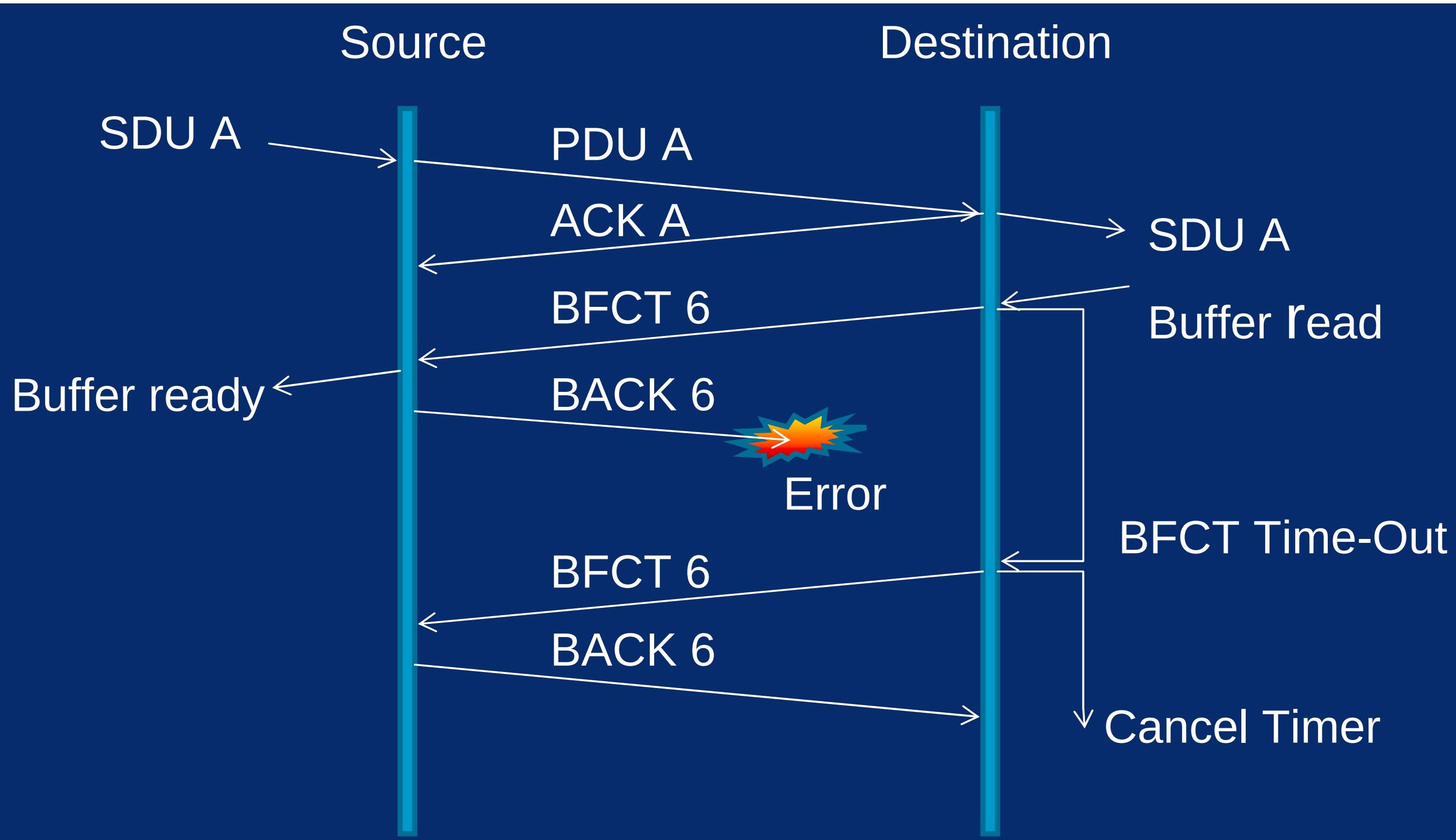
End to End Flow Control

- Why do we need flow control?
  - SpaceWire uses worm hole routing
  - A blockage at a destination
  - Can cause disruption through network
- Two options
  - Throw away packets if no room in destination buffer
    - Wastes system bandwidth
    - Hinders timeliness
  - Use flow control


