18th SpW WG meeting

Guaranteeing real-time property of SpW network based on SpW-D

24-Apr.-2012

Hiroaki TAKADA

Executive Director and Professor
Center for Embedded Computing Systems, Nagoya Univ.

Chairman, TOPPERS Project

Email: hiro@ertl.jp URL: http://www.ertl.jp/~hiro/

Self Introduction – Hiroaki Takada

Current Positions

- Professor, Nagoya University
- Executive Director, Center for Embedded Computing Systems (NCES), Nagoya University
- Chairman, TOPPERS Project

  and several others

Major Research Topics

- real-time operating systems for embedded systems
- real-time scheduling and analysis
- electronic system-level design of embedded systems
- automotive embedded systems and network
- energy optimization for embedded systems
- functional safety

TOPPERS Project

- Develop various open-source software for embedded systems including RTOS and promote their use.
- Incorporated as a non-profit organization in 2003.

Building a widely used open-source OS as Linux in the area of embedded systems!

Applications of TOPPERS OS

- TOPPERS OS is widely used in the industries.
  - consumer applications
  - industrial applications
  - automotive systems
- TOPPERS OS is planned to be used in some space systems.

ASTRO-H (JAXA)
under development

H−IIB（JAXA）
under development

Table of Contents

Joint Project of JAXA and NCES

- JAXA and NCES are conducting a joint project for spreading the use of SpW to wider range of applications.

Goals of the Project

- to propose a flexible methodology for guaranteeing the real-time property of SpW network based on SpW-D
- to develop a software platform (real-time kernel + SpW middleware) supporting the proposed methodology

Study Group on SpW Real-Time Property

- ISAS and NCES host a study group with Japan SpW User Group (including NEC, MELCO, and MHI) for:
  - gathering requirements on the SpW real-time property guaranteeing methodology from industries,
  - investigating the experience and problems obtained from the development of ASTRO-H, and
  - collecting opinions on the proposed methodology.

Approach of our Investigation

Starting Point

- SpW-D standard under discussion
  - [1] P. Armbruster: SpW-D Overview and Trade-offs, 16th SpW WG meeting.
  - [2] S. Parkes: SpW-D Proposed Contents of Standard, 17th SpW WG meeting.
- ASTRO-H SpW Network Design Criteria

