Avoiding SpaceWire Network Congestion

Martin Suess, Albert Ferrer
SpaceWire 2011 San Antonio TX
09/11/2011

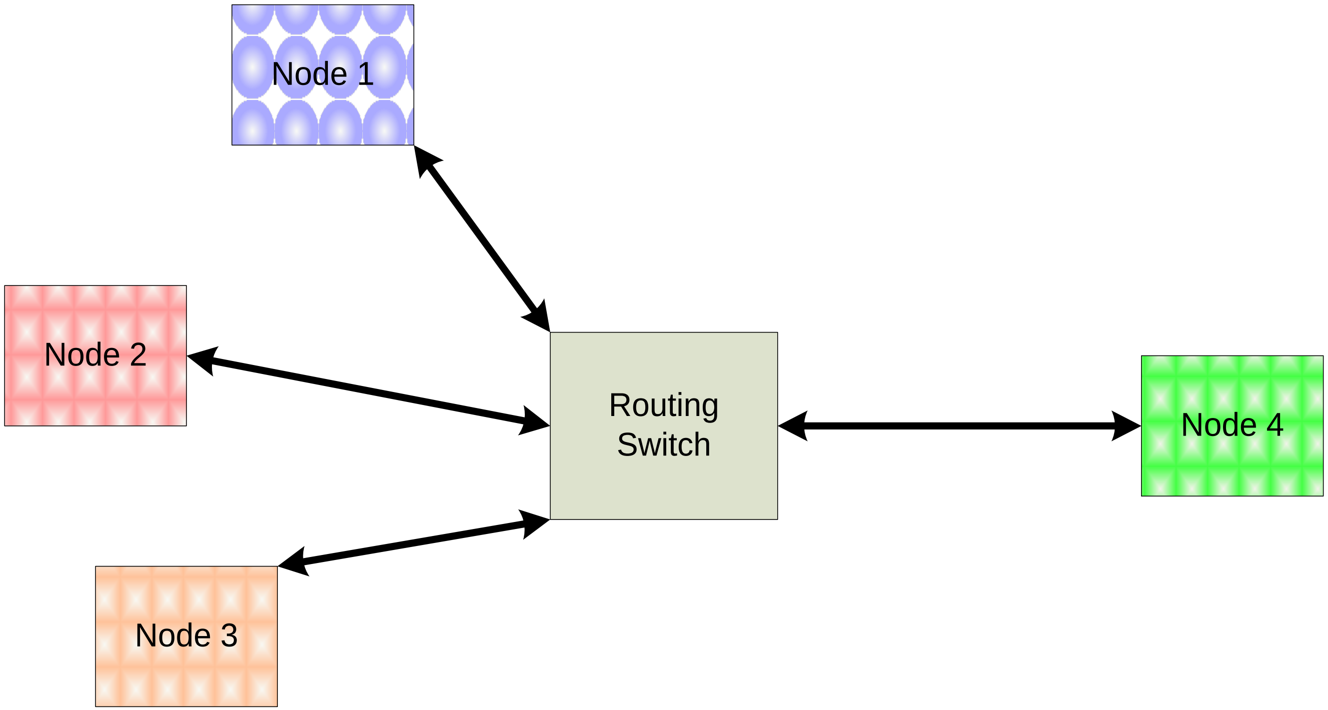
European Space Agency

Overview

- Simple network example
- Impacts of the SpaceWire switching scheme for network throughput
- Relation between maximum latency and node buffer requirement
- Node parameters that need to to be specified
- Conclusion