Space Technology Centre
University of Dundee
Source
Destination
SDU A
PDU A
ACK A
SDU A
BFCT 1
Buffer read
Buffer ready
BACK 1
SDU B
PDU B
ACK B
SDU B
BFCT 2
Buffer read
Error
BFCT Time-Out
BFCT 2
Buffer ready
BACK 2
SDU C
PDU C
ACK C
SDU C
BFCT 3
Buffer read
Buffer ready
BACK 3

Space Technology Centre
University of Dundee




Source
Destination
SDU A
PDU A
ACK A
SDU A
BFCT 6
Buffer read
Buffer ready
BACK 6
Error
BFCT Time-Out
BFCT 6
BACK 6
Cancel Timer

Encapsulation

- Encapsulates
  - PDUs
  - ACKs
  - BFCTs
  - BACKs
  - Etc
- Into SpaceWire packets

Space Technology Centre
University of Dundee

Space Technology Centre
University of Dundee

PDU Encapsulation

First byte sent

	Destination SpW Address	Destination SpW Address	Destination SpW Address
Destination Logical Address	SpW Protocol ID	Source Logical Address	Channel
Type / Redundancy	Sequence Number	Data Length	Header CRC
Data	Data	Data	Data
Data	Data	Data	Data
Data	Data	Data	Data
Data	Data	Data	Data
CRC MS	CRC LS	EOP	