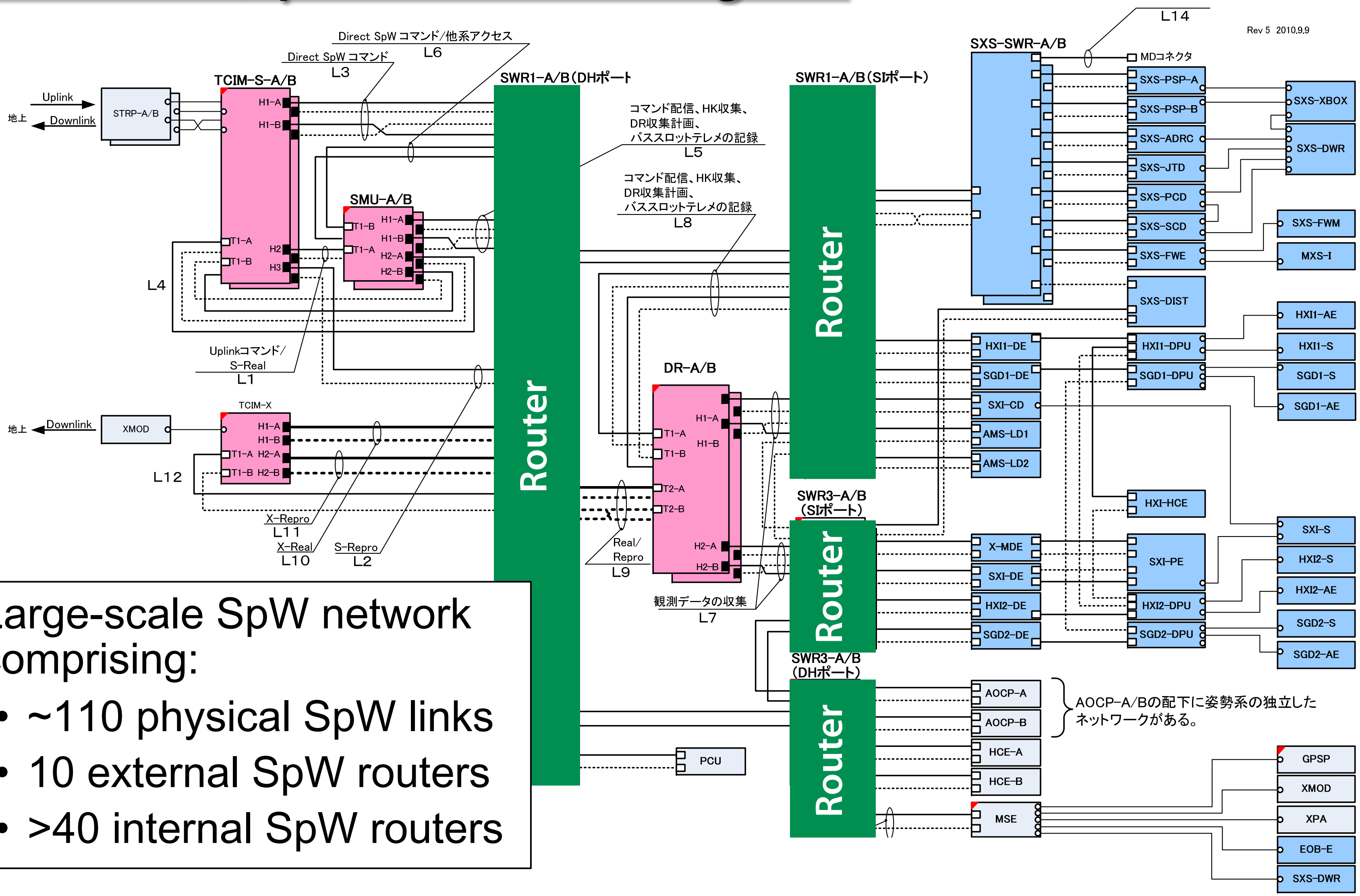
Gathering Additional Requirements

- from space system industries
- from the experience and problems obtained from the development of ASTRO-H

Some Concepts/Functions Added to SpW-D

- Some additional concepts/functions are added to SpW-D for satisfying the additional requirements.

ASTRO-H SpW Network Diagram



Large-scale SpW network comprising:

- ~110 physical SpW links
- 10 external SpW routers
- >40 internal SpW routers

Additional Requirements

Upper Layer Protocol

- Any upper layer protocol on SpW should be supported.
  - The upper layer protocol is responsible for the packet loss detection/recovery, and for packet segmentation/desegmentation (when necessary).

Assumption on RMAP Implementation

- High throughput should be obtained even when RMAP target nodes are implemented with software.

More Extensive FDIR Mechanisms

- Additional FDIR mechanisms need to be specified for handling some failures that are not fully covered by the current SpW/SpW-D FDIR mechanisms.
- Incorporated as optional features because existing SpW devices cannot support them.

Concepts/Functions Added to SpW-D

Subnet Concept and Flexible Time-Slotting

- A SpW network can be divided into some subnets.
- The length and the number of time-slots can be different for different subnets.

Mode Concept

- Subnet and schedule can be changed for each mode.

Split RMAP Transaction

- RMAP reply packet transferred in different time-slot should be allowed.

