ISO/IEC JTC 1/SC 25 N 1164

Final Committee Draft ISO/IEC FCD	
Date: **2006-05-01**	Reference number: ISO/IEC JTC 1/SC 25 **N 1164**
Supersedes document SC 25 N/A	

THIS DOCUMENT IS STILL UNDER STUDY AND SUBJECT TO CHANGE. IT SHOULD NOT BE USED FOR REFERENCE PURPOSES.

ISO/IEC JTC 1/SC 25 **INTERCONNECTION OF INFORMATION TECHNOLOGY EQUIPMENT** Secretariat: Germany (DIN)	Circulated to P- and O-members, and to technical committees and organizations in liaison for: - discussion at - comment by - voting by (P-members only) **2006-09-11** Please return all votes and comments in electronic form using the attached template directly to the SC 25 Secretariat by the due date indicated.

ISO/IEC 24740

Title: **Information technology – Responsive Link (RL)**

Project: 1.25.13.10.18

Introductory note: **This FCD is distributed for approval as FDIS.**

The new work item proposal has been distributed with SC 25 N 974 and JTC 1 N 7436.

The work has been approved as recorded in SC 25 N 1005. The CD had been distributed with SC 25 N 1060and found substantial support as reported in SC 25 N1093. The comments have been resolved as recorded in SC 25 N 1093. The document has been updated accordingly and distributed here with for approval as FDIS.

REQUESTED: ACTION **National Member Bodies of ISO/IEC JTC 1/SC 25 are requested to vote on this document.**

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights (not listed in the draft) of which they are aware and to provide supporting documentation.

Medium: Defined

No. of pages:

Address Reply to: Secretariat, ISO/IEC JTC 1/SC 25, Dr.-Ing. Walter P. von Pattay Member of ZVEI FV 7 & FV 8, Gotthelfstr. 34, D- 81677 München, Germany Tel.: +49/89/923 967 57 ¸ Tfx.: +49/89/923 967 59 (only on request) EM: Walter@Pattay.com

Home page: http://www.iec.ch/sc25

FINAL COMMITTEE DRAFT | **ISO/IEC FCD 24740**

FINAL
COMMITTEE DRAFT

ISO/IEC
FCD 24740

Information technology –
Responsive Link (RL)

This document constitutes the communications protocol and interface of the *Responsive Link* that inter-connects computers for distributed real-time control applications.

CONTENTS

Address Reply to: Secretariat, ISO/IEC JTC 1/SC 25, Dr.-Ing. Walter P. von Pattay
Member of ZVEI FV 7 & FV 8, Gotthelfstr. 34, D- 81677 München, Germany
Tel.: +49/89/923 967 57 ¸ Tfx.: +49/89/923 967 59 (only on request)
EM: Walter@Pattay.com

Home page: http://www.iec.ch/sc25

List of Figures

List of Tables

INFORMATION TECHNOLOGY –
RESPONSIVE LINK (RL)

FOREWORD

1) ISO (International Organization for Standardization) and IEC (International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

2) In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

3) All users should ensure that they have the latest edition of this publication.

4) No liability shall attach to IEC or ISO or its directors, employees, servants or agents including individual experts and members of their technical committees and IEC or ISO member bodies for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication of, use of, or reliance upon, this ISO/IEC publication or any other IEC, ISO or ISO/IEC publications.

5) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

6) The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this International Standard may involve the use of patents concerning Japanese patent application No.3460090, No.2002-111658, US patent application No.6901074, and EU patent application No.99123864.3.

   The ISO and IEC take no position concerning the evidence, validity and scope of this patent right. The holder of this putative patent right has assured the ISO and IEC that they are willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of the putative patent rights is registered with the ISO and IEC. Information may be obtained from:

   National Institute of Advanced Industrial Science and Technology
   AIST Water Front 3F, 2-41-6, Aomi, Koto-ku, Tokyo 135-0064, Japan.

7) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights other than those identified above. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 24740 was prepared by subcommittee 25: Interconnection of information technology equipment, of ISO/IEC joint technical committee 1: Information technology.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

INTRODUCTION

The *Responsive Link* standard defines a real-time communication protocol between computers networked in a machine or in a site with point-to-point interfaces. Complex machines, such as robots, automobiles, and network routers, have a growing demand for distributed processing. In addition, modernization of facilities such as factories, offices, schools, and homes is creating a ubiquitous computing environment. Unlike conventional PC applications for documentation and Internet applications that exchange texts without hard time constraints, these types of cooperative computing require real-time and reliable responses to physical events occurring in the real world. Although most conventional real-time systems employ a single processor that invokes tasks corresponding to outside events, the need for distributed processing architecture is inevitable for the larger-scale systems mentioned above, because operational targets are physically separated and scalable processing power becomes necessary as the number of components increases. In order for distributed nodes to cooperate in real-time, an interconnecting network shall realize real-time communication. The *Responsive Link* provides such real-time communication between computers by providing dual full-duplex communication channels, fixed-size prioritized packets, packet overtaking based on priority, and automatic error correction.

Real-time

An operation of a dynamic system is called a **real-time** operation if the combined reaction- and operation-time of the task is shorter than the allowed maximum delay, with respect to circumstances outside the operation. A system is said to be a *hard real-time* system if the correctness of an operation depends not only on logical correctness, but also on the time at which it is performed. An operation performed after the deadline is, by definition, incorrect, and has no value. In a *soft real-time* system the value of an operation declines steadily after the deadline expires. Whether a system is hard real-time or soft real-time, and how much time is allowed before a deadline are application dependent.

Although it is known that static estimation of the time required for generating a response is important in order for a computer to output a response before a deadline, such static estimation is often very difficult or impossible. Instead, most real-time systems rely on priority-based scheduling.