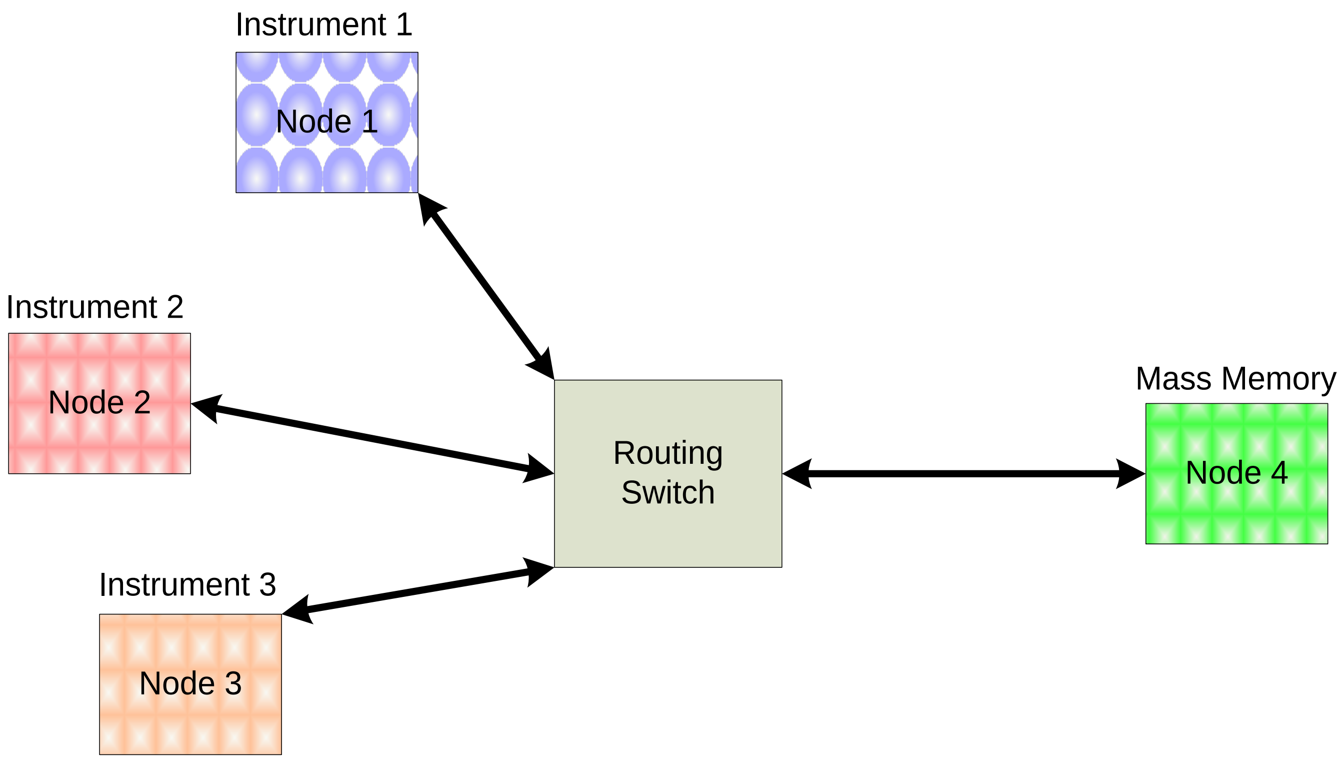
Simple SpaceWire Network Example

- Network with 4 nodes and one routing switch
- The routing switch uses round robin arbitration