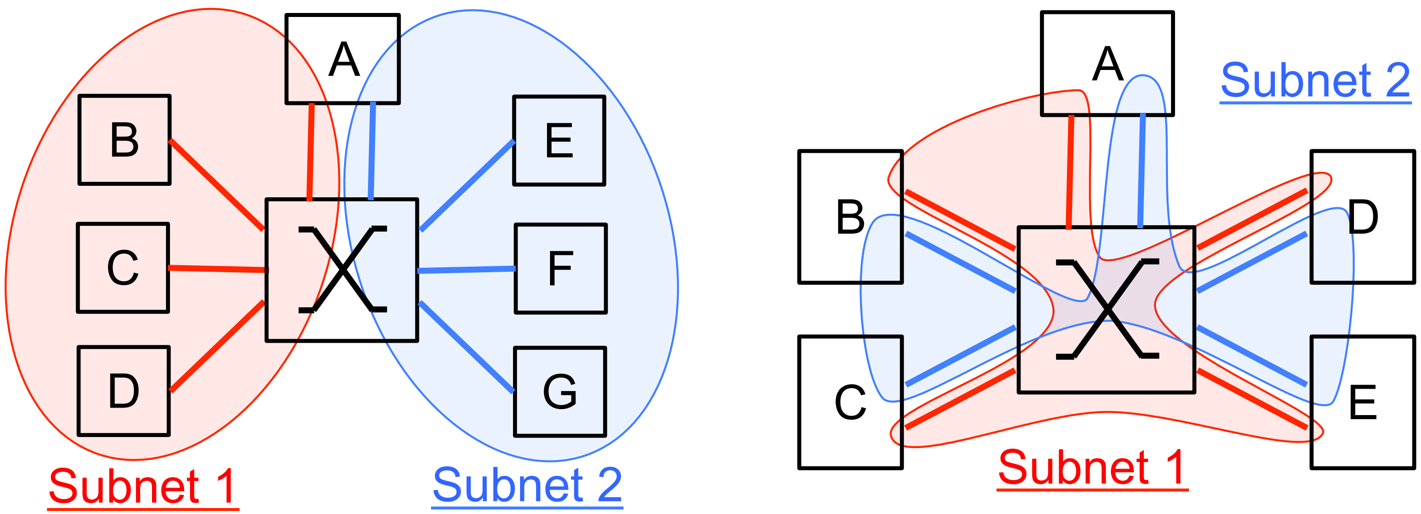
Accommodation of Any Upper Layer Protocol

- Limiting upper layer protocol is not necessary.
- Any SpW packet can be allowed to be transferred.

Additional FDIR Mechanisms (optional)

Subnet Concept

- A SpW network can be divided into some subnets which do not share any link.
  - Routers and end-nodes can be shared as long as there is no interference.
- Packets transferred among different subnets are inhibited.
- SpW-D can be applied on a per subnet basis.

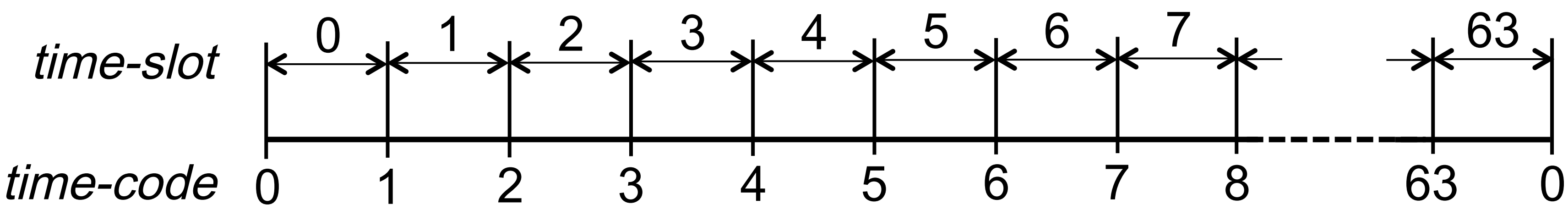


Flexible Time-Slotting

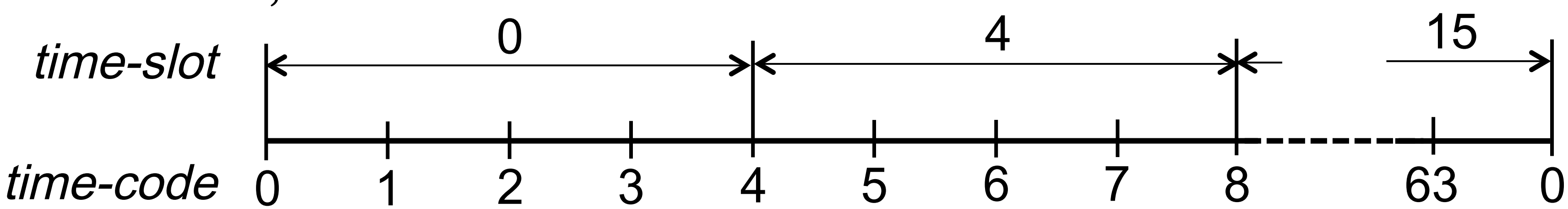
- Time-slots are delimited by the time-codes whose time values are multiples of 2^n (n = 0,…,6), where n is determined for each subnet.
  - When n = 0, Time-slots are delimited by all time-codes (corresponds with current SpW-D standard).
  - Effective in improving throughput with longer time-slots, when the time-code period needs to be kept short by different reasons.
- The number of time-slots is 2^m, where m is determined for each subnet.
- When $n+m > 6$, the mechanism to distribute extended time-code is necessary.