Last byte sent

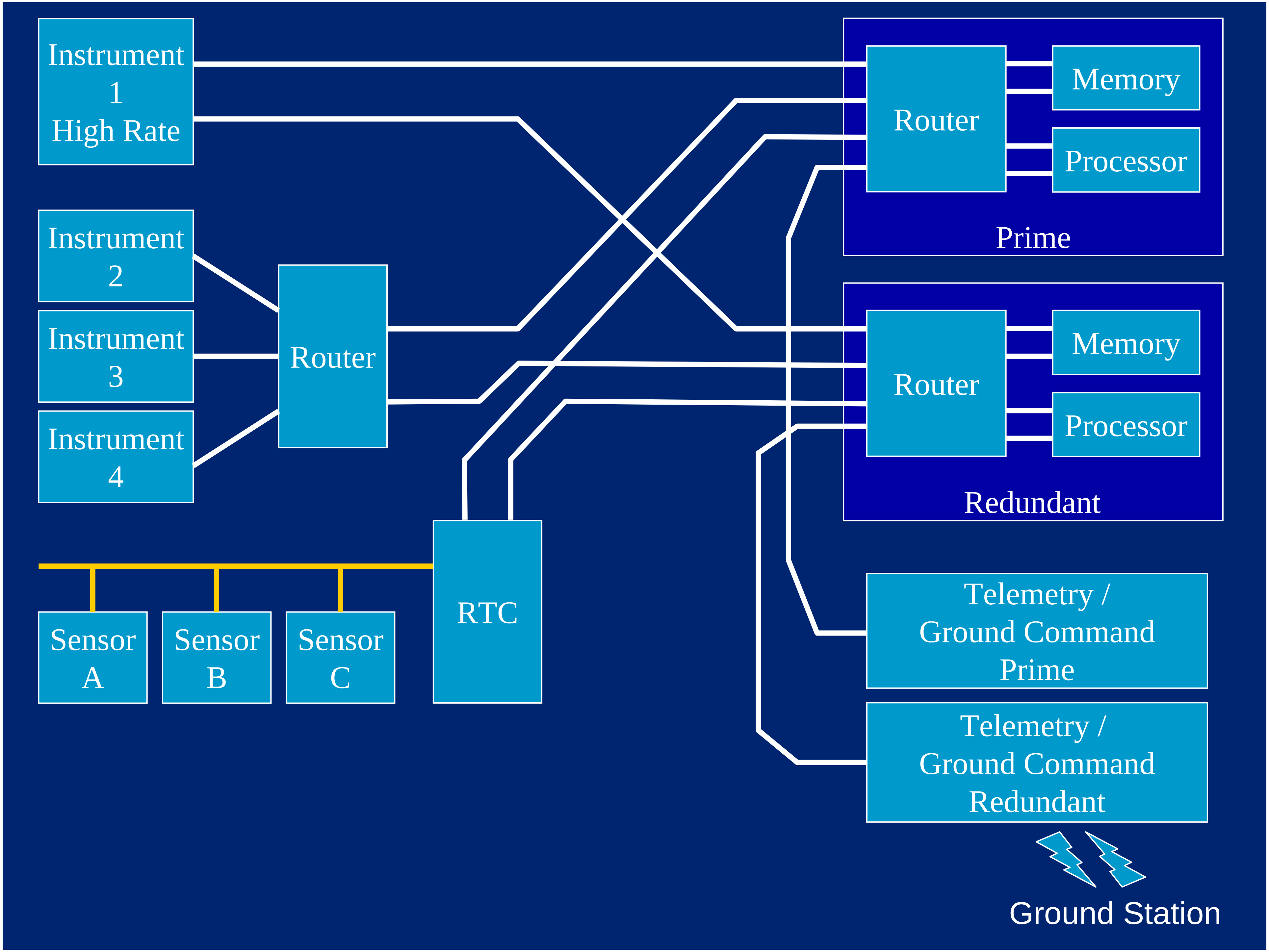
Priority

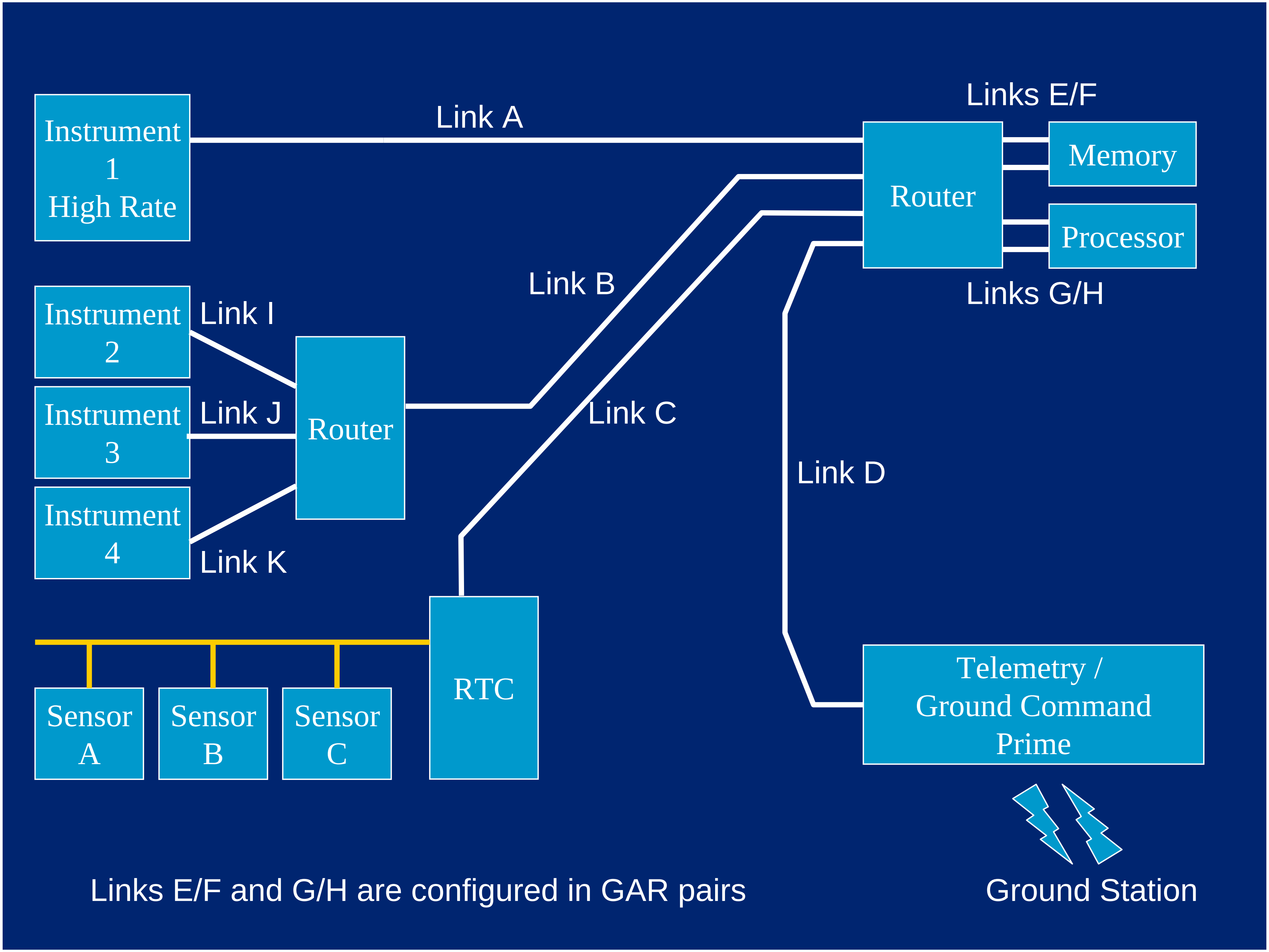
- Priority provided for both asynchronous and synchronous systems
- Separate source and destination channel buffers used for each level of priority
- PDU sent from source channel buffer with lowest number first.
  - Simple approach
  - Other alternatives being considered