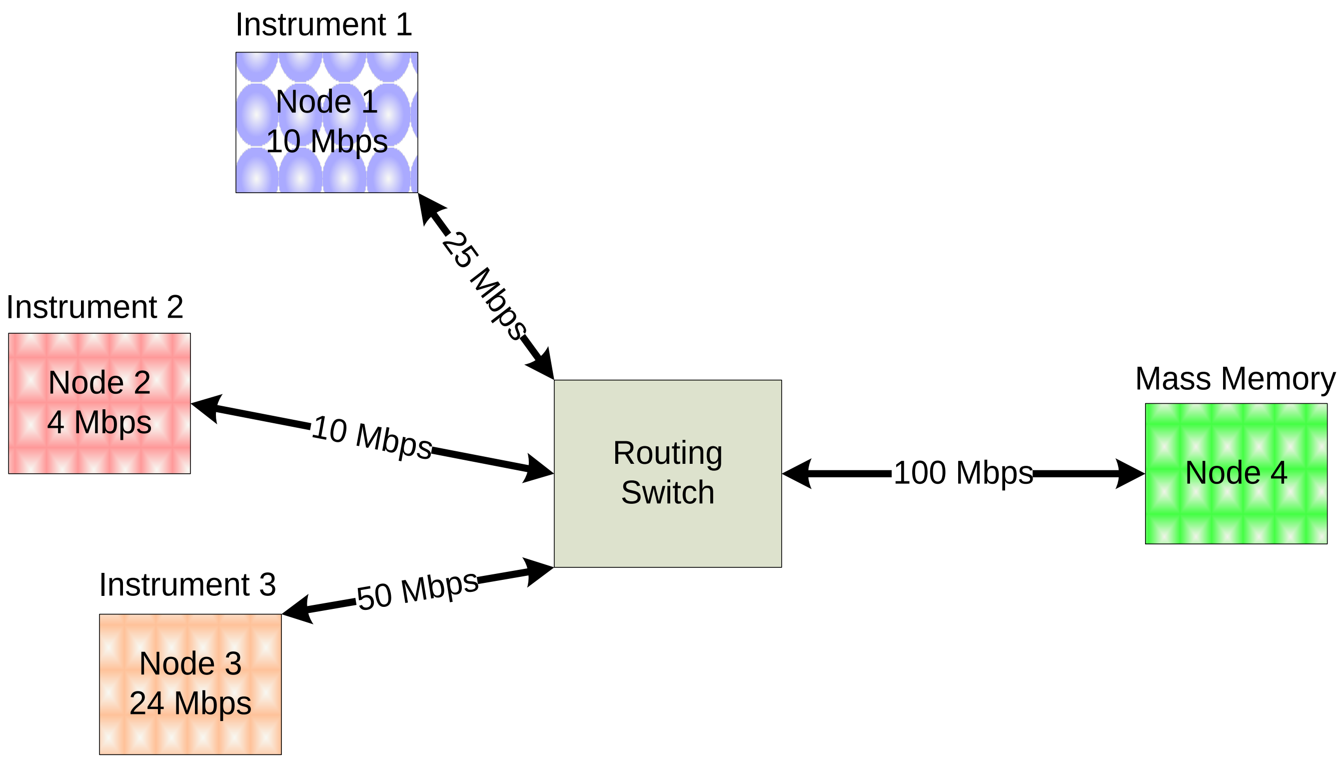
Simple SpaceWire Network Example

- Node 1 to 3 are instruments need to transfer the payload data generated to the mass memory in node 4