Real-time scheduling

As real-time scheduling algorithms, the Earliest Deadline First (EDF) scheduler, the Rate Monotonic (RM) scheduler, and their variations have been established, as explained in Annex B. These algorithms commonly schedule tasks based on priorities determined by margins to deadlines. Thus, an important function is preemption, i.e. the scheduler deprives the execution privilege of the lower-priority task and allows the higher-priority task to run.

In order to realize real-time communication, a function similar to task preemption shall also be employed in communication. There are several methods for this communication preemption: provision of two separate channels for high- and low-priority packets, setting different communication paths according to priority and assigning a shorter path for high-priority communication, and allowing higher-priority packets to overtake lower-priority packets. *Responsive Link* has all of these capabilities.

Important features

Responsive Link has the following distinctive features that allow realization of real-time communication:

- Separation of data transmission and event transmission (See Annex A.)
- Prioritized routing: When multiple packets with different priority levels are sent to the same destination, different routes can be set in order to realize exclusive communication lines or detours. (See Annex E.)
- Priority-based packet overtaking: The packet with the highest priority overtakes the other packets at each node. (See Annex B and Annex D.)

- Packet acceleration/deceleration using priority replacement: Packet priority can be replaced with a new priority at each node in order to accelerate/decelerate packets under distributed control and realize real-time communication.
- Fixed packet size: 64-byte data and 16-byte event
- Point-to-point serial link (See Annex A.)
- Independent routing of the data link and the event link (See Annex E.)
- Hardware forward error correction to prevent retransmission.
- Other useful features are as follows:
- Variable link speeds (533, 267, 133, 66,7, 33,3, 16,7, and 8,3) Mbit/s
- Automatic reconfiguration (Plug & Play)
- Topology free
- Low latency (240 ns required to pass through one routing node at 66,7 Mbit/s)

Typical applications and operations

As a typical application of the *Responsive Link*, Figure 1 shows a distributed control configuration of a humanoid robot. The electronic control part of the humanoid robot consists of several control nodes with local sensing and actuating devices. The distributed controllers are connected to each other by *Responsive Link*. In this figure, rectangles represent node controllers, and dotted lines show the *Responsive Link* cable, which is a point-to-point serial link.

For a humanoid robot to walk stably, a servo loop of 1 ms or shorter is needed. In this configuration, the farthest two nodes can exchange a 16-byte packet within 5 *μsec*. Since the time is guaranteed not to fluctuate, the distributed control of a humanoid is considered to be sufficiently possible.

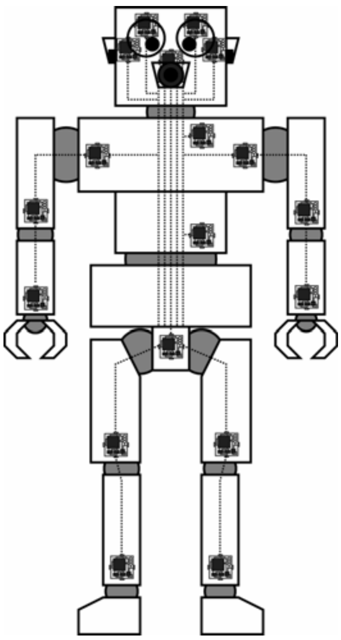

Figure 1 - A humanoid robot

INFORMATION TECHNOLOGY – *RESPONSIVE LINK* (RL)

1 Scope

This standard specifies the communications protocol and interface of *Responsive Link*, the real-time communication for parallel/distributed control. This standard roughly corresponds to the functions specified in layer 1 to layer 4 of the OSI reference model.

The purpose of this standard is to facilitate the development and use of *Responsive Link* in real-time systems by providing a common data protocol. This standard provides a real-time communication protocol for interconnections among distributed real-time systems, including embedded systems, control systems, amusement systems, robot systems, and intelligent buildings.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

2.1 Approved References

IEC 61156-5*, Multicore and symmetrical pair/quad cables for digital communications - Part 5: Symmetrical pair/quad cables with transmission characteristics up to 600 MHz - Horizontal floor wiring - Sectional specification*

ISO/IEC 11801:2002, *Generic cabling for customer premises*

ISO/IEC 15018, Generic cabling for homes

IEC 60603-7, *Connectors for frequencies below 3 MHz for use with printed boards - Part 7: Detail specification for connectors, 8-way, including fixed and free connectors with common mating features, with assessed quality*

IEC 60603-7-3, *Connectors for electronic equipment - Part 7-3: Detail specification for 8-way, shielded, free and fixed connectors, for data transmissions with frequencies up to 100 MHz*

IEC 60603-7-4, *Connectors for electronic equipment - Part 7-4: Detail specification for 8-way, unshielded, free and fixed connectors, for data transmissions with frequencies up to 250 MHz*

IEC 60603-7-7, *Connectors for electronic equipment - Part 7-7: Detail specification for 8-way, shielded, free and fixed connectors, for data transmission with frequencies up to 600 MHz (category 7, shielded)*

IEC 61076-3-104 Ed.2*: Connectors for electronic equipment - Part 3-104: Rectangular connectors - Detail specification for 8-way, shielded free and fixed connectors, for data transmissions with frequencies up to 1000 MHz (under consideration)*

ISO/IEC 7498-1, *Open System Interconnection – Basic reference model: The basic model)*

2.2 Other References

For information on the current status of the listed documents or regarding availability, contact the indicated organization.

TIA/EIA-644, *Low-Voltage Differential Signaling (LVDS)*

Note - Telecommunications Industry Association (TIA) / Electronic Industries Alliance (EIA) 644-1995 standard specify Low-Voltage Differential Signaling (LVDS).