Space Technology Centre
University of Dundee

Synchronous

- Reliable AND Timely
- Need to manage network bandwidth

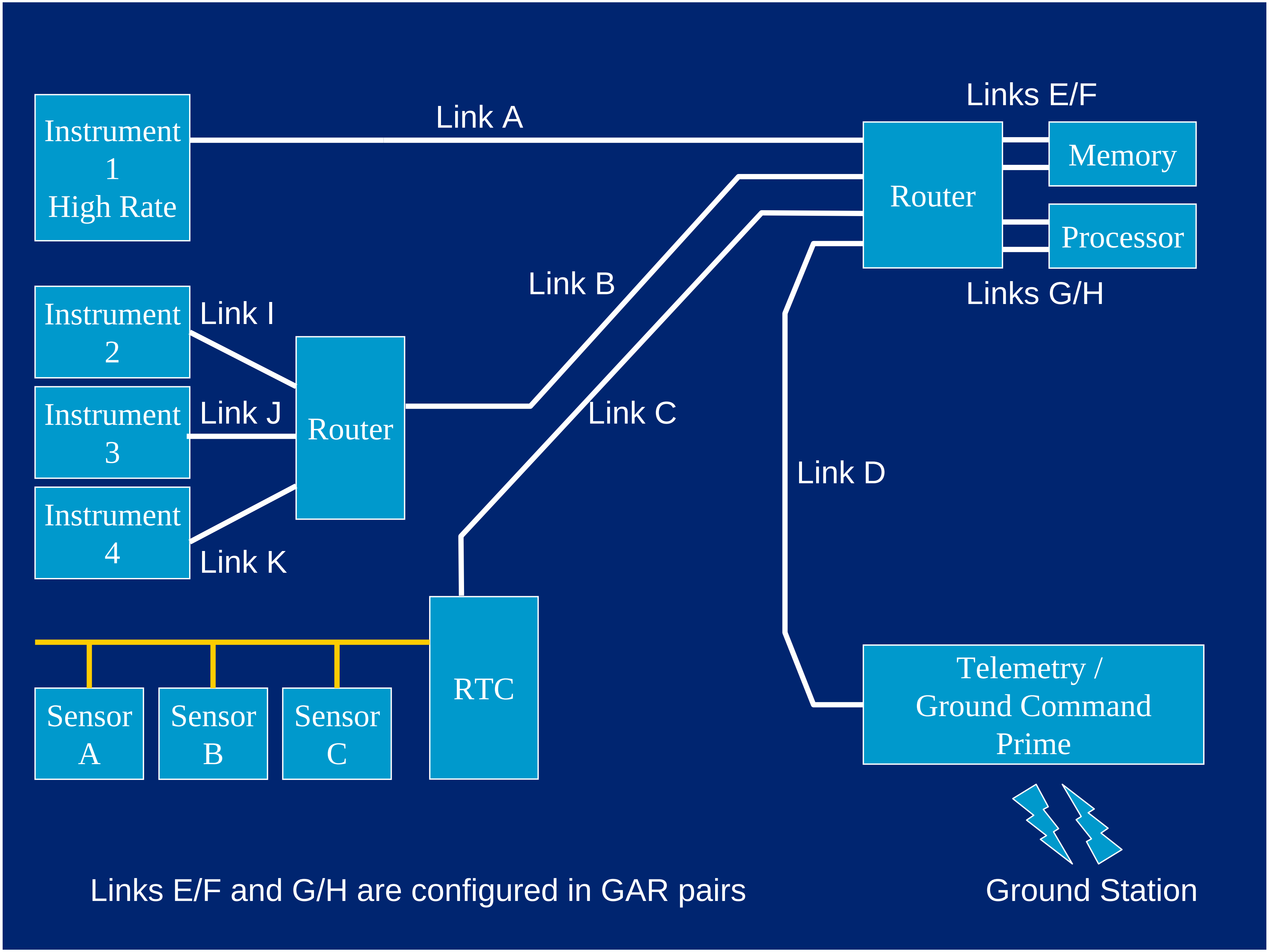

Instrument
1
High Rate
Instrument
2
Instrument
3
Instrument
4
Router
Router
Memory
Processor
Prime
Router
Memory
Processor
Redundant
RTC
Sensor
A
Sensor
B
Sensor
C
Telemetry /
Ground Command
Prime
Telemetry /
Ground Command
Redundant
Ground Station


Instrument 1 High Rate
Link A
Links E/F
Router
Memory
Processor
Links G/H
Instrument 2
Link I
Instrument 3
Link J
Instrument 4
Link K
Router
Link B
Link C
Link D
RTC
Sensor A
Sensor B
Sensor C
Telemetry / Ground Command Prime
Links E/F and G/H are configured in GAR pairs
Ground Station

Links are Critical Resources

- Multiple data paths share some links
- Routers are non-blocking
- So if the output link is not blocked the inputs will not be blocked
- Routers are not critical resources

Table 3-1 Utilisation of resources (links)		
Link	Left to right / up	Right to left/ down
A	Not shared	Processor commands and ground commands
B	Instruments 2, 3, 4	Processor commands and ground commands
C	RTC	Processor commands and ground commands
D	Telecommands	Data from memory for down link
E/F	Instruments 1, 2, 3, 4 and RTC Processor commands and ground commands	Date from memory for down link
G/H	Data from instruments or memory for processing	Processor commands Processed data
I	Not shared	Processor commands and ground commands
J	Not shared	Processor commands and ground commands
K	Not shared	Processor commands and ground commands



