SpaceWire-T
&
SpaceWire-RT

Steve Parkes, Albert Ferrer-Florit
Space Technology Centre,
University of Dundee

Space
Technology
Centre
University of Dundee

Contents

- Key drivers for SpaceWire-RT
- QoS provided
- SpaceWire-T
- Scheduled system
- Implementation
- SpaceWire-T vs SpaceWire-RT

Space Technology Centre
University of Dundee

Key drivers for SpaceWire-RT

- QoS layer for SpaceWire
- Provide a range of QoS to meet requirements of most missions
  - As defined by CCSDS SOIS
- Support past paradigms and new ones
  - Application level FDIR
  - Network level FDIR for PnP and ORS
- Able to be implemented using existing SpaceWire devices
  - E.g. SpW-10X and RTC
- Simple and low implementation overhead

Space Technology Centre
University of Dundee

QoS provided

- CCSDS SOIS QoS adopted
- Best Effort
  - Single attempt to deliver
  - In order delivery
- Assured (R)
  - Retry in event of failure to deliver
- Resource Reserved (T)
  - Single attempt
  - Over reserved network resource
- Guaranteed (RT)
  - Reserved network resource
  - Retry in event of failure to deliver

Space Technology Centre
University of Dundee

SpaceWire-T

- Directed by ESA to investigate
  - Removing retries and redundancy switching from SpaceWire-RT
  - Include ACK in the Best Effort and Resource-Reserved services
- Aims to
  - Simplify implementation
  - Reduce amount of testing

SpaceWire-T

- Over Asynchronous network provides:
  - Best Effort QoS
  - ACK for Best Effort QoS
    - So that the end user application is informed when something fails to be delivered
- Over Scheduled network provides:
  - Determinism
  - Resource Reserved QoS
  - With ACK

SpaceWire-T

- Does NOT provide
  - Retries:
    - Left up to each application to implement
  - Redundancy:
    - Responsibility of system level application
    - Which is not part of the network
  - Assured or Guaranteed QoS

Determinism

- Determinism provided by reserving network resources
- Time-slots used to split up network bandwidth into intervals that can be reserved
- Schedule table determines who can send in each time-slot
- Time-slots defined using time-codes

Schedule using Time-slots

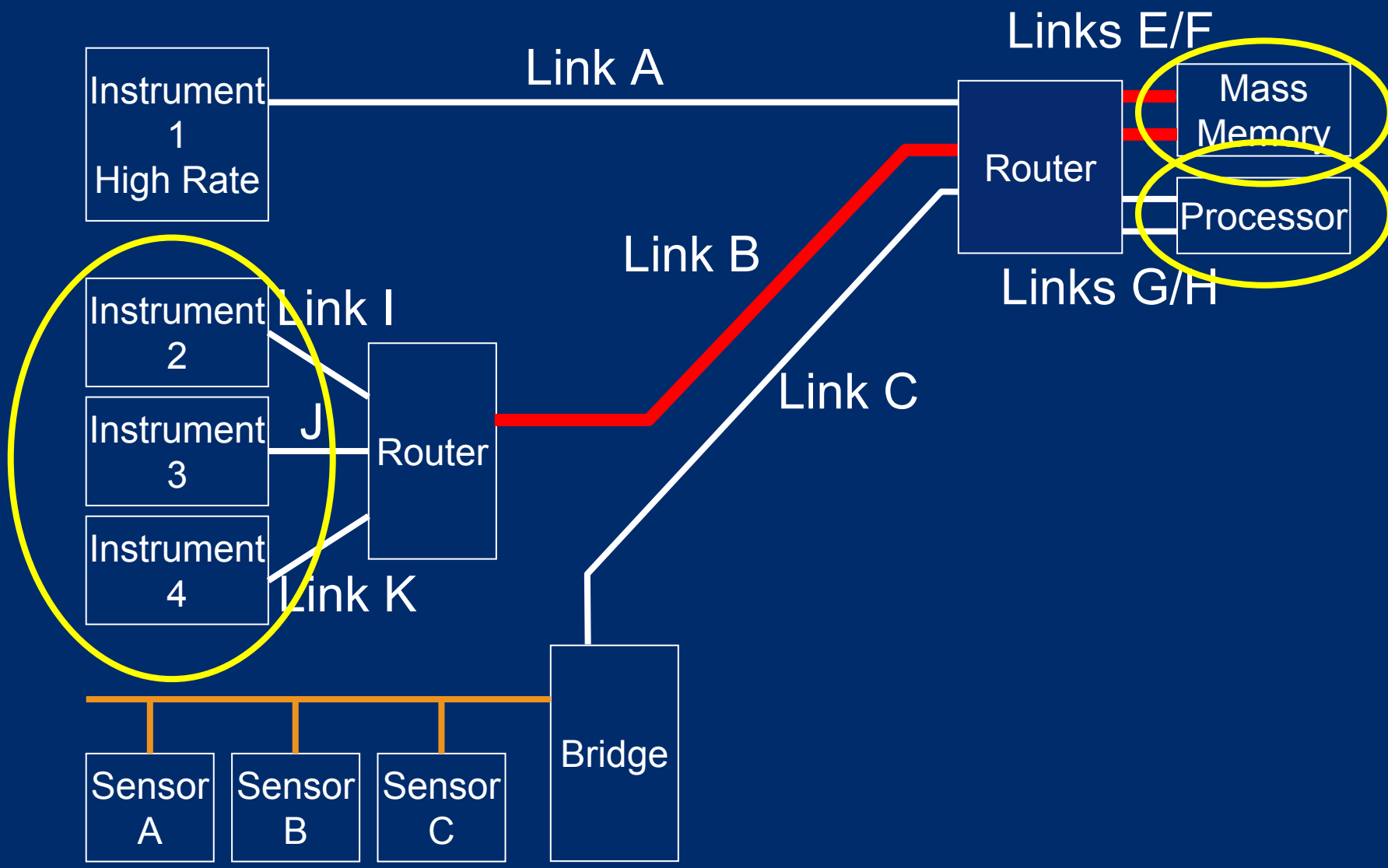
	Time-slot										
Channel	0	1	2	3	4	5	6	7	8	…	63
41/42/1											
52/70/1											
53/70/1											
54/70/1											
60/60/1											
80/70/1											
80/54/1											