Time-Slotting Examples

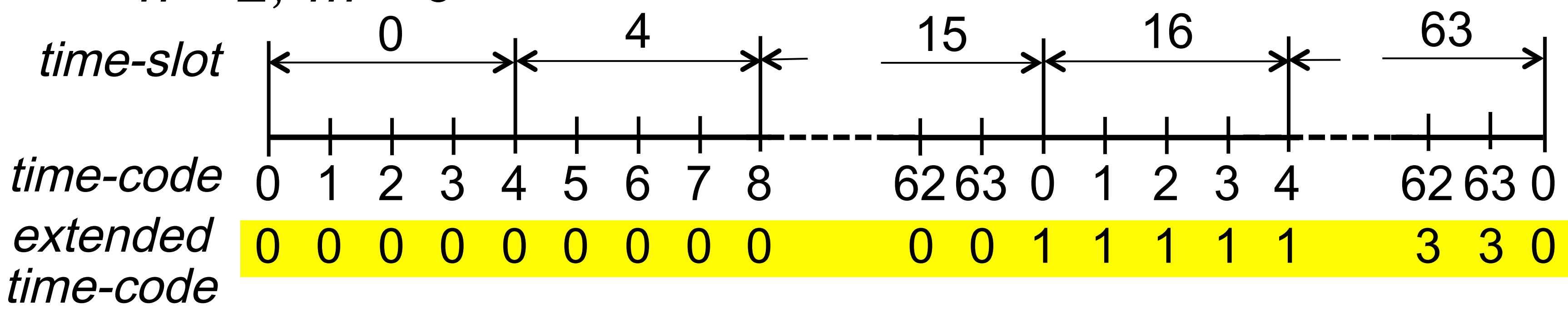
- $n = 0,\ m = 6$



- $n = 2,\ m = 4$



- $n = 2,\ m = 6$



Mode Concept

- The concept of “mode” is introduced to cope with the following cases.
  - Operational mode of the system is changed.
  - Packet routing and schedule need to be changed in case of a link failure.
- Subnet, time-slot, and schedule (of SpW-D) can be determined for each mode.
  - Mode change can occur at the end of a time-slot.
- The mechanism to distribute “current mode” to all nodes in the SpW network is assumed (similar to the distribution mechanism of extended time-code).
  - The mode distribution mechanism is out of the scope of our investigations.