3 Definitions, terms and abbreviations

3.1 Definitions

For the purpose of this standard, the following definitions apply:

3.1.1
byte
A group of eight bits.

3.1.2
data
A data set transmitted through a data link.

3.1.3
data link
A medium on which data packets are transmitted.

3.1.4
data packet
A data packet consists of a 4-byte header including the network address with a priority, a 56-byte payload, and a 4-byte trailer.

3.1.5
destination
A node at the end of the interface that receives data.

3.1.6
event link
A medium on which event packets are transmitted.

3.1.7
event
A data set transmitted through an event link.

3.1.8
event packet
An event packet consists of a 4-byte header including the network address with a priority, an 8-byte payload, and a 4-byte trailer.

3.1.9
packet
A data set sent from a source to a destination. There are two kinds of packets: the data packet and the event packet.

3.1.10
priority
Priority is an 8-bit value from 0 (0x00) to 255 (0xff) attached to a packet. The larger the number, the higher the priority.

3.1.11
source
A node at the end of the interface that transmits data.

3.1.12
word
A unit of information, consisting of 32 bits. A word contains an ordered set of four bytes.

3.2 Terms and abbreviations

BUFX	overtaking BUFfer Xth.
DPLL	Digital Phase Lock Loop.
DRAM	Dynamic Random Access Memory
FATAL	FATAL bit to indicate uncorrectable error.
INFO	INFOrmation digits to be transmitted.
LSB	Least Significant Bit.
MSB	Most Significant Bit.
NRZI	Non-Return-to-Zero-Inverted.
RED	REDundant digits for error correction.
SDRAM	Synchronous DRAM.

4 Conformance

For an entity to conform to this international standard, the following applies:

a) The structure and configuration shall conform to the requirements of Clause 5.1.

b) The layer 1 structure and configuration shall conform to the requirements of Clause 6.1.

c) The layer 2 protocol shall conform to the requirements of Clause 7.

d) The layer 3 protocol shall conform to the requirements of Clause 8.

e) The layer 4 protocol shall conform to the requirements of Clause 9.

5 Responsive Link structure

5.1 Outline of Responsive Link

For better real-time performance a *Responsive Link* consists of two communication links, namely a data link and an event link. A node has at least one *Responsive Link* port. Normally, each node has a *Responsive Link* switch with multiple ports. The *Responsive Link* switch of each node has a unique node address. The priority of a packet is set by software, such as a real-time scheduler, based on any time constraints. A source node address, a destination address, and a priority are attached to the header of a packet by software for routing. In order to route a packet from a source node to a destination node according to the address with priority, every *Responsive Link* switch has a routing table and routing capability. An input packet to an input port of the *Responsive Link* switch is routed to (an) output port(s) according to the routing table. The data link and the event link are routed independently. When multiple packets with different priorities are sent to the same destination, different routes can be set in order to realize exclusive communication lines or detours.

In order to realize fine-grained real-time communication, packets with higher priorities overtake other packets at each node. Packet priority can be replaced with a new priority at each node in order to accelerate/decelerate packets under distributed control. The link speed of *Responsive Link* can be changed dynamically between (533, 267, 133, 66,7, 33,3, 16,7, 8,3) Mbit/s in order to balance real-time capability, power consumption, and noise tolerance. *Responsive Link* also supports automatic reconfiguration (Plug & Play) in order to reconstruct systems easily.

5.2 OSI Reference Model

This standard covers the functionality of layer 1 to layer 4 of the OSI reference model, although the packet format does not have a layered structure for better real-time performance, especially for the routing hardware to handle packets efficiently. (See ISO/IEC 7498-1.)

6 Layer 1 (physical layer)

6.1 Separate transmission of data and event

The event transmission shall be separated from the data transmission as shown in Figure 2. (See Annex A.) Each link shall be composed of a full-duplex serial link that connects point-to-point. The data link and the event link can be used for both soft and hard real-time communication. Although each of event and data links can ensure real-time communication, better real-time service is accomplished when they are used cooperatively. (See clause 8.2.3 and clause 8.2.4.)