Source / Destination / Channel Number

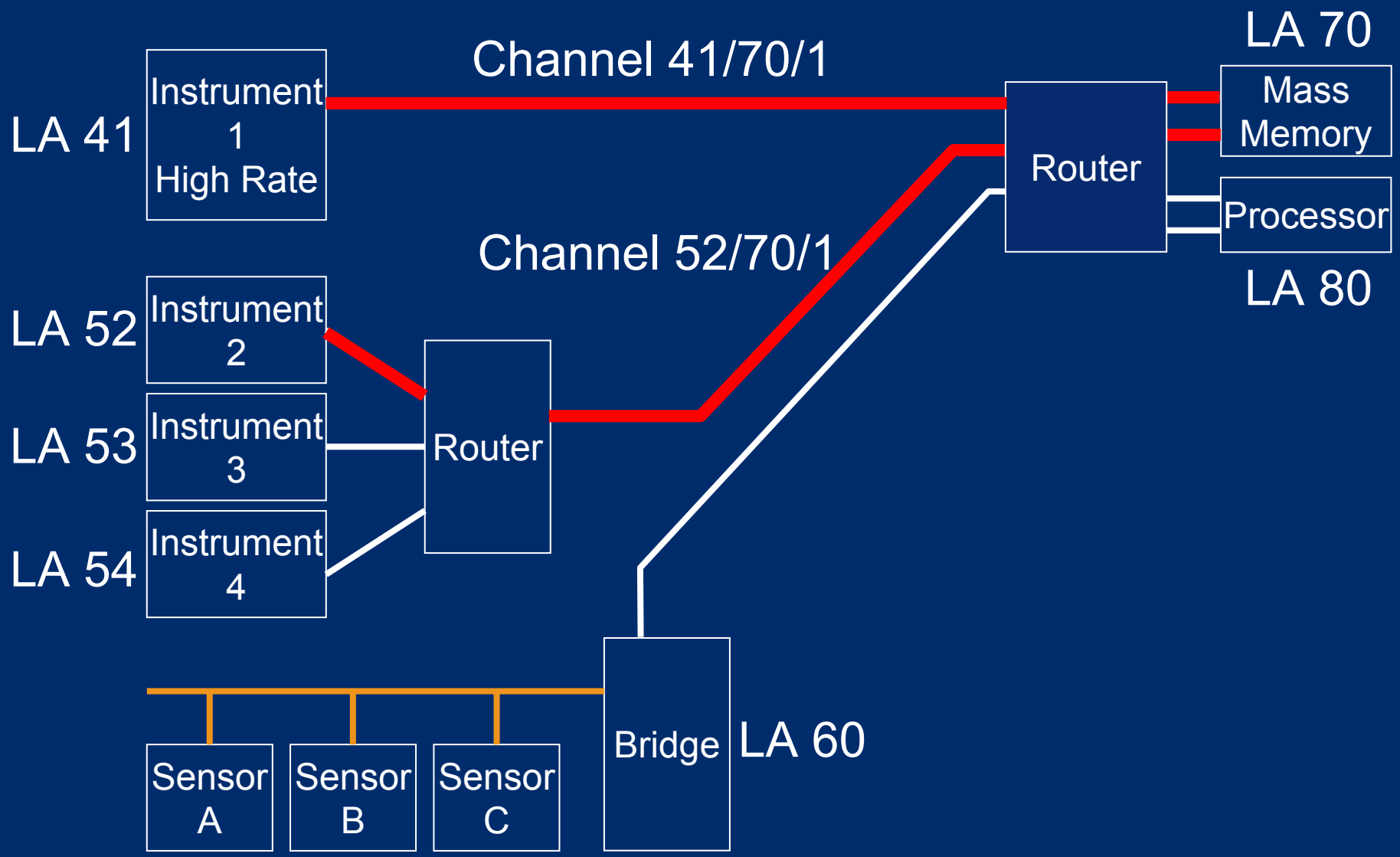
SpaceWire Links as Resources

- Resources to be managed are:
  - SpaceWire links
- Routers need not be managed
  - Since they are “non-blocking” switches
  - If the links are available the switch will always be able to forward a packet
- Source and destination buffers
  - Allocated to each channel
  - No conflict