Instrument 1 High Rate
Link A
Links E/F
Router
Memory
Processor
Links G/H
Link B
Instrument 2
Link I
Instrument 3
Link J
Router
Link C
Instrument 4
Link K
Link D
RTC
Sensor A
Sensor B
Sensor C
Telemetry / Ground Command Prime
Ground Station
Links E/F and G/H are configured in GAR pairs

Table 3-2 Channel allocations			
Channel No.	Traffic	Links used L to R / Up	Links used R to L / Down
1	Instrument 1 to memory	A, E/F	
2	Instrument 2 to memory	I, B, E/F	
3	Instrument 3 to memory	J, B, E/F	
4	Instrument 4 to processor for processing	K, B, G/H	
5	RTC sensor data to memory	C, E/F	
6	Processor to memory – processed data	E/F	g/h
7	Memory to telemetry		e/f, d
8	Processor commands to any other unit	E/F	g/h, a, b, c, i, j, k
9	Ground commands to any other unit	D, E/F, G/H	a, b, c, i, j, k

Scheduling

- Delivery of PDUs according to a predefined schedule.
- May be used
  – to support deterministic data delivery
  – to reserve bus/sub-network bandwidth
- Scheduling function splits up the bandwidth on a bus/sub-network using time division multiplexing.
- A number of equal duration time-slots are used.

Time-slots

- Means of dividing network bandwidth between channels
- Equal divisions of time
  - during which a discrete set of communications can take place
- Time-slots distributed by SpaceWire time-codes
- 64 time-code values
- 64 time-slots for minor cycle or epoch
- Used to separate time-slots in a scheduled system

Scheduled System

- When a time-code arrives
- Any node that has a channel scheduled to communicate in that time-slot
- Can send a packet
- Or one packet and allow time for a retry

Scheduling

- Knowledge about the communication schedule held in each end-point
- So that they know when (i.e. in which time-slot) they are allowed to transmit data.