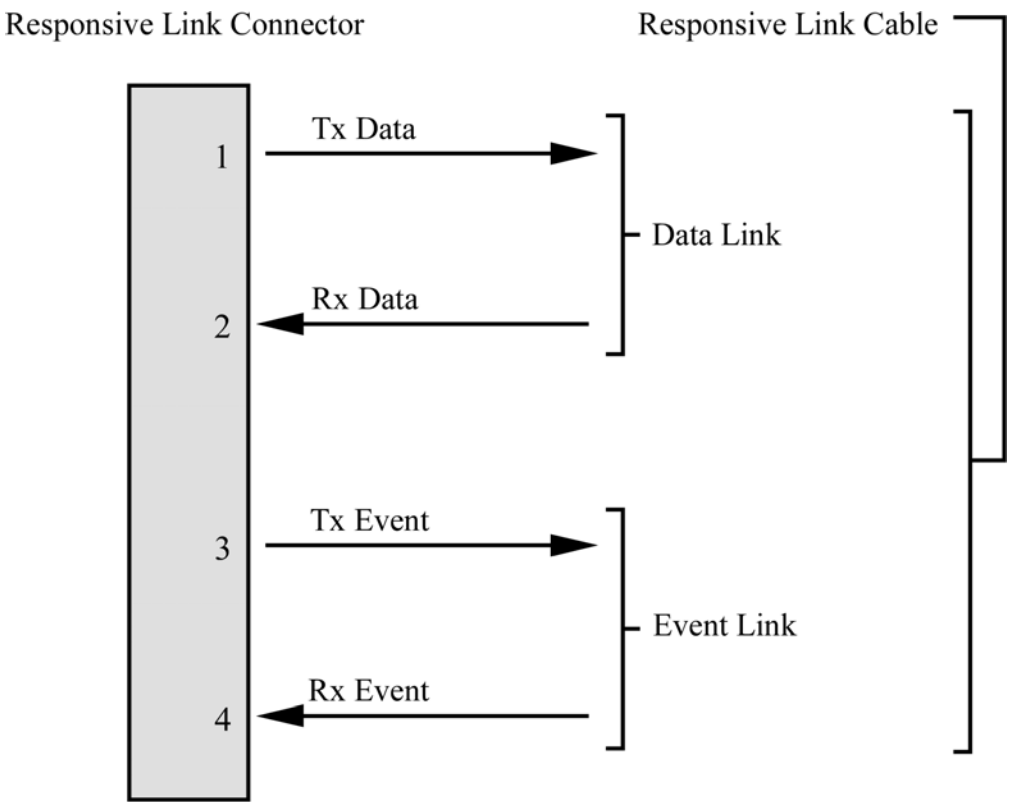

Figure 2 – Logical Interface of Responsive Link

6.2 Physical Interface

The recommended physical interface of Responsive Link is described at Annex C.

7 Layer 2 (data link layer)

7.1 Error correction

7.1.1 General

Responsive Link shall provide error-free transmission for reliable real-time control. Error correction shall be performed by hardware at every hop. *Responsive Link* shall perform error correction on a byte-by-byte basis. Four extra bits shall be attached to each byte for this purpose. This 12-bit frame enables one-bit error correction and two-bit error detection for a one-byte data (8 bits of information digits).

Since *Responsive Link* attaches 4 bits for error correction to a byte (8 bits), the communication speed at 100 MHz is approximately equal to 66,7 Mbit/s.

7.1.2 CODEC

The CODEC shall handle a 12-bit frame consisting of 8 bits of information digits and 4 bits of redundant digits. First, the original 8 bits of information digits shall be encoded into a 12-bit frame by the Hamming cyclic coding method. Additional 'zero' bits shall be then inserted into a bit sequence in which more than five successive 'one' bits exist in order to ensure bit synchronization. Finally, the encoded digits shall be transmitted in a Non-Return-to-Zero-Inverted (NRZI) waveform. Each coding method is described in the following section.

7.1.3 Error correction encoding

The Hamming cyclic code with the generator polynomial of x^4+x+1 shall be used to detect and correct errors in a transmitted frame. With this coding, in which 4 bits of redundant digits shall be appended to the LSB side of an 8-bit information data, any 1-bit error in a frame can be corrected at a receiver without re-sending.

The redundant digits can be generated as the remainder of the operation in which an 8-bit message is shifted to the left by 5 bits and then divided by the generator polynomial '10011' in modulo 2 arithmetic, i.e., 'XOR'ing. At the receiver, the received 12-bit frame is divided by '10011' in modulo 2 arithmetic, and a 4-bit syndrome is obtained as the remainder. Since the syndrome is peculiar to the position of a single-bit error, as shown in the Table 1, the receiver can detect and correct the error by inverting the corresponding digit. Each frame shall be transmitted on the link in a bit-by-bit manner from the MSB.

Table 1 - Syndrome and error digits

Syndrome	Position of the error digit
0000	00000000 0000 (no error)
0001	00000000 0001
0010	00000000 0010
0100	00000000 0100
1000	00000000 1000
0011	00000001 0000
0110	00000010 0000
1100	00000100 0000
1011	00001000 0000
0101	00010000 0000
1010	00100000 0000
0111	01000000 0000
1110	10000000 0000

7.1.4 Bit stuffing

When five successive ‘one’ digits appear in a transmitted bit stream, an additional ‘zero’ digit shall be inserted after them. This bit stuff procedure decreases the DC component caused by the long succession of ‘one’ digits and ensures bit synchronization by the DPLL in the receiving circuits.

7.1.5 NRZI encoding

Each digit in a frame shall be eventually encoded as a Non-Return-to-Zero-Inverted (NRZI) waveform and transmitted. With the NRZI encoding, the signal level on the link shall change from low to high or from high to low when a ‘zero’ digit is transmitted, and keep the previous level when a ‘one’ digit is sent.

7.2 Frame format

7.2.1 Packet

A packet, which consists of multiple 12-bit frames, shall have different structures for the data link and the event link.

- Data packet: 64 frames
- Event packet: 16 frames

7.2.2 Frame

A frame shall consist of 12 bits, including 8 bits of information digits and 4 bits of redundant digits, as shown in Table 2.

Table 2 - Frame format

INFO	RED
dddddddd	rrrr
INFO: Information digits to be transmitted. RED: Redundant digits for error correction	

7.2.3 Setup pattern

After the power is first applied, or after an unexpected burst link error occurs, the synchronization between the sender and the receiver can be lost. In such a situation, the link shall be initialized explicitly. The encoder in the initial mode shall send the setup pattern shown below. Since six successive 'one' digits violate the bit stuffing rule, the decoder can distinguish the pattern from normal frames and thus switches to the initial mode. The initialized decoder interprets the first receiving frame after the initialization as the start frame of a new frame.

Setup pattern
000001111110

7.2.4 Idle pattern

When an encoder has no actual communication data, the encoder shall send the idle pattern shown below in order to maintain the synchronization of the link.

Idle pattern
000000001111

7.2.5 Bit synchronization and clock rate

A Digital Phase Lock Loop (DPLL) mechanism should be used for decoding. The received signal is sampled at a rising edge of the base clock for the DPLL. The number of samplings for a 1-bit transmission can be dynamically changed by setting the sampling mode. The base clock for the DPLL is (400, 800, 1 600, or 3 200) MHz, and the number of samplings is 4, 8, 16, or 32. Therefore, the maximum clock rate of *Responsive Link* is (100, 200, 400, or 800) MHz.