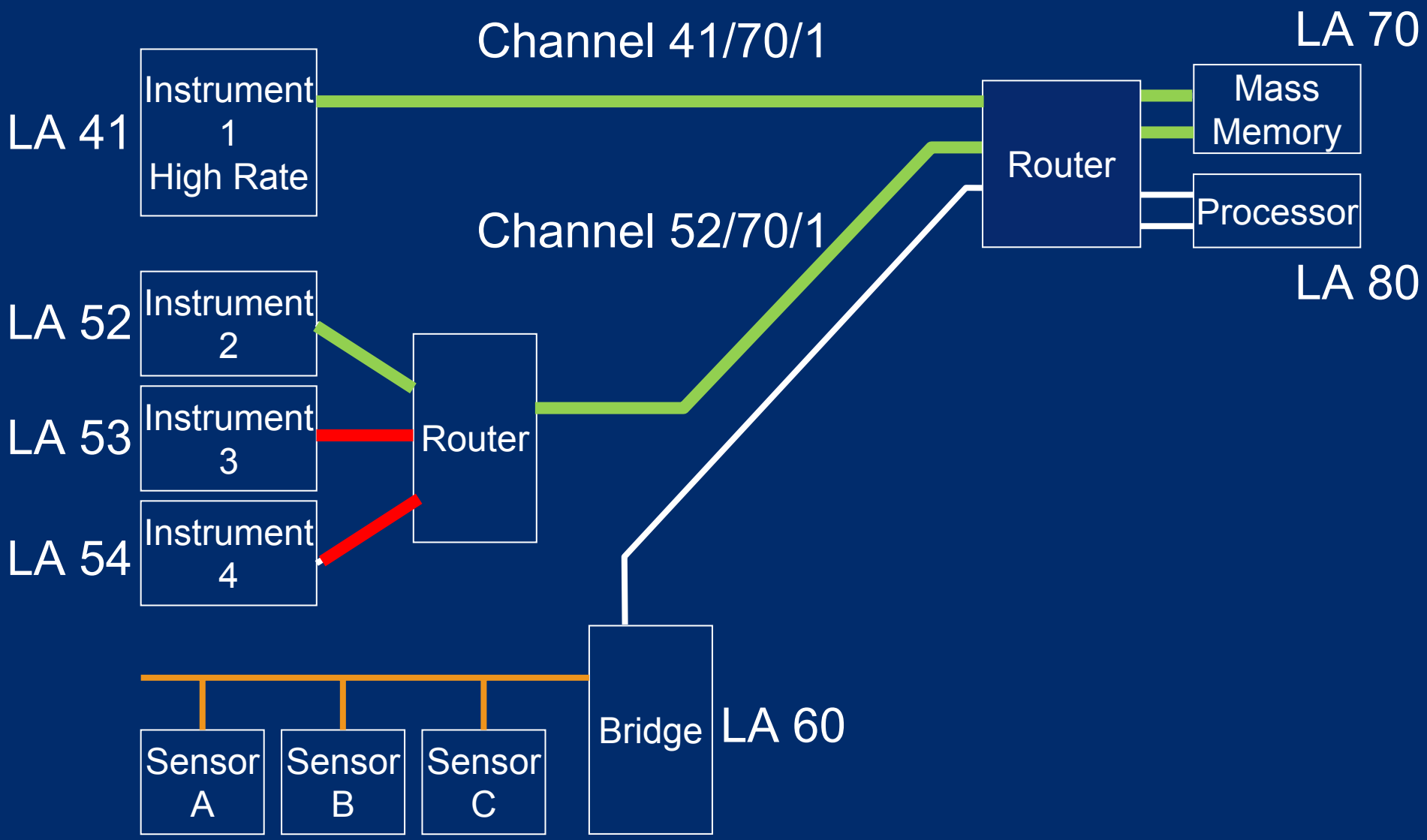
Example Shared Resources



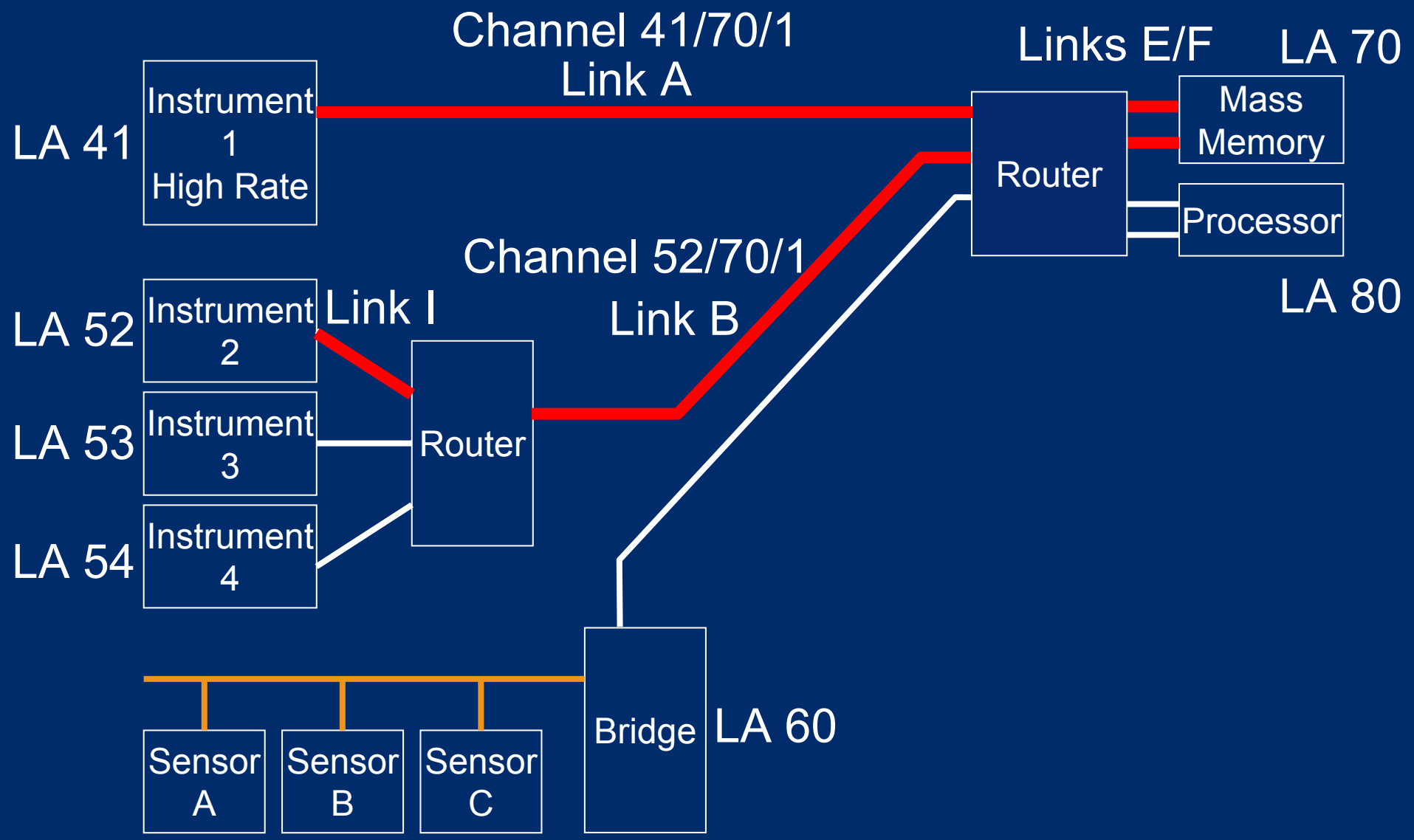
Example Channels



Example Scheduling



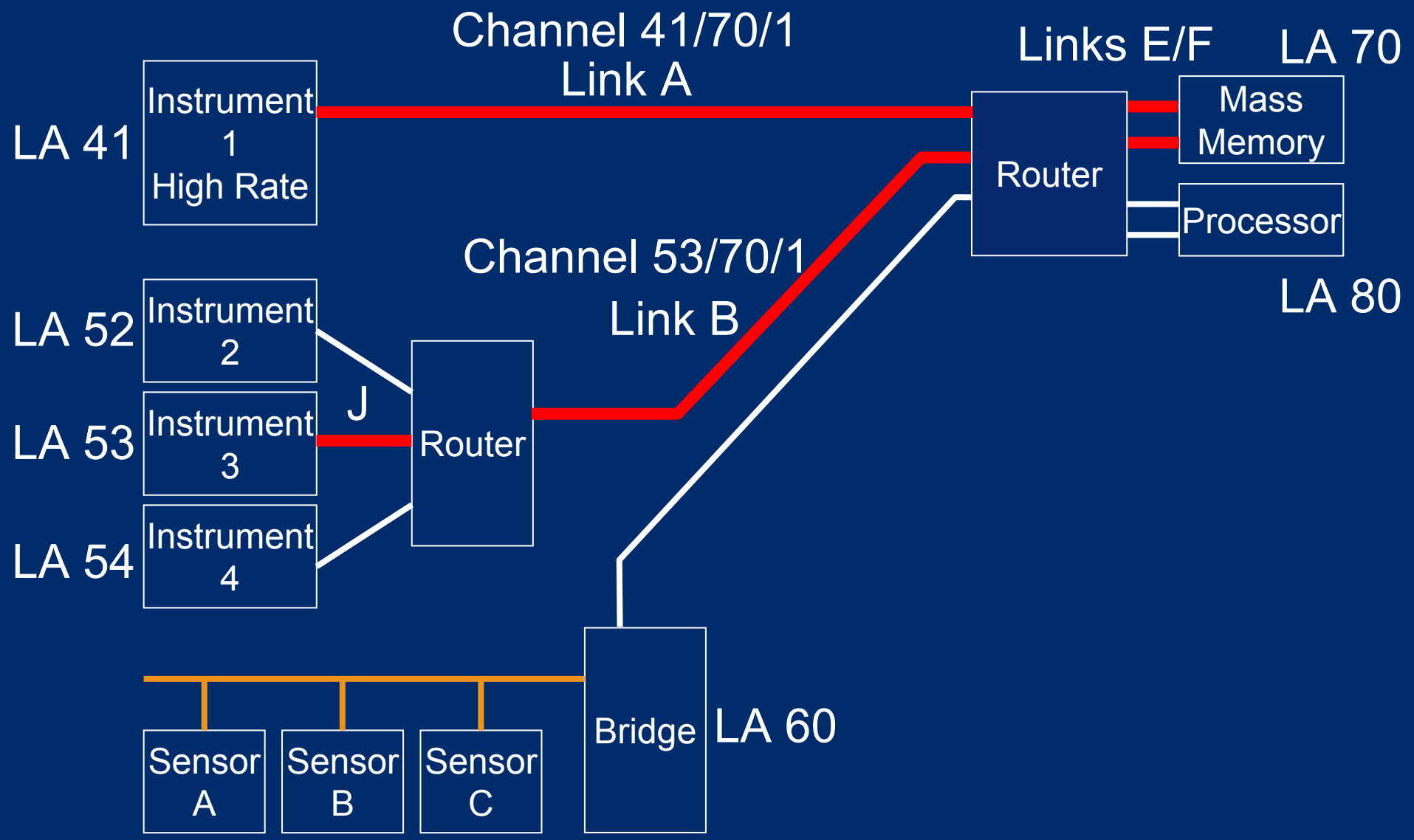
Example Allocating Time-Slots



Allocating Channels to Time-Slots

	Slot 0
41/70/1	A, E/F
52/70/1	I, B, E/F
53/70/1	
54/70/1	
60/60/1	
80/70/1	
80/xx/1	

Example Allocating Time-Slots



Allocating Channels to Time-Slots

	Slot 0	Slot 1
41/70/1	A, E/F	A, E/F
52/70/1	I, B, E/F	
53/70/1		J, B, E/F
54/70/1		
60/60/1		
80/70/1		
80/xx/1		

Allocating Channels to Time-Slots

	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7	Slot 8	…	Slot 63
41/70/1	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
52/70/1	I, B, E/F				I, B, E/F						
53/70/1		J, B, E/F									
54/70/1			K, B, G/H								
60/60/1				C, E/F							
80/70/1						g/h, E/F					
80/xx/1							E/F, g/h, a,b,c, I,j,k				