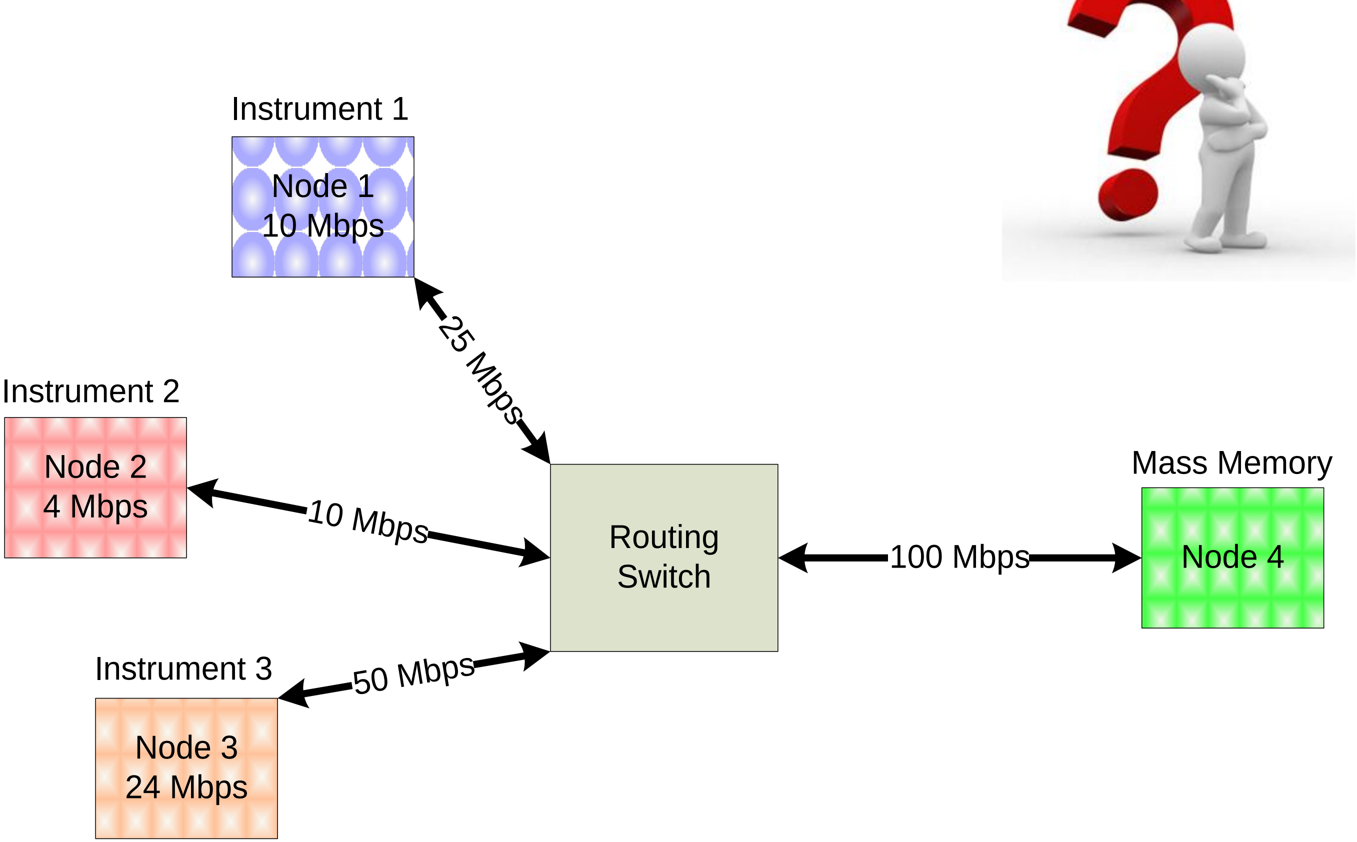
Simple SpaceWire Network Example

- The instruments generate their data at a **fixed rate**
- The link data rates are set to the following data rates



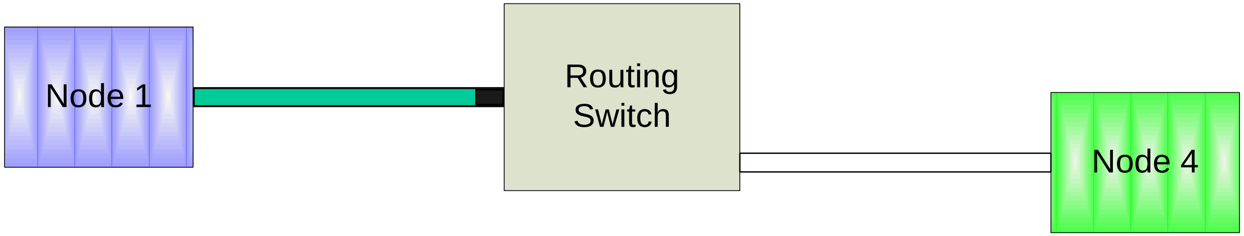
Simple SpaceWire Network Example

Will it work ?!