The DPLL detects edges of the receiving signal and generates a receiving clock according to the period of the current sampling mode. The receiving clock is adjusted so that the raising edge appears in the center of two edges of the receiving signal.

7.2.6 Error handling

The encoding scheme of *Responsive Link* can automatically detect and correct any 1-bit error in a frame. However, when errors of greater than 2 bits are present in a frame, the error correction mechanism does not work. In such a situation, the calculation of the syndrome results in one of the following two cases:

1. Value not listed in Table 1

In this case, the decoder can detect an unrecoverable fatal error. If a fatal error is detected in the header (the first four frames) of a receiving packet, then the decoder shall abort the reception of the packet and should interrupt the controller (processor). When a fatal error is present in another part of the packet, the decoder shall not try to correct digits in the received data and set the FATAL bit in the status frame of the packet.

2. Value is listed in Table 1

Although the probability of this case is very low, the decoder cannot detect the occurrence of the fatal error in this case. Since this error is indistinguishable from other correctable 1-bit errors, the received frame is inadequately modified by the decoder. This situation allows transmission of an incorrect packet and is highly undesirable. Therefore, when simple 1-bit errors are corrected in two successive packets, the decoder shall consider this to be a fatal error that cannot be corrected, and so handle the packet in the manner described above.

The above information shall be stored in the trailer of the packet. (See clause 8.2.3 and clause 8.2.4.)

7.3 Automatic reconfiguration (plug&play)

Responsive Link shall support automatic reconfiguration, also known as the Plug & Play function, at the layer 2 protocol level as specified in this subclause.

When the decoder of the *Responsive Link* receiver receives seven successive 'one' digits, the link controller shall interpret this signal as an indication that the *Responsive Link* cable is disconnected. Then, the *Responsive Link* controller shall interrupt its host controller (processor) to inform it of the link-down status.

When the decoder of the *Responsive Link* receiver receives the initial packet (setup pattern) in the disconnected state, the link controller shall regard the *Responsive Link* cable to be engaged. Then, the *Responsive Link* controller shall interrupt its host controller (processor) to inform it of the link-up status.

When in the disconnected state, the encoder of the *Responsive Link* transmitter shall always try to send the initial packet. In this manner, when an active *Responsive Link* cable is plugged into the *Responsive Link* connecter, the encoder can force initialization of the connected decoder at the opposite side.

8 Layer 3 (network layer)

8.1 Packet overtaking function

Each packet shall have a priority. A packet shall consist of a header (a source address and a destination address with a priority), a payload, and a trailer (control and status information). If no collision occurs at a node, then the data is transmitted through the node. If a collision occurs at a node, then the packet with the higher priority shall overtake the other packets. (See Annex B and Annex D.)

If packets having the same priority continuously collide at the node, then the round-robin rule shall be applied and the packet that arrived first shall sent out first.

8.2 Responsive Link packet format

8.2.1 Header format

A packet shall have a 4-byte header that represents a source address, a destination address, and a priority, as shown in Figure 3. The source and destination addresses shall be assigned with 12-bit fields, and the priority shall be assigned with 8-bit fields. Priority[7 to 4] corresponds to the 7 to 4 bits of the 8-bit priority, and Priority[3 to 0] corresponds to the 3 to 0 bits of the 8-bit priority. Each node shall have a unique node address.

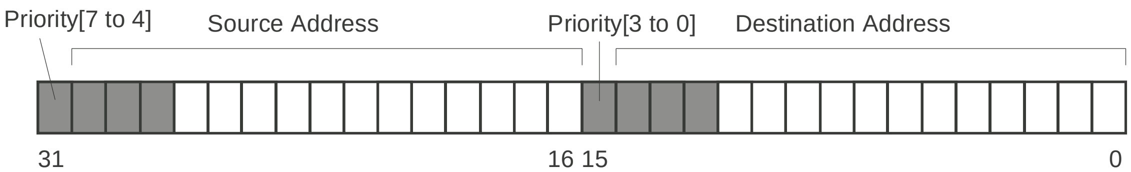

Figure 3 - Header format

8.2.2 Priority

Each packet shall have an 8-bit priority included in a packet header. The greater priority number, the higher priority. The highest priority is 255 (0xff), and the lowest priority is 0 (0x00).

8.2.3 Data packet

Figure 4 shows the format of a *Responsive Link* data packet.

A data packet shall consist of a 4-byte header composed of a network address with a priority, a 56-byte payload, and a 4-byte trailer composed of control and status information.

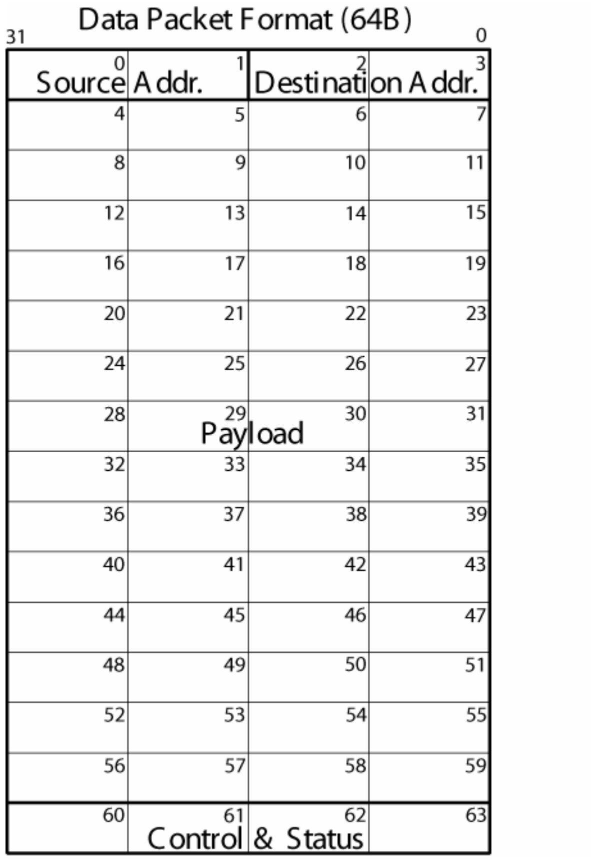

Figure 4 - Data packet format

Figure 5 shows the format of the 4-byte trailer.