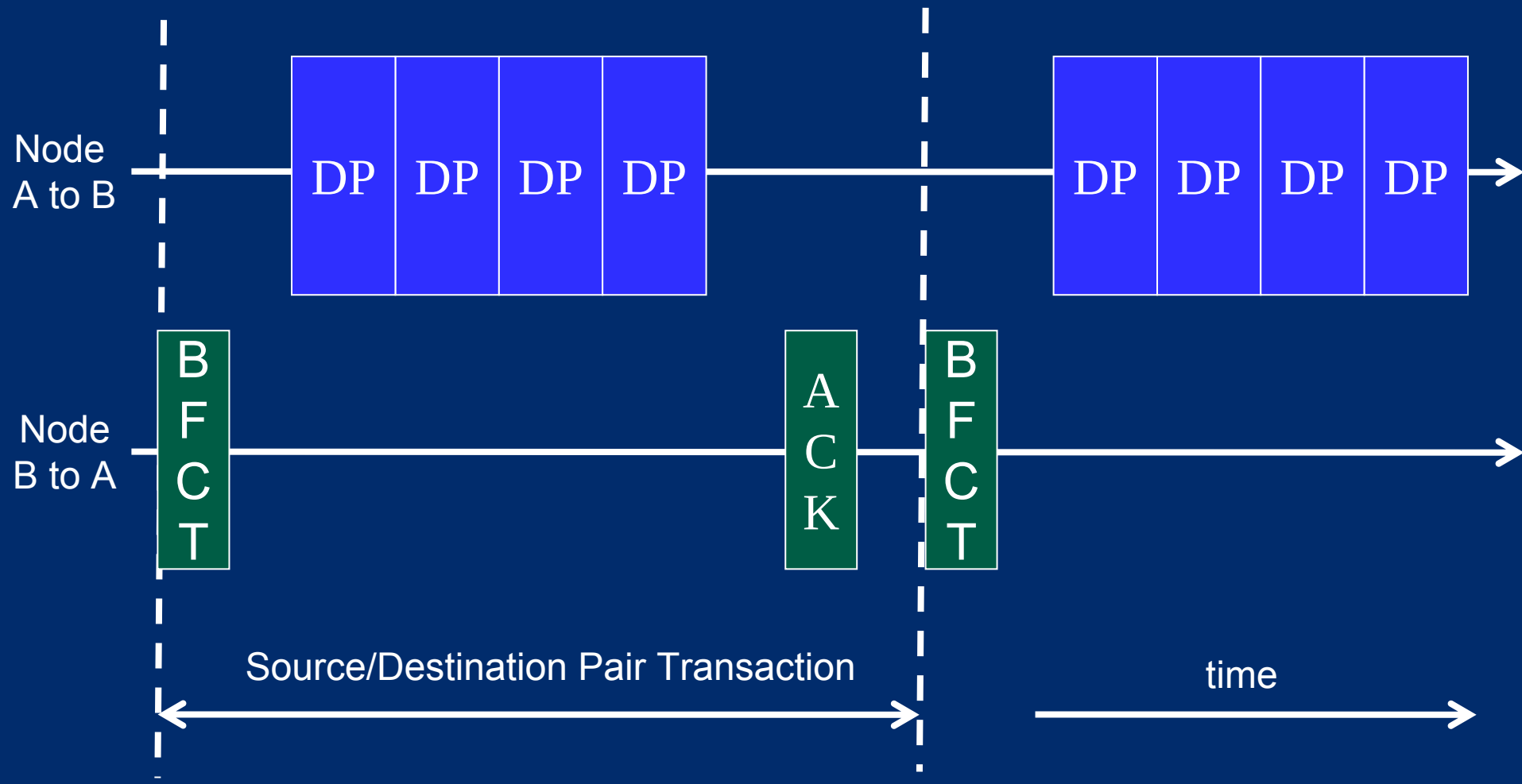
Sending and Receiving Data

- How to send and receive data without conflicts:
- Avoid sending data over conflicting paths
- 1. Buffer flow control
  - Do I have anything to send?
  - Is there room in the destination buffer?
  - If there is no room in the destination
    - avoid sending data PDU or it will block the network
- 2. Send the data
  - Send one or more data PDUs
- 3. Confirm that data arrived
  - Did the data PDUs arrive ok?

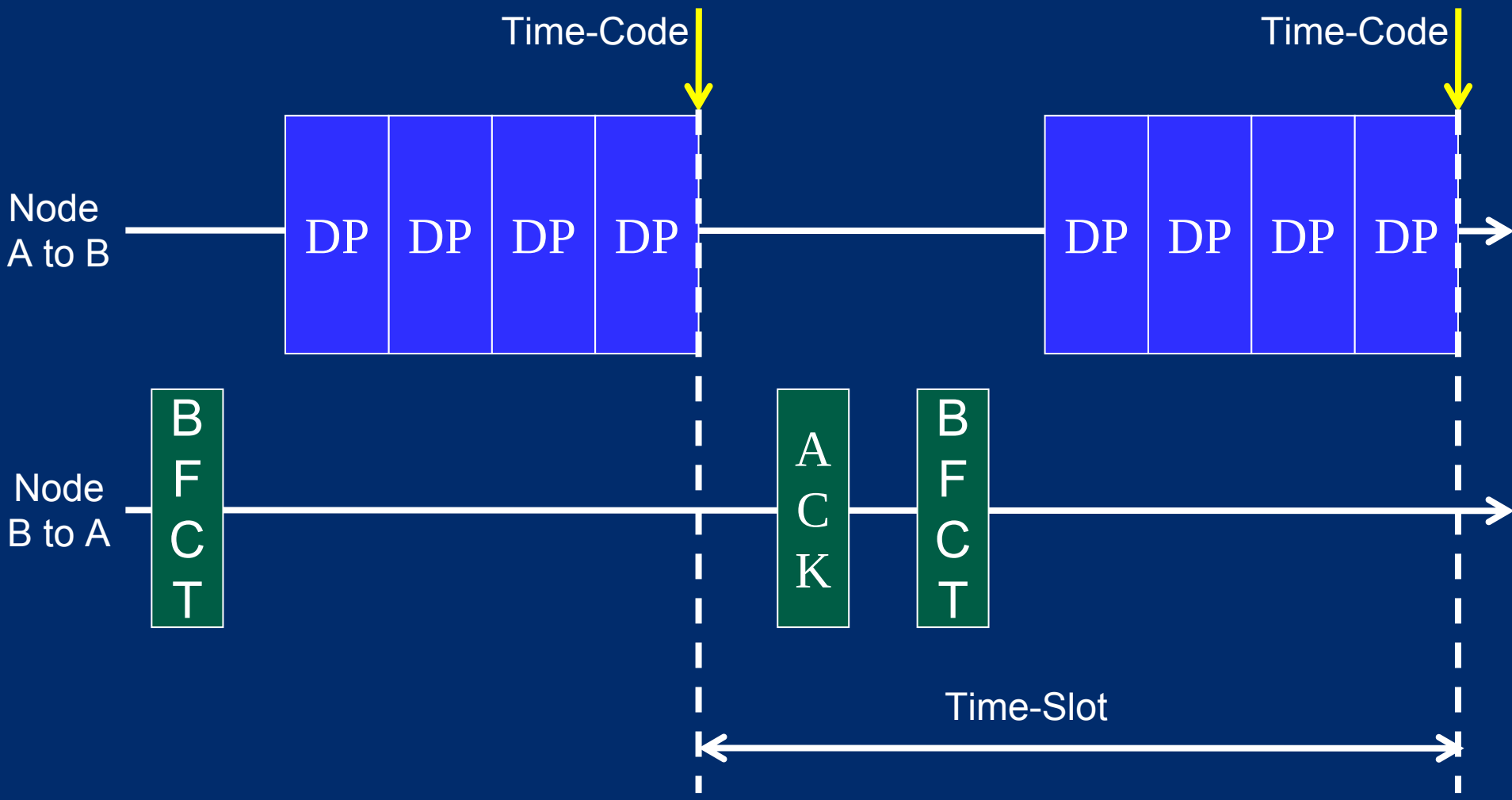
Resources for Flow-Control and ACKs

- Time-slots split into three parts:
  - Flow control phase
    - Which channels for this time-slot have room in destination channel buffer?
  - Data PDU transfer phase
    - Send data PDUs
    - For channels with room in destination channel buffer
  - Acknowledgment phase
    - Send acknowledgement of receipt of data PDUs
- This is the logical ordering