Split RMAP Transaction

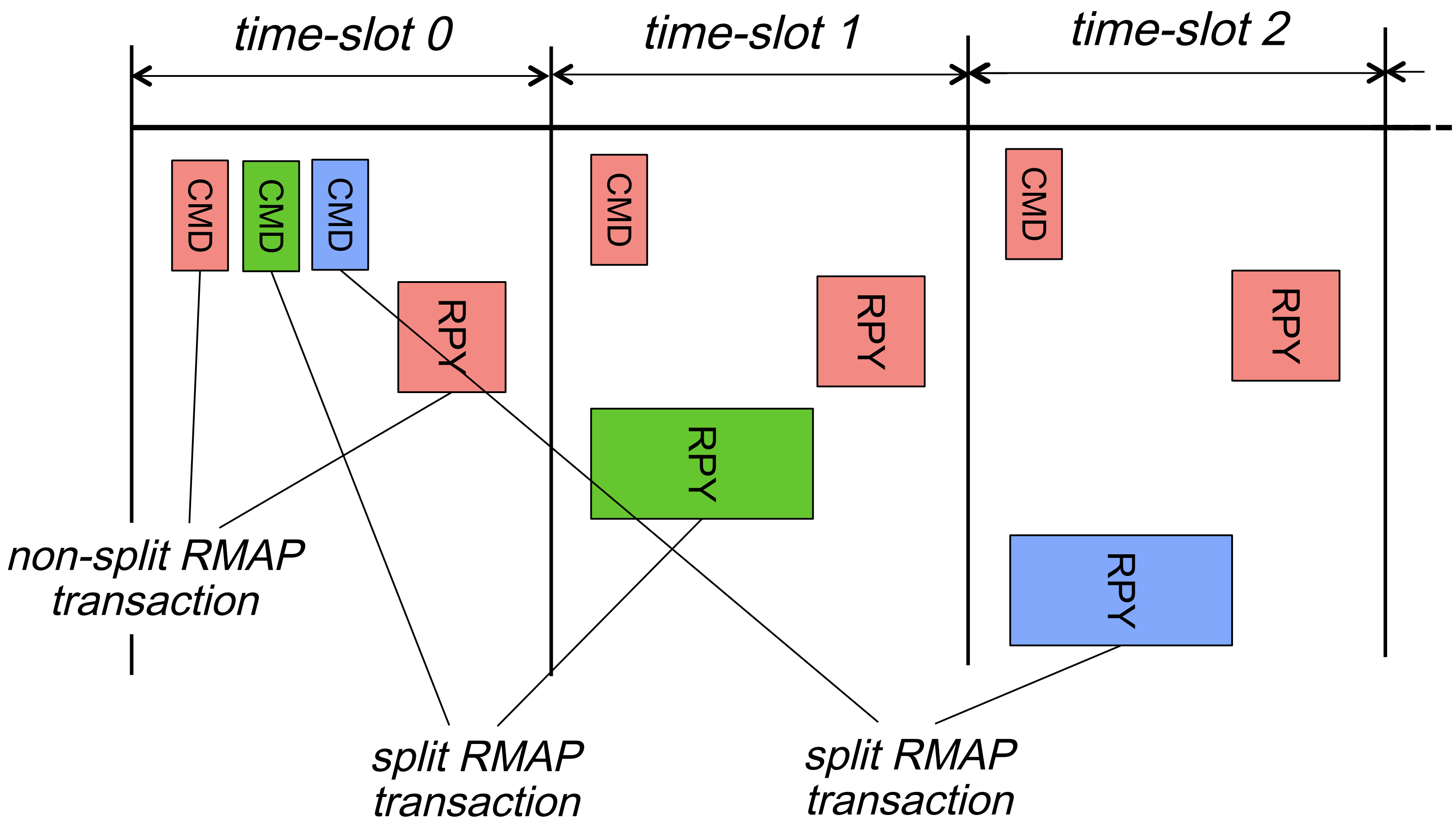
Motivation

- When an RMAP target node is implemented with software, the maximum RMAP reply delay is quite long, thus degrading the throughput.
  - RMAP reply delay = the time since an RMAP command is received until the RMAP reply is prepared.

Solution

- An RMAP target node is allowed to send the RMAP reply in predetermined later time-slot.
  - Only intelligent target nodes (implemented with processors and software) are allowed to use this option.

RMAP Transaction Examples



Schedule Information

- Each *intelligent* end-node has a schedule information which shows the packets that can be transferred.
- A schedule information holds a list of following data for each *link*, for each *mode*, and for each *time-slot*.
  - packet type (RMAP command, RMAP reply, others)

  when the packet type is not RMAP reply:
  - list of destination nodes
  - total size of packets

  when the packet type is RMAP command:
  - total size of RMAP reply packets
  - time-slot number in which the RMAP reply is scheduled
  - a list of reply addresses for each destination

  when the packet type is RMAP reply:
  - RMAP initiator node
  - time-slot number in which the RMAP command is sent

Schedule Information Example (1)

- example schedule information for a link of an intelligent end-node for a mode

time-slot number	packet type	destination node list/ RMAP initiator node	RMAP command/ reply time-slot number	total size of packets	total size of RMAP reply packets
0	RMAP command (non-verified write)	11, 12, 13, 14	0	1024	20
	others	15		500	
2	others	16		2000	
8	RMAP command (read)	11, 12, 13, 14	8	40	1024
10	others	17		2000	
12	RMAP command (read)	11, 12, 13, 14	12	40	1024
…	…	…	…	…	…
62	others	21		2000	

* The list of reply addresses for each destination is omitted.