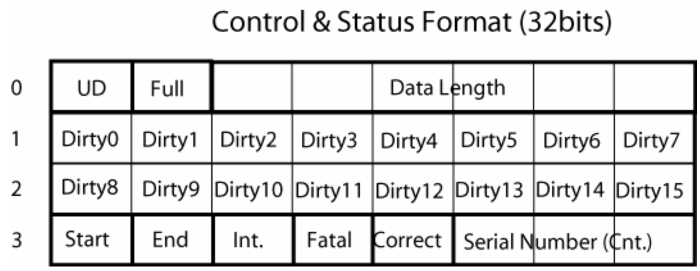

Figure 5 - Trailer format of data packet

- UD: Reserved.
- Full: This bit shall be set to 1 if all of the payload data (56 bytes) are valid data. Otherwise, this bit shall be set to 0. This flag shall be set by the Layer 4 hardware or software at the source node.
- Data Length: Length of valid payload data (0 to 56). This flag shall be set by the Layer 4 hardware or software at the source node.

- Dirty0 to Dirty15: Indicates which word (4 bytes) in the packet has an error. For example, if the 2nd word of the packet has an error, then Dirty1 bit shall be set to 1. This flag shall be set by the Layer 2 hardware as a result of the error correction procedure.
- Start: This bit shall be set to 1 if this packet is the start packet. Otherwise, this bit shall be set to 0. This flag shall be set by the Layer 4 hardware or software at the source node.
- End: This bit shall be set to 1 if this packet is the end packet. Otherwise, this bit shall be set to 0. This flag shall be set by the Layer 4 hardware or software at the source node.
- Int: The *Responsive Link* shall generate an interrupt to the destination node controller (processor) when this packet reaches the destination node if this bit is set to 1. Otherwise, this bit shall be set to 0. This flag shall be set by the Layer 4 hardware or software at the source node.
- Fatal: This bit shall be set to 1 by the Layer 2 hardware if any byte in the packet has an unrecoverable fatal error. Otherwise, this bit shall be set to 0.
- Correct: This bit shall be set to 1 by the Layer 2 hardware if this packet had errors that have been corrected. Otherwise, this bit shall be set to 0.
- Serial Number: Serial number of the packet. The start packet shall have the serial number 0, and this number shall be incremented in following packets. The serial number shall return to 0 after it reaches 7, and the sequence shall be repeated. This flag shall be set by the Layer 4 hardware or software at the source node.

In order to transmit a large chunk of data that cannot be accommodated in one packet, the start and end bits are used to identify the first packet and the last packet in a series of packets. If the logical size of a packet is smaller than or equal to 56 bytes then both the start bit and the end bit shall be set.

8.2.4 Event packet

Figure 6 shows the format of an event packet of *Responsive Link*. The event packet shall consist of a 4-byte header, an 8-byte payload, and a 4-byte trailer.

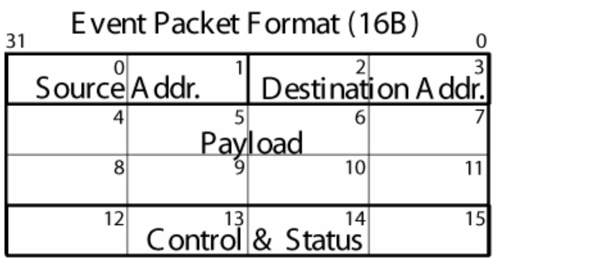

Figure 6 - Event packet format

Figure 7 shows the format of the 4-byte trailer.

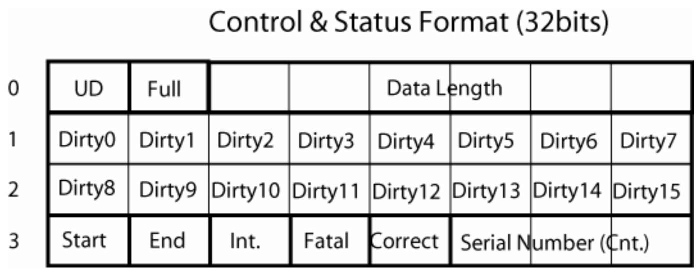

Figure 7 - Trailer format of event packet

- UD: Reserved.