Transferring data from node A to node B



Time-Slot and Transaction



Scheduled Implementation

- When time-code received
  - Stop sending any more DPs
  - Wait
    - For network to become silent
  - Send ACKs for any DPs just received
  - Wait
    - For ACKs to propagate across network
  - Send Buffer Flow Control Tokens (BFCTs)
    - To indicate room in destination buffers
  - Wait
    - For BFCTs to propagate across network
  - Send any Data PDUs

Send BFCTs

- Look at schedule table entry for current time-slot
- If node is a destination of any channels that can send data in that time-slot
- Then check available space in destination buffers for those channels
- Send this information to source using a BFCT

Scheduled BFCT PDU Encapsulation

First octet sent

	Destination SpW Address	Destination SpW Address	Destination SpW Address
Destination Logical Address	SpW Protocol ID	Source Logical Address	Type = SBFCT
BFCT 0-1	BFCT 2-3	BFCT 4-5	BFCT 6-7
BFCT 8-9	BFCT 10-11	BFCT 12-13	BFCT 14-15
BFCT 16-17	BFCT 18-19	BFCT 20-21	BFCT 22-23
BFCT 24-25	BFCT 26-27	BFCT 28-29	BFCT 30-31
Header CRC	EOP		

Max 16-channels per source/destination pair

Last octet sent

Space Technology Centre
University of Dundee

Space Technology Centre
University of Dundee

Send Data PDUs

- Look at schedule table entry for current time-slot
- IF
  - Channel appears in current time-slot in schedule table
  - AND Channel has data to send
  - AND Channel has room in destination buffer
  - AND Channel is highest priority channel which fulfils above criteria
- THEN
  - Send one or more DPs from that channel

Scheduled Data PDU Encapsulation

First octet sent

	Destination SpW Address	Destination SpW Address	Destination SpW Address
Destination Logical Address	SpW Protocol ID	Source Logical Address	Type = DP
Channel	Sequence Number	Data Length	Header CRC
Data	Data	Data	Data
Data	Data	Data	Data
Data	Data	Data	Data
Data	Data	Data	...
Data CRC	EOP		

Last octet sent

Space Technology Centre
University of Dundee

Send ACKs

- Send acknowledgement to source
- Acknowledging all DPs that were received
- In previous time-slot
- From that source.

Scheduled ACK PDU Encapsulation

First octet sent

	Destination SpW Address	Destination SpW Address	Destination SpW Address
Destination Logical Address	SpW Protocol ID	Source Logical Address	Type = BACK
Channel Number	Sequence Number	Channel Number	Sequence Number
Channel Number	Sequence Number	Channel Number	Sequence Number
Channel Number	Sequence Number	Channel Number	Sequence Number
Header CRC	EOP		

Last octet sent

Space Technology Centre
University of Dundee

Alternative Scheduled ACK PDU

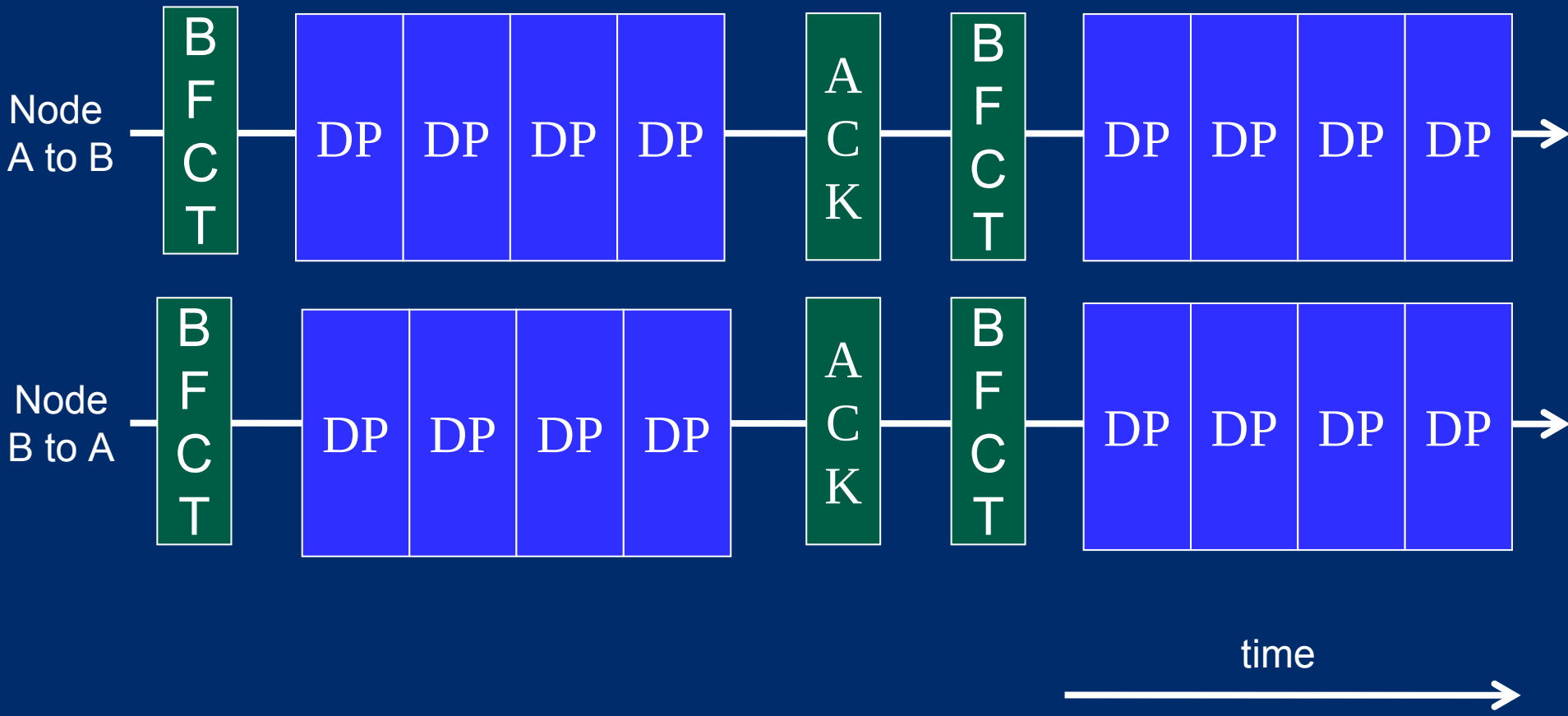
First octet sent

	Destination SpW Address	Destination SpW Address	Destination SpW Address
Destination Logical Address	SpW Protocol ID	Source Logical Address	Type = BACK
SEQ CH0,1	SEQ CH2,3	SEQ CH 4,5	SEQ CH 6,7
SEQ CH 8,9	SEQ CH 10,11	SEQ CH 12,13	SEQ CH 14,15
SEQ CH 16,17	SEQ CH 18,19	SEQ CH 20,21	SEQ CH 22,23
SEQ CH 24,25	SEQ CH 26,27	SEQ CH 28,29	SEQ CH 30,31
Header CRC	EOP		