Space Technology Centre
University of Dundee

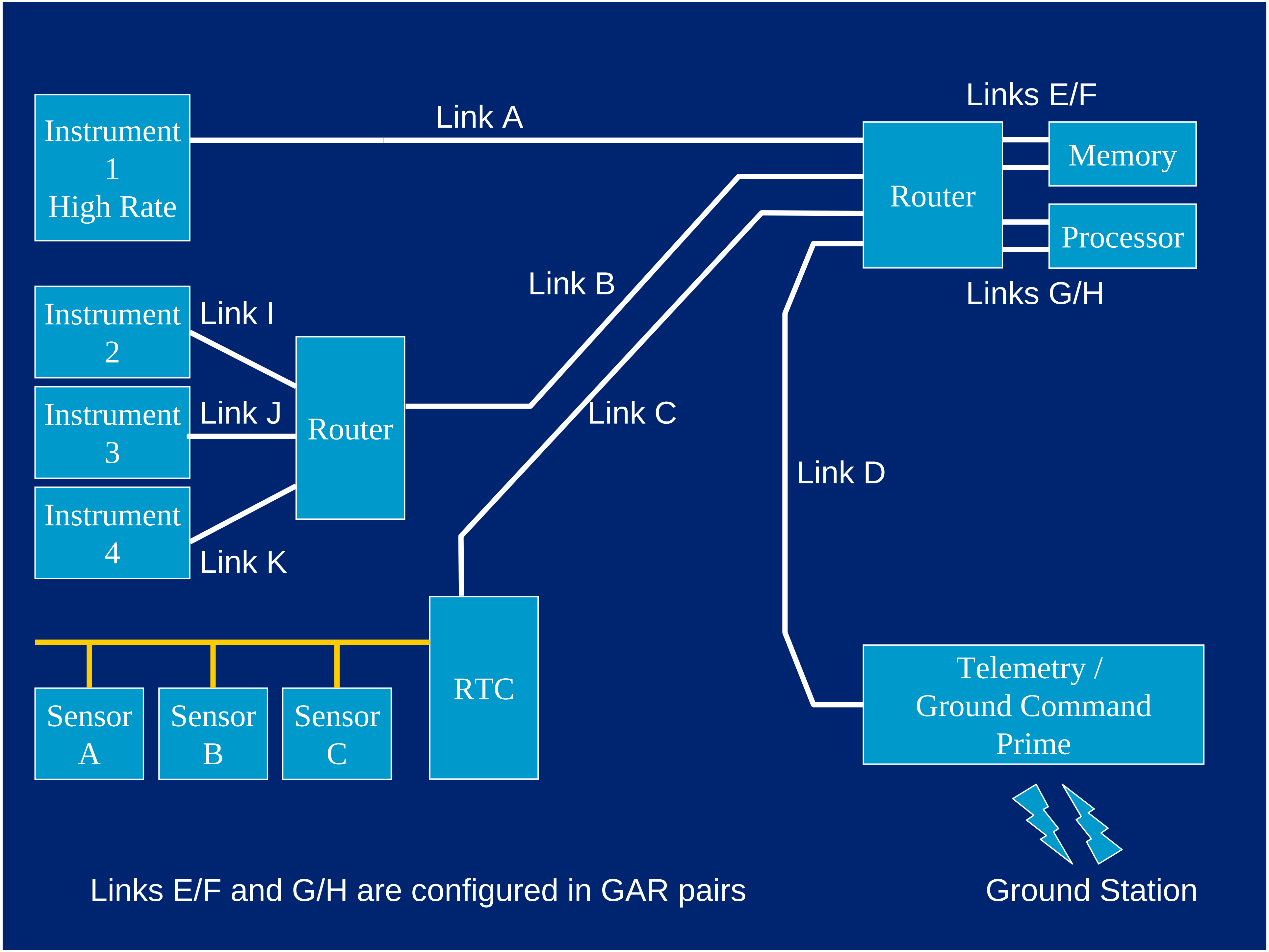
Scheduling on SpaceWire

- Channels assigned to time-slots
- So that there is no conflicting resources
- Assignment of channels to time-slots is a schedule table