- Full: This bit shall be set to 1 if all of the payload data (8 bytes) are valid data. Otherwise, this bit shall be set to 0. This flag shall be set by the Layer 4 hardware or software at the source node.
- Data Length: Length of valid payload data (0–8). This flag shall be set by the Layer 4 hardware or software at the source node.
- Dirty0 to Dirty15: Indicates which byte in the packet has an error. For example, if the 2nd byte of the packet has an error, then Dirty1 bit shall be set to 1. This flag shall be set by the Layer 2 hardware as a result of the error correction procedure.
- Start: This bit shall be set to 1 if this packet is the start packet. Otherwise, this bit shall be set to 0. This flag shall be set by the Layer 4 hardware or software at the source node.
- End: This bit shall be set to 1 if this packet is the end packet. Otherwise, this bit shall be set to 0. This flag shall be set by the Layer 4 hardware or software at the source node.
- Int: The *Responsive Link* shall generate an interrupt to the destination node controller (processor) when this packet reaches the destination node if this bit is set to 1. Otherwise, this bit shall be set to 0. This flag shall be set by the Layer 4 hardware or software at the source node.
- Fatal: This bit shall be set to 1 by the Layer 2 hardware if any byte in the packet has an unrecoverable fatal error. Otherwise, this bit shall be set to 0.
- Correct: This bit shall be set to 1 by the Layer 2 hardware if this packet had errors that have been corrected. Otherwise, this bit shall be set to 0.
- Serial Number: Serial number of the packet. The start packet shall have the serial number 0, and this number shall be incremented in following packets. The serial number shall return to 0 after it reaches 7, and the sequence shall be repeated. This flag shall be set by the Layer 4 hardware or software at the source node.

In order to transmit a large chunk of data that can not be accommodated in one packet, the start and end bits are used to identify the first packet and the last packet in a series of packets. If the logical size of a packet is smaller than or equal to 8 bytes, then both the start bit and the end bit shall be set.

8.3 Routing

Responsive Link realizes an end-to-end connection by setting the routing tables of all nodes along the transmission path from a source node to a destination node.

8.3.1 Routing table

Each node shall have a routing table to control the packet routing and the priority replacement function. Figure 8 shows the routing table of a network switch with five-ins and five-outs.

The reference part shall be the same as the header of a packet.

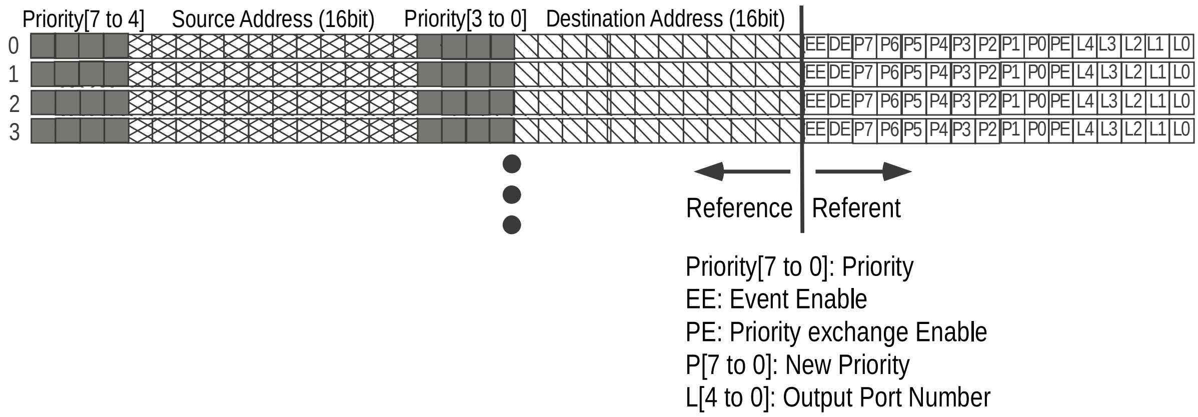

Figure 8 - Routing table

The referent part shall be composed of five parts as follows:

- EE (Event Enable): This bit shall be set to 1 if the event link is valid; otherwise, this bit shall be set to 0.
- DE (Data Enable): This bit shall be set to 1 if the data link is valid; otherwise, this bit shall be set to 0.
- PE (Priority Exchange): This bit shall be set to 1 if the priority replacement function is enabled; otherwise, this bit shall be set to 0.
- P[7 to 0]: New priority level (8 bits). Valid if PE is set to 1.
- L[4 to 0]: Output link numbers. For example, if the L3 bit is set, input packets shall be routed to the link3 port. If several bits are set, then multi-cast is indicated. If all of the bits are set, then broadcast is indicated.

8.3.2 Independent routing of data and event

The Event link and the Data link may have different routes to the same destination, because the EE bit and the DE bit can be set independently. (See Annex D.)

8.3.3 Priority-based routing

Priority bits are used for matching the routing table. So different routes can be set according to priority, even if the network address is the same. The route of priority “0”, which is the lowest priority, shall be used as the default route of the network address. If the combination of the network address and the priority is matched completely in the routing table, then the matched line shall be used. If, in the routing table, the network address matches, but the priority does not match, then the line with priority 0 shall be used. (See Annex E.)

9 Layer 4 (transport layer)

9.1 Priority replacement for packet acceleration/deceleration

The priority of packets can be replaced at each node. If the PE bit is set, Priority[7 to 0] in Figure 8, which is the original priority of the input packet, shall be replaced with P7 to P0 in Figure 8, which becomes the new priority. The original priority is used for routing and overtaking packets at the current node, and the new priority is used at the next node. (See Annex B.) If the priority of a packet is replaced, then the route of the packet may be changed thereafter.

A packet in which the priority is replaced with a higher priority is accelerated from the next node. A packet in which the priority is replaced with a lower priority is decelerated from the next node.

The packet control can be realized by setting the routing table at each node in a distributed control manner.

9.2 Multi-link

Responsive Link supports multi-links. Any two nodes in a system can communicate with each other on multiple different links by assigning different priorities to each packet, as described in 3. In this manner, the priority is used for route selection. Thus, the multi-link can increase the bandwidth between communication-intensive nodes. (See Annex E.)

9.3 Stream data transmission

In order to transmit a large amount of data that cannot fit into one packet, the following fields in the packet trailer shall be set at the source node. (See clause 8.2.3 and clause 8.2.4.)