Last octet sent

Space Technology Centre
University of Dundee

node A to node B and node B to node A



Space Technology Centre
University of Dundee

Fault detection

- Can use bi-directional transfer capability to check for failures
- I.e. At start of transaction
  - Expect to receive BFCT from other end of source/destination pair
- At end of transaction
  - Expect to receive ACK from other end of source/destination pair
- One node is checking operation of other node
- Extend to all node checking that they are only receiving from devices they are permitted to receive from

Space Technology Centre
University of Dundee

SpaceWire-T Architecture

Architecture

- User interface
  - Interface to users of SpaceWire-RT
- Segmentation
  - Chops SDUs into segments to send in Data PDUs (DPs)
  - Ensures that a large SDU does not hog the SpW network
- End to end flow control
  - Check destination buffer ready before sending packet
  - Ensures that DPs accepted immediately by destination to prevent blocking

Architecture

- Ack
  - Acknowledgement used to confirm receipt
- Address translation
  - SpW logical addresses used to identify nodes
  - Translates from logical address to path or logical address
- PDU Encapsulation
  - Wraps data PDUs with necessary header information
  - Encodes ACKs, BFCTs, etc

Architecture

- Resource reservation
  - Time-slots and scheduling traffic
  - Time-slots 200-300 μs to allow software and hardware implementation

User Application Interface

- Any SpaceWire application can run over SpW-T
- SpW-T uses same conceptual model as SpaceWire
- User interface is via source and destination channel buffers
  - User writes information into a source buffer
  - SpaceWire-T transfers this information across the SpaceWire network
  - User reads information from destination buffer
  - Source user is informed if delivery failed

User Application Interface

- User interface in source:
  - Source
    - logical address of source node
  - Destination
    - logical address of destination node
  - Channel number
    - Specific channel between source and destination
    - Defines QoS
  - Cargo
    - Information that is to be sent
  - Separator
    - Distinguishes one user data entity from next

User Application Interface

- User interface in destination:
  - Indication
    - New piece of user information has started to arrive
  - Source
    - logical address of source node
  - Destination
    - logical address of destination node
  - Channel number
    - Specific channel between source and destination
    - Defines QoS
  - Cargo
    - Information that has been received
  - Separator
    - Distinguishes one user data entity from next

Segmentation

- Segmentation allows multiple, large, user data entities to be transferred at the same time - interleaving
- Each SpW-T data PDU can carry up to 256 bytes of user information
- Large user data entities have to be slit up into 256 byte segments
- Split into one or more data PDUs containing 256 bytes user information
- Followed by zero or one data PDUs containing fewer than 256 bytes user information

Address Translation

- Nodes identified using SpaceWire logical addresses.
  - A unit can have more than one SpaceWire logical address
  - Only one node can have a specific logical address
- Up to 223 logical addresses
  - Sufficient for most spacecraft applications
- Routing can be done with path and/or logical addressing
- Node identification done with logical addresses

Address Translation Table

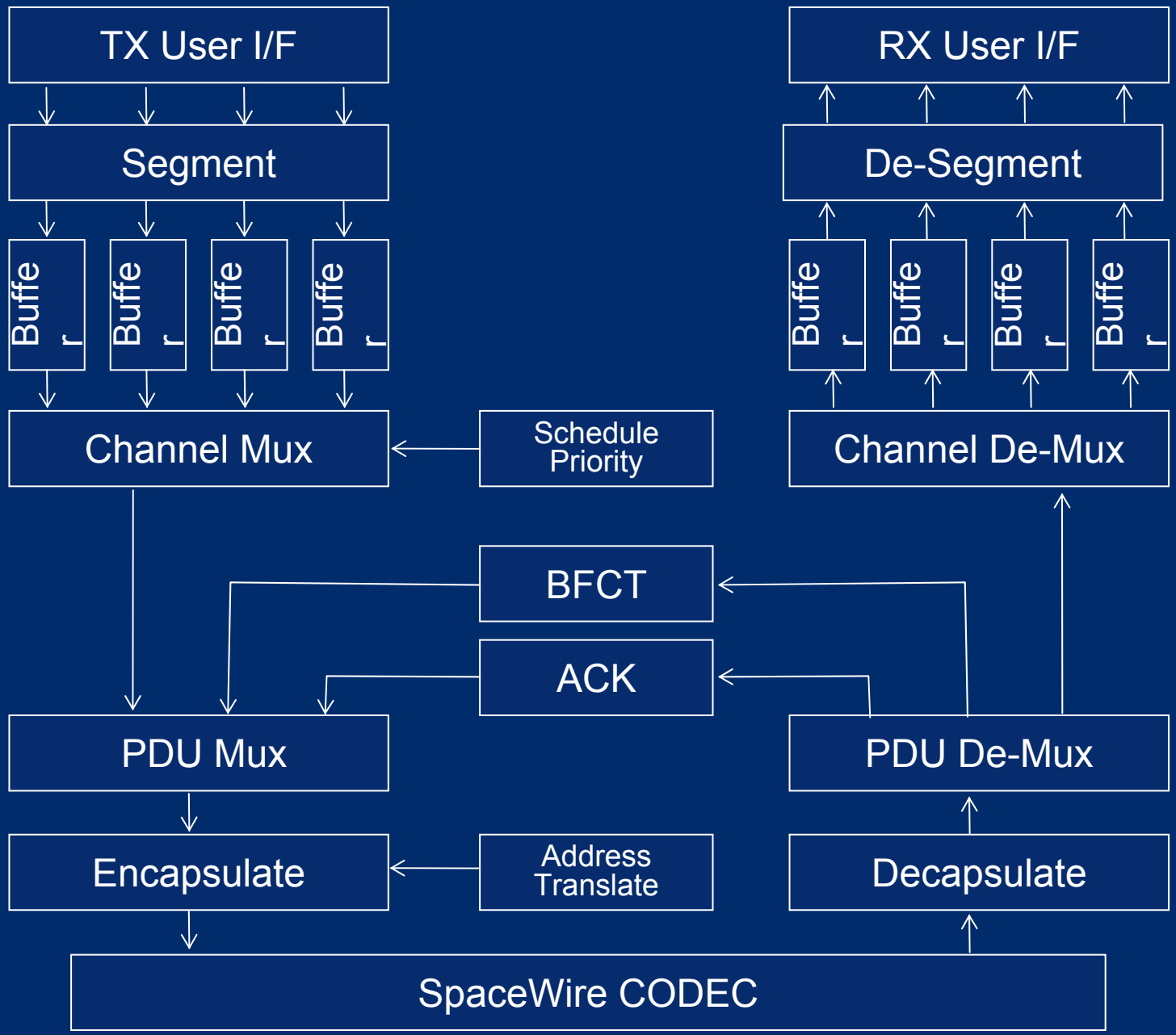
SpaceWire-T Logical Address	SpaceWire Address
120	120
124	1, 6, 5, 2, 124
150	1, 150

Space Technology Centre
University of Dundee

Address translation

- Address tables for ACK, BFCT
  - Accessed via SLA, DLA, Ch#
  - Held in each node
  - May require updating if network changes
- Multiple network configuration regimes may be incorporated in the table
  - To allow rapid re-organisation of channel paths
  - Depending on network state

Architecture Diagram



Space Technology Centre
University of Dundee

Space
Technology
Centre
University of Dundee

SpaceWire-T
Questions on Requirements

Some questions about SpW-T

- What to do if a DP goes missing?
- Do we simply stop and wait for the system level FDIR to investigate?
- If it is part of an SDU
  - Discard remainder of SDU?
  - Inform source of error?
  - Inform destination of error?
  - Deliver part of SDU to destination application?
- What about an ACK going missing?
  - Could mean rest of SDU is thrown away...
- How many channels per source/destination pair?