Wormhole Switching

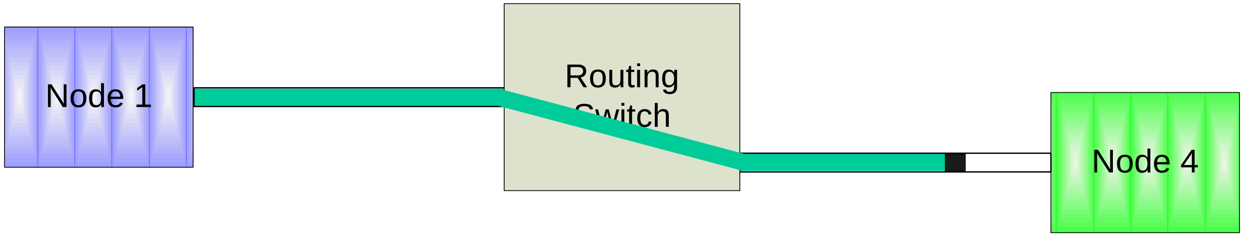
- SpaceWire routing switches are specified to use Wormhole Switching
- This is keeping the switches simple, low power and requires little memory
- Together with link flow control it ensures that no packets are dropped by the network
- It has been developed in the context of Transputer Systems and today often used in Network-on-Chip



The routing switch receives the packet and checks availability of the requested output port

Wormhole Switching

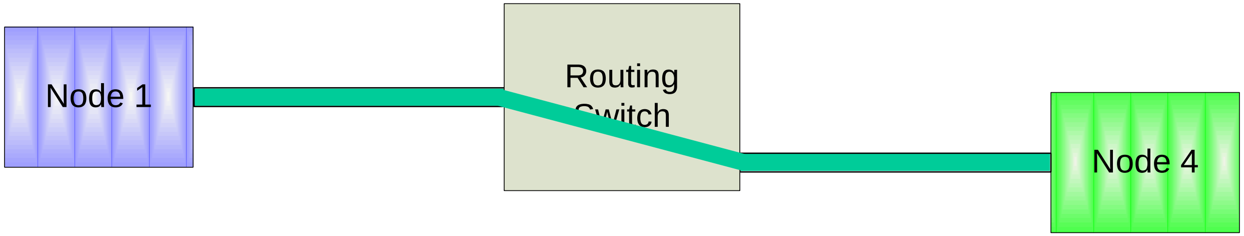
- SpaceWire routing switches are specified to use Wormhole Switching
- This is keeping the switches simple, low power and requires little memory
- Together with link flow control it ensures that no packets are dropped by the network
- It has been developed in the context of Transputer Systems and today often used in Network-on-Chip



The routing switch connects input and output port and let the packet flows through

Wormhole Switching

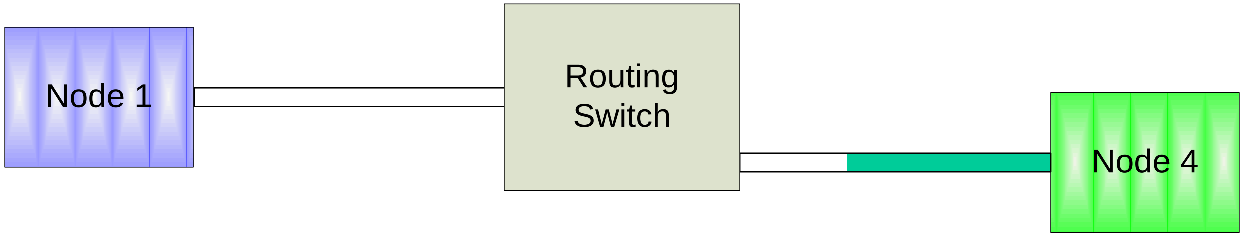
- SpaceWire routing switches are specified to use Wormhole Switching
- This is keeping the switches simple, low power and requires little memory
- Together with link flow control it ensures that no packets are dropped by the network
- It has been developed in the context of Transputer Systems and today often used in Network-on-Chip



While the packet is transferred the complete end-to-end path is occupied

Wormhole Switching

- SpaceWire routing switches are specified to use Wormhole Switching
- This is keeping the switches simple, low power and requires little memory
- Together with link flow control it ensures that no packets are dropped by the network
- It has been developed in the context of Transputer Systems and today often used in Network-on-Chip