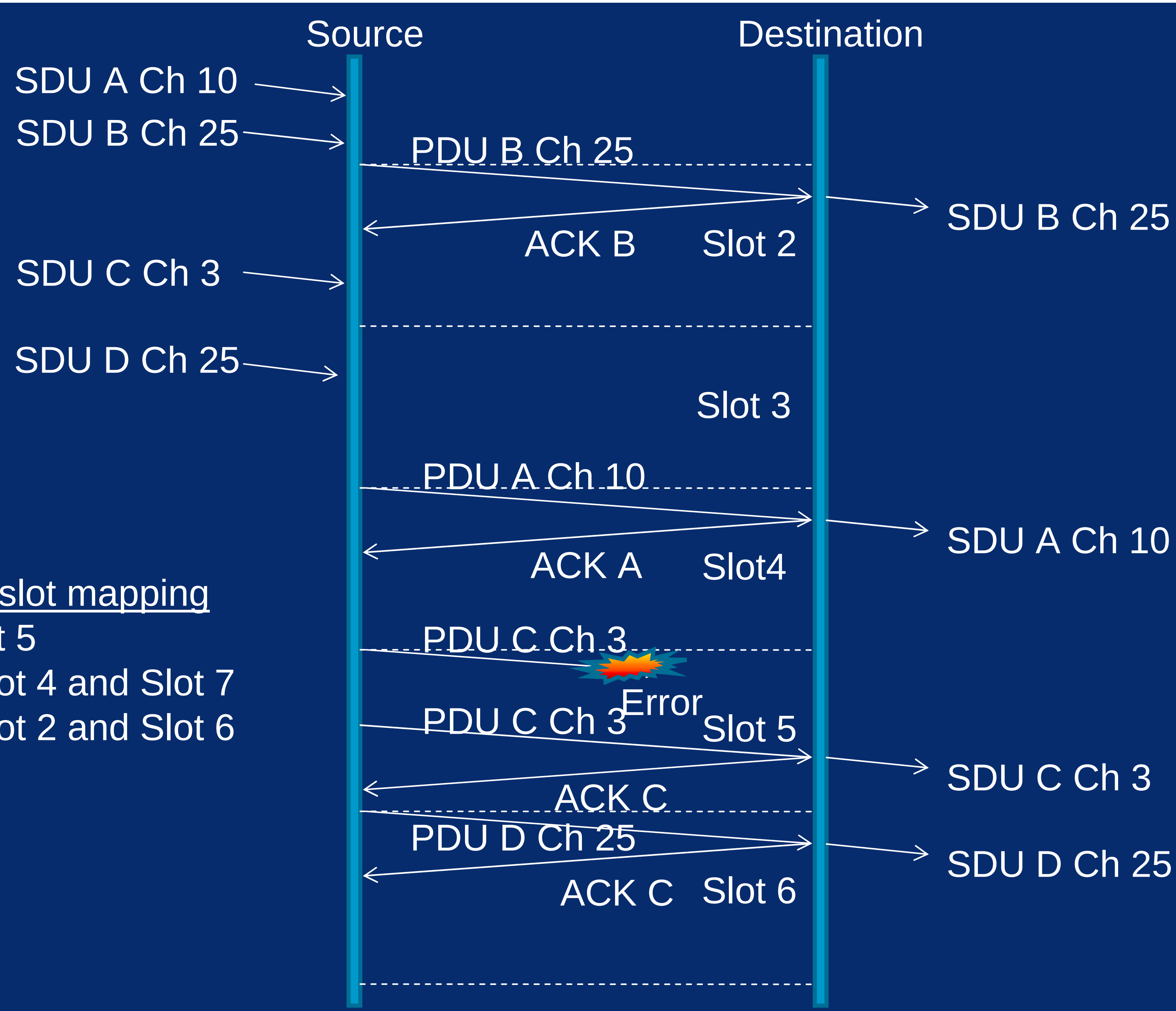
Space
Technology
Centre
University of Dundee

Slot Allocation in a Scheduled System

	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7	Slot 8	…	Slot 63
Channel 1	A, E/F	A, E/F	A, E/F	A, E/F	A, E/F	A, E/F	A, E/F	A, E/F			A, E/F
Channel 2	I, B, E/F				I, B, E/F						
Channel 3		J, B, E/F									
Channel 4			K, B, G/H								
Channel 5				C, E/F							
Channel 6						g/h, E/F					
Channel 7	e/f, d	e/f, d	e/f, d	e/f, d	e/f, d	e/f, d	e/f, d	e/f, d			e/f, d
Channel 8							E/F, g/h, a,b,c,I,j,k				
Channel 9								D,E/F,G/H a,b,c,i,j,k			D,E/F,G/H a,b,c,i,j,k



Instrument 1 High Rate
Link A
Links E/F
Router
Memory
Processor
Links G/H
Instrument 2
Link I
Instrument 3
Link J
Router
Instrument 4
Link K
Link B
Link C
Link D
RTC
Sensor A
Sensor B
Sensor C
Telemetry / Ground Command Prime
Links E/F and G/H are configured in GAR pairs
Ground Station



Channel to slot mapping

Ch 3 -> Slot 5

Ch 10 -> Slot 4 and Slot 7

Ch 25 -> Slot 2 and Slot 6

Summary

- Protocols discussed in April 2007
  - Developed by CCSDS TCONS
- Requirements formalised
- Initial protocol definition
- Prototyping of various protocols to reduce risk
- Feedback from SpW WG on initial protocol definition
- Next steps
  - Revised protocol definition
  - Protocol implementation and testing

Space Technology Centre
University of Dundee