SpaceWire-T

- Is retry required?
  - CCSDS say yes i.e. is in SOIS specifications
- Complexity
  - Once there is an ACK the additional complexity to retry is very small
- Have to implement retry in every application
- Time-outs etc are network dependent
  - Complexity in managing time-outs

Space Technology Centre
University of Dundee

SpaceWire-RT Prototype

Prototype Performance

- Concentrated on scheduled network
- Includes retry and redundancy
- Aimed to assess performance of software implementation
  - Two AT7913E RTCs taking to each other
  - Over AT7910E SpW-10X based network
  - Existing radiation tolerant SpaceWire components
- Initial performance assessment:
  - 200 Mbits/s links
  - 300 µs time-slot
  - 6 DPs of 256 bytes each sent per time-slot
  - Actual data rate of 40 Mbits/s

Prototype Performance

- Expect to be able to improve software performance
  - Goal 200 µs time-slot
- Compatible with hardware implementation
  - 200 µs time-slot
  - 12 DPs of 256 bytes each per time-slot
  - 123 Mbits/s data rate
- Initial prototype implementation
  - To check likely performance of software solution

Fault tolerance

- Retry
  - Recovers from errors
  - With minimum buffer requirements (DP size)
- Redundancy
  - Alternative paths can be invoked automatically

Space Technology Centre
University of Dundee

SpaceWire-RT vs SpaceWire-T

Higher Level FDIR

- Providing FDIR at application level is complex
- Adds complexity to the system
  - Because of extra information that needs to be passed between network and application
  - E.g. Time-outs are dependent upon network
- Adds complexity to the application
  - Each application has to provide its own retry mechanism
- Information has to be buffered in larger units
  - SDUs instead of PDUs
- Requires more memory

SpW-T using SpW-RT

- SpW-T can be implemented with SpW-RT
- Try once on prime path and report if this fails.
  - Number of attempts on prime path = 1
  - Number of attempts on redundant path = 0
  - Autonomous reconfiguration = disabled
  - Simultaneous retry = off
- Address table has prime paths only

SpaceWire Logical Address	Prime SpaceWire Address	Redundant SpaceWire Address
120	120	-
124	1, 6, 5, 2, 124	-
150	1, 132	-

Recommendation

- SpaceWire-RT
- Specify implementation option
- Without retry and redundancy switching
- Add ACK to Best Effort service
- Remove Basic service

- Meets all requirements
- Full CCSDS SOIS compliance
- SpW-T and simplified validation
- SpW-RT and PnP/ORS capabilities

Space
Technology
Centre
University of Dundee

Space Technology Centre
University of Dundee

Spare Slides

When a DP arrives

- If there is room in the destination buffer
  - Put DP in destination buffer
  - Update received sequence number for that channel

When an BFCT arrives

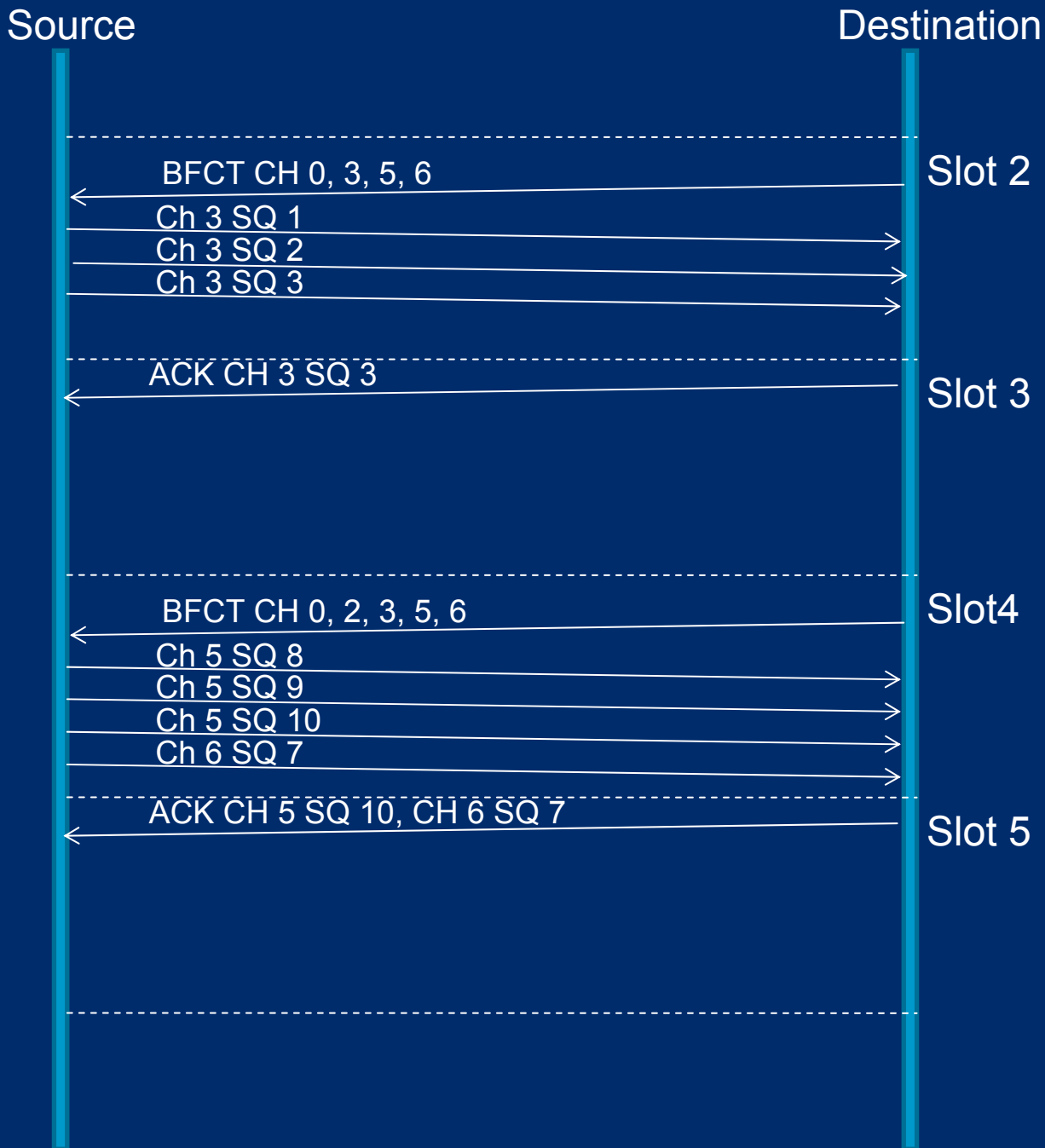
- On receipt of a BFCT
  - Extract destination buffer status information
  - Update buffer credit counters in source

