SpaceWire-RT Protocol Definition Comments

Prof. Yuriy Sheynin

St. Petersburg State University of Aerospace Instrumentation

E-mail: sheynin@online.ru

SpaceWire-RT

Introduces a set of useful features that complement the existing functionality of the SpaceWire standard,
of SpaceWire networks as a communication entity
(for some applications)

- E2E flow control
- In-order delivery
- Ensured delivery of data
- Regular features to support communication paths redundancy for fault-tolerant communications

+

- Adds QoS features,
as much, as they could be implemented over a standard SpaceWire interconnection

Timeliness

SpaceWire –RT provides

- guaranteed *throughput*,
- but <u>not</u> guaranteed *latency*

Good (high) ***throughput*** doesn’t guarantee good (low) ***latency*** of the particular packet delivery

For Real-Time systems just the Latency could be important

Time-slot margins

“Time-slots are distributed in SpaceWire using SpaceWire time-codes” [p.42]

- Skew and Jitter of time slot marks (time-codes) distribution
- Skew in local timers
- Larger margins for time-slots could be required

Managing routers

“SpaceWire routers need not be managed by the resource reservation mechanism” [p.39]

Don’t think so.

Routing tables in routers actually correlate with the specified SpaceWire-RT mechanisms and should be managed.

Timeouts

“Wait a bit for acknowledgements to propagate…”

Timeouts should be specified.
And it wouldn’t be an easy questions…

Link opposite directions transfer interference

“Link resources (one direction) do not conflict with uppercase link resources (other direction).” [p.43]

It isn’t exactly so.

The link-level FCT symbols are running in the opposite direction and consume some of its throughput (4 bits per 64 bits = 1/16)

Add it to the Resource Reservation margin

Priority and mixed scheduling algorithms

“Scheduling can be combined with priority so that several channels all to the same destination and using the same network resources for communication but with different priority levels are mapped on to time-slot.
When the time-slot arrives, the highest priority channel (the one with the lowest channel number) which has data to send and room in its destination buffer, is sent first.” [p.66]

It is a **PPS** (*Priority Processor Sharing*)scheduling algorithm

It gives low latency for high priority packets,
but can't guarantee that there wouldn’t be interference, cross influence between the channels

The **GPS-style** (*Generalized Processor Sharing*) scheduling could be better to specify for such mode of operation

Generalized Processor Sharing

For every channel $\boldsymbol{i} \in \mathbf{[1,N]}$

a minimal $\boldsymbol{r_i}$ throughput is specified,

$\boldsymbol{r}$ is the link throughput

$$\sum_{i=1}^{N} r_i \leq r$$

$$g_i(t) = r \cdot \frac{r_i}{\sum_{j \in B(t)} r_j},$$