Only after the EOP has passed the occupied link is accessible for other packets again

Link Occupation Duty Cycle

- The continuous data stream from the instruments is segmented at the at the source node
- A link occupation duty cycle can be defined as:

$$link\_occupation\_duty\_cycle = \frac{average\_source\_data\_rate}{achievable\_link\_bandwidth}$$

- The *average_source_data_rate* corresponds to instrument data rate
- The *achievable_link_bandwidth* during transmission is determined by the slowest link on the data path

Note: Due to the parity and the data-control flag used per 8 data bits only 80% of the gross data rate is available for data

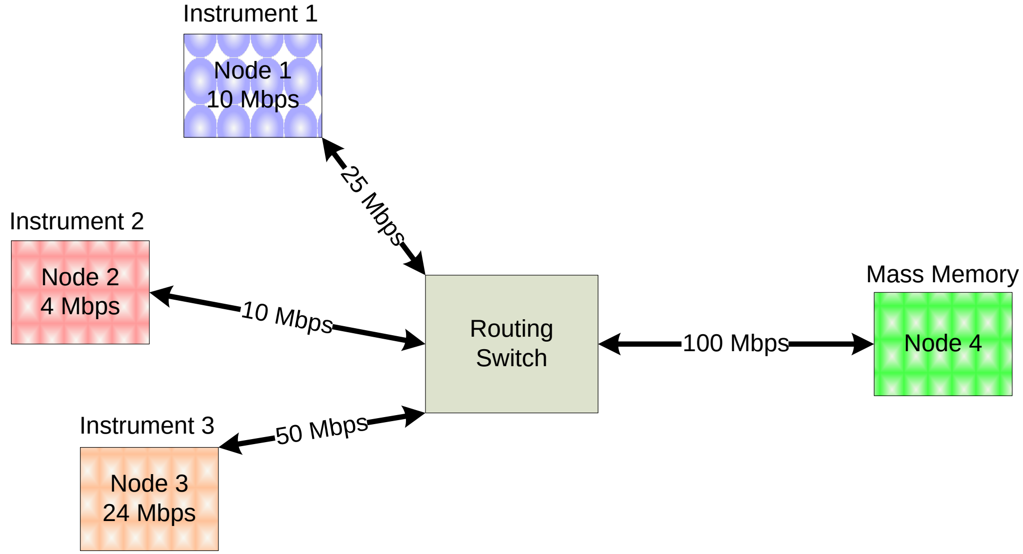
Link Occupation Duty Cycle

– For the simple example given before the link occupation duty cycle is:

– Node 1: $$\frac{10Mbps}{0.8\cdot 25Mbps} = 50\%$$

– Node 2: $$\frac{4Mbps}{0.8\cdot 10Mbps} = 50\%$$

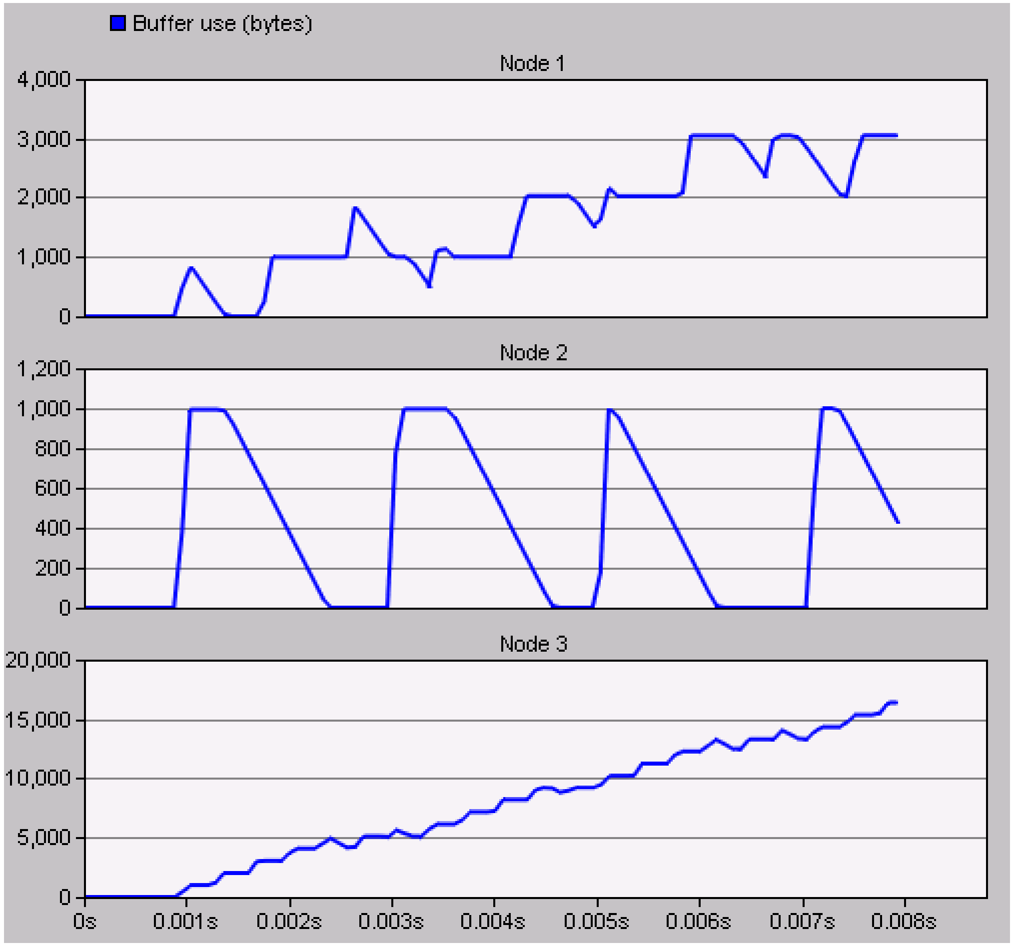
– Node 3: $$\frac{24Mbps}{0.8\cdot 50Mbps} = 60\%$$



– The link to node 4 is already fully occupied the traffic from node 1+2
– Including the traffic from node 3 it would reach 160%
– The described simple network shows excessive congestion and is not stable