Schedule Information Example (2)

- RMAP commands in time-slot 0 are replied in the same time-slot (non-split RMAP transaction).
- RMAP commands to in time-slot 8 and 12 are replied in time-slot 10 and 14, respectively (split RMAP transaction).

time-slot number	packet type	destination node list/ RMAP initiator node	RMAP command/ reply time-slot number	total size of packets	total size of RMAP reply packets
0	RMAP command (non-verified write)	11, 12, 13, 14	0	1024	20
	others	15		500	
2	others	16		2000	
8	RMAP command (read)	11, 12, 13, 14	*10*	40	1024
10	others	17		2000	
12	RMAP command (read)	11, 12, 13, 14	*14*	40	1024
…	…	…	…	…	…

Schedule Information Example (3)

- To the RMAP command received in time-slot 5 and 13 from node 1, this end-node replies in time-slot 6 and 14, respectively (split RMAP transaction).

time-slot number	packet type	destination node list/ RMAP initiator node	RMAP command/ reply time-slot number	total size of packets	total size of RMAP reply packets
0	RMAP command (non-verified write)	11, 12, 13, 14	0	1024	20
	others	15		500	
6	*RMAP reply*	*1*	*5*		
8	RMAP command (read)	11, 12, 13, 14	9	40	1024
10	others	17		2000	
12	RMAP command (read)	11, 12, 13, 14	14	40	1024
14	*RMAP reply*	*1*	*13*		
…	…	…	…	…	…

Additional FDIR Mechanisms (optional)

Discard packet at the end of each time-slot

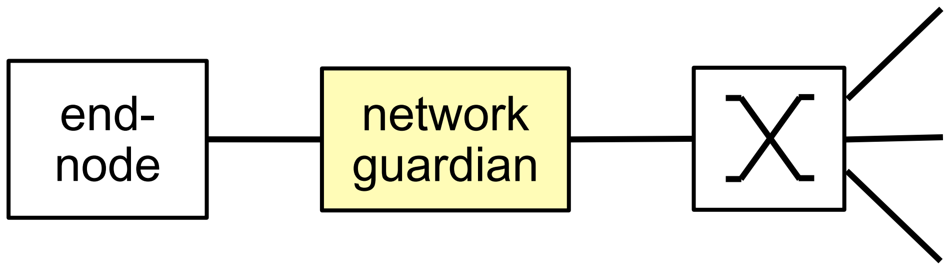
- The current SpW/SpW-D FDIR mechanisms do not cover the following cases.
  - A faulty end-node sends a long packet that is not fit in the time-slot.
  - A faulty end-node stops sending FCT while it receives a packet.
- At the end of each time-slot, routers and end-nodes that do not finish sending/receiving a packet should do the following FDIR processes.
  - When the node do not finish receiving a packet, it discards the data in its receive buffer, sends FCTs, and discards succeeding data until an EOP/EEP is received.
  - When the node do not finish sending a packet, it discards the remaining data and tries to send EEP.

Inhibit packet transferred among different subnets

- With the introduction of subnet concept, routers should inhibit the packets transferred among different subnets.

Network Guardian

- an additional device connected between a router and a end-node for adding optional FDIR mechanisms to an existing SpW device.
- If the network guardian has the schedule information that the end-node should have, it can detect illegal packet transfers and discard them.



Current and Future Activities

Methodology Guidebook

- We are making a guidebook that explains the proposed methodology for guaranteeing the real-time property of SpW network.
- A recommended calculation method whether a certain set of packets fit in a time-slot is included.

Software Platform Development

- We will develop a SpW middleware running on our real-time kernel and supporting tools that support the proposed methodology.
  - The SpW middleware is responsible for sending packets following the given schedule information.
- We plan to release “SpaceWire OS” consisting of the real-time kernel and the SpW middleware next year.