When an ACK arrives

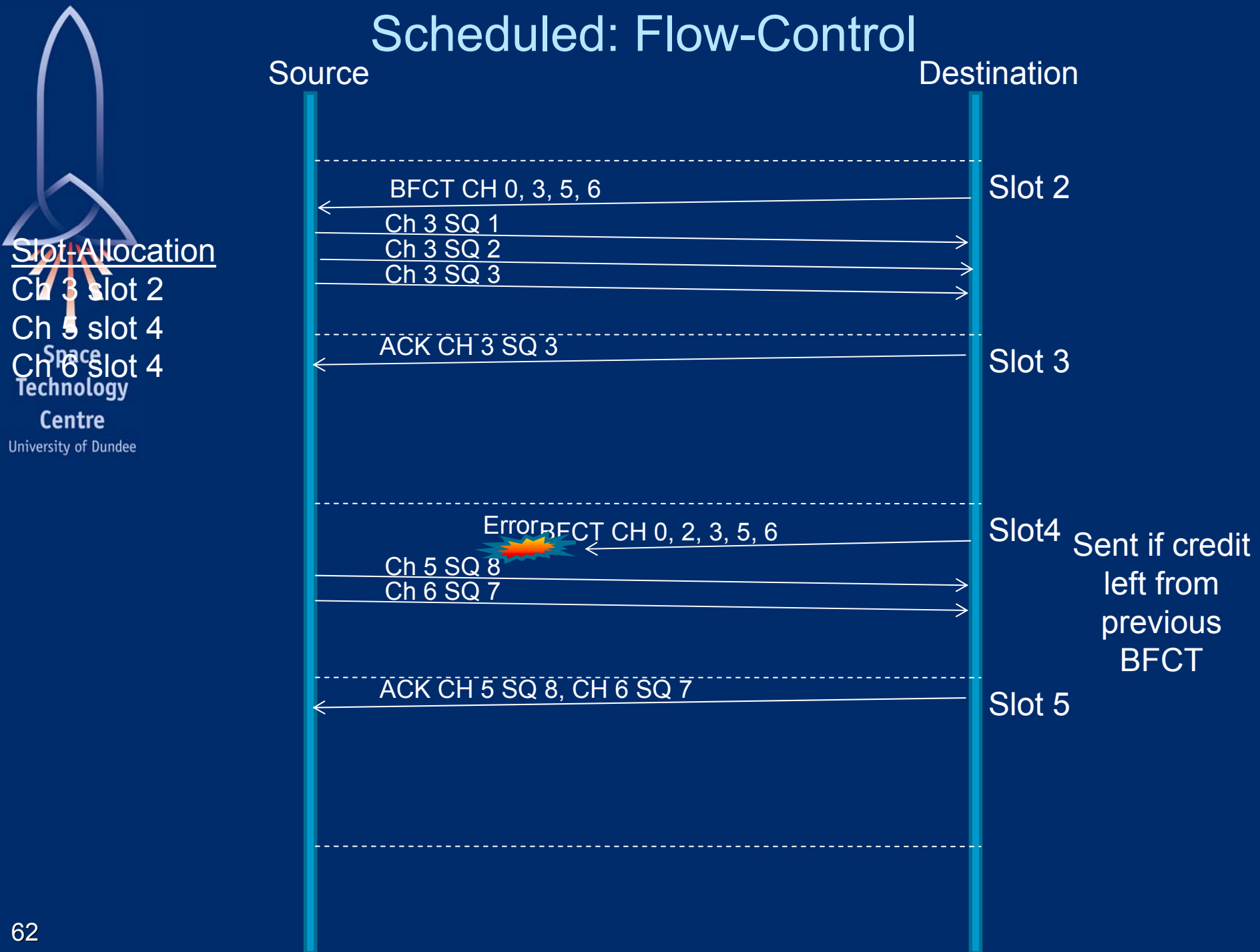
- On receipt of an ACK
  - Extract ACK information for each channel
  - If sequence number is equal to that of last DP sent
  - Then data received OK
  - Else restart the channel

Scheduled: Flow-Control

Slot Allocation
Ch 3 slot 2
Ch 5 slot 4
Ch 6 slot 4



Scheduled: Flow-Control

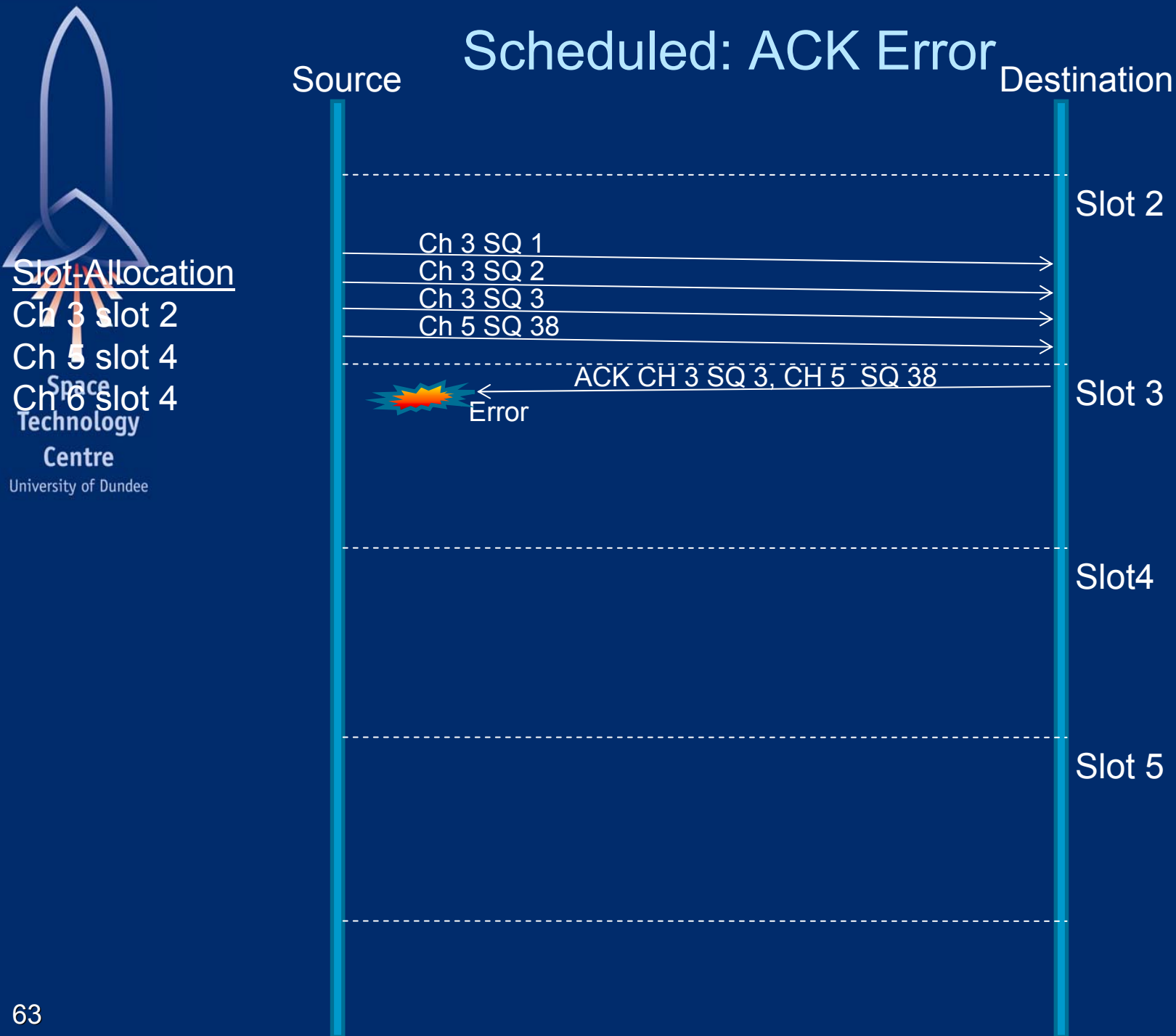


Slot Allocation
Ch 3 slot 2
Ch 5 slot 4
Ch 6 slot 4

Sent if credit left from previous BFCT

Space Technology Centre
University of Dundee

Scheduled: ACK Error



Slot Allocation

Ch 3 slot 2

Ch 5 slot 4

Ch 6 slot 4

Space Technology Centre
University of Dundee

Space Technology Centre
University of Dundee

Example Onboard Data-Handling Architecture