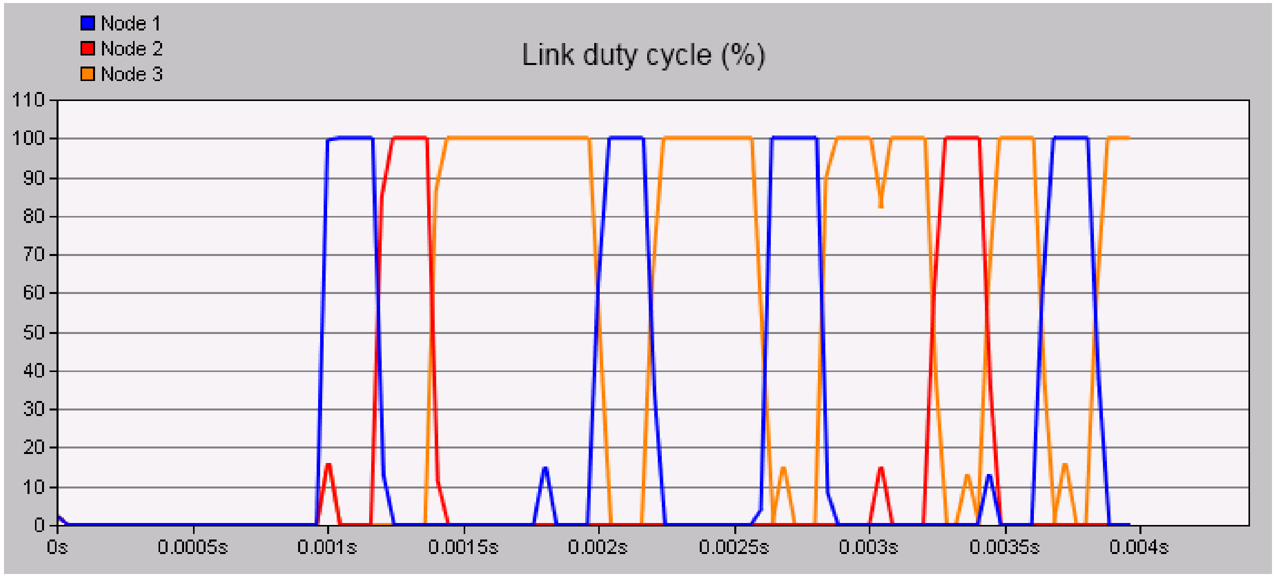
Excessive Network Congestion

- The excessive Network congestion is illustrated with an OpNet simulation
- Segment size equals 1k
- For nodes 1 and 3 the effective data rate with congestion is less than instrument data rate
- This would result in a local buffer overflow



Solution for the simple example

- The resulting link occupation duty cycle must be kept below 100%
- One solution is to run all links at 50Mbps
- The link duty cycles of nodes 1 to 3 result in: 25%+10%+60%=95%
- One other solution could be to run the links from node 2 and 3 at 100Mbps: 50%+5%+24%=79%
- Such a network configuration is called stable



Achievable Bandwidth

- Is it sufficient to select the correct SpaceWire link speeds?
- Internal properties of the transmitting or receiving node can further reduce the *achievable_link_bandwidth*
- This limitation can be e.g. due to
  - Lower node internal bandwidth
  - Arbitration of hardware resources
  - Software schedule
  - Interrupt latency
- The *achievable_link_bandwidth* is determined by the slowest element in the data path including the transmit and the receive node

Node Data Buffer

- Also a network in stable configuration requires sufficient buffer space
- Node buffer overflow results in data loss
- In the instrument source node the data first have to be buffered and segmented
- Once a whole data segment is buffered it can be transmitted
- The required buffer size depends on the product of the average source data rate and maximum latency

Maximum Latency

– An upper bound for the latency of each data flow can be calculated

$$D(flow,hop) = T(flow,hop) + \sum_{\substack{h \in \text{hops} \\ \text{left}}} \sum_{\substack{p \in \text{input} \\ \text{ports}}} \max\left[D(f,h)\right]\Big|_{\forall f \text{ using } p \text{ and } (h+1)}$$

Latency without congestion

Latency due to congestion produced by other flows

– The latency is the sum of:
  – Time required for segment transfer
  – Latency due to network congestion
– The upper bound of the latency depends on:
  – Segment lengths
  – Target node and switch latency
  – Network topology
  – Interfering data flows
  – Effective link speed

Node Level Requirements

- In addition to the link speed a number of node parameters have to be specified.
- For the source node:
  - Minimum effective data rate with which a packet can be sent out on the network
  - Maximum average data rate the node is allowed to send out
  - Segment or packet size
  - Minimum source buffer size
- For the target node:
  - Segment or packet size,
  - Minimum effective data rate with which a single segment or packet can be received from the network,
  - Maximum delay time before packet is received at the effective data rate
  - Minimum average receive data rate that can be sustained which may sometimes be less than the effective receive data rate

Requirement Verification

- For more complex networks and flows the set of requirements can be obtained by simulation
- They need to be imposed on the units already at Phase B
- The requirements should be tested for each node at sub-system level
- This approach ensures the correct interoperation of all nodes and data flows once integrated at system level

Conclusions

- SpaceWire is a wormhole switch based network
- Utilising different link speeds in the same network has to be used with great care
- It is not enough to specify the SpaceWire link speed for a unit
- The data flows present in the on-board network are typically well defined
- There exist an upper bound for the maximum latency
- The buffers to cope with the network latency are located inside the transmitting node
- Sufficient transmit buffer size is required to prevent data loss