SpaceWire RT 2.1

4Links®

Although the principle of allocating time slots appears simple, and has been used elsewhere, its application to SpaceWire appears to be getting very complex (and, therefore, increasingly hard to validate).

- Is SpaceWire fundamentally different from other networks?
  - Because it does *not* behave like a bus or …?
- Is there a fundamental conflict between requirements
  - e.g. guaranteed delivery (with retries) and timely delivery (limiting the number of retries)?

SpaceWire RT 2.1

4Links

Despite the ever-increasing size of the document, there are still many issues regarding a practical implementation.

- We do not yet understand the proposal well enough to provide support.

SpaceWire RT 2.1

4Links®

How robust is the proposal?

What is the behaviour under abnormal conditions

- What happens if a destination node fails to accept data fast enough and a packet lasts into another time slot?
- What happens if a node sends to much data (jabbering node)– or takes too long sending it?

SpaceWire RT 2.1

4Links

Backward compatibility?

- Nodes
  - It appears that *all* nodes must be changed to meet the RT requirements, even if they have no RT behaviour.
  - Can devices such as SMCS332 be used without a CPU – e.g. as FIFO interfaces etc.
- Router
  - Support for a second time code (if needed)?
  - Access control to cut-off over-long (size or time) packets at the end of a time slot?

SpaceWire RT 2.1

4Links®

Simplification?

Much of the complexity appears to originate with extensive Quality of Service provision

- Should some of the QoS be left to higher layers?
  - e.g. guaranteed delivery