- Full: This bit shall be set to 1 if all of the payload data are valid data. Otherwise, this bit shall be set to 0.
- Data Length: Length of valid payload data.
- Start: This bit shall be set to 1 if this packet is the start packet. Otherwise, this bit shall be set to 0.
- End: This bit shall be set to 1 if this packet is the end packet. Otherwise, this bit shall be set to 0.
- Int: The *Responsive Link* shall generate an interrupt to the destination node controller (processor) when this packet reaches the destination node if this bit is set to 1. Otherwise, this bit shall be set to 0.
- Serial Number: Serial number of the packet. The start packet shall have the serial number 0, and this number shall be incremented in following packets. The serial number shall return to 0 after it reaches 7, and the sequence shall be repeated. This flag shall be set by the Layer 4 hardware or software at the source node.

Annex A
(informative)

Characteristics of real-time communications

Generally, communication data are divided into minimum units, herein called packets, for transmission. Because multi-task systems are required to simultaneously communicate with other systems, they process concurrently and transmit packets to other systems.

The size of normal data, including image data, sound data, and sensor data, is large, whereas the size of event data, including inter-module interrupts, inter-module synchronization, and control commands, is very small. Thus, there are a large number of normal data packets and a small number of event packets that are very important for real-time simultaneous control on a single shared communication line. The event communication is required to be hard real-time, and the data communication, including multi-media data transmission (moving picture, voice, etc.), may be required to be soft real-time. It is very important to keep the communication latency of events as short as possible and to guarantee their arrival time in a real-time system that controls I/O devices.

It is difficult to realize a hard real-time system using conventional communication that shares a single communication line for data and events, because the communication latency of events cannot be estimated accurately. It is also difficult for conventional communication to bound a time constraint, because the communication bandwidth depends on the number of communication nodes on a serial bus and changes dynamically.

Moreover, the following trade-off in real-time communication exists.

Soft real-time: Guaranteed communication bandwidth => high throughput for bulky data

Hard real-time: Guaranteed communication latency => short latency for events

The packet size becomes larger as the throughput becomes higher. At the same time, the packet latency becomes longer in this case, as shown in Table A.1. The higher the communication throughput, the longer the latency. On the other hand, if the packet size becomes smaller, the packet latency becomes shorter, whereas the throughput becomes lower as a result of the increased overhead.

Table A.1 – Syndrome and error digits

Points \ Packet size	**Larger**	**Smaller**
Throughput	Higher	Lower
Latency	Longer	Shorter

Therefore, a *Responsive Link* consists of two point-to-point full-duplex communication links.

Annex B
(informative)

Real-time scheduling

There are several real-time scheduling algorithms, including Earliest Deadline First (EDF), which is an optimal dynamic scheduling algorithm, and Rate Monotonic (RM), which is an optimal static scheduling algorithm.

The EDF algorithm translates the deadline to a priority. The priority of the task with the earliest deadline becomes the highest.

The RM algorithm translates the cycle time to a priority. The task with the shortest cycle is assigned to have the highest priority.

Many other real-time scheduling algorithms also translate the time constraint to a priority.

Figure B.1 shows an example of EDF scheduling. Priority-based scheduling is performed at every clock tick and at timings when tasks are released (invoked), as well as at execution finish.

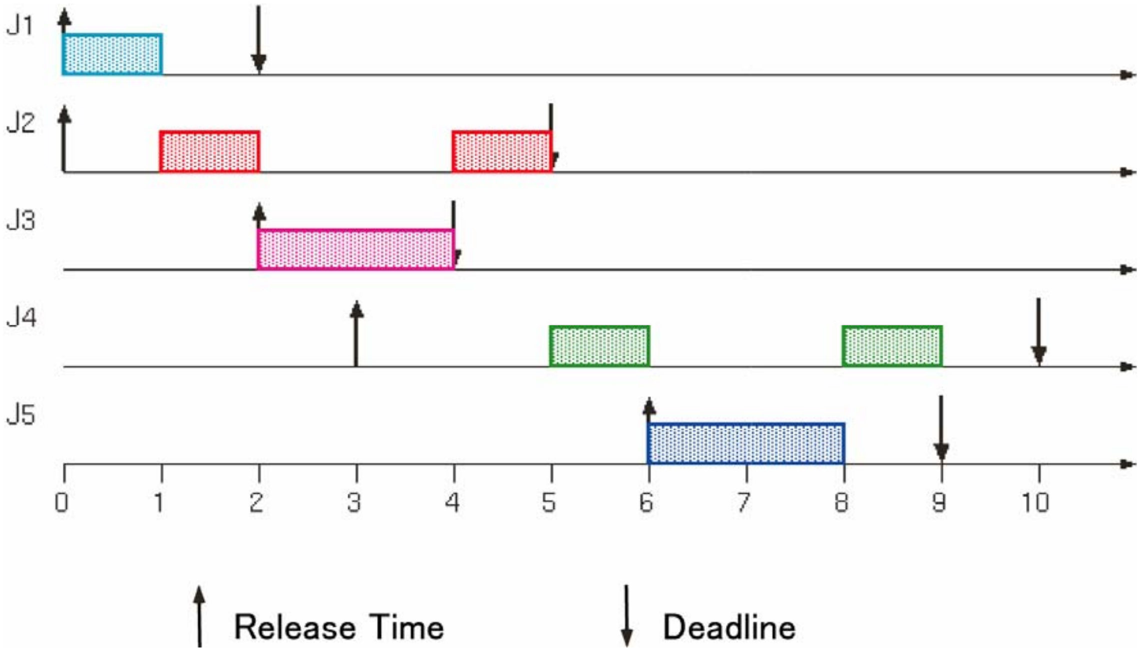

Figure B.1 - EDF scheduling

This real-time task scheduling process can be regarded as an overtaking process, i.e. tasks with higher priorities are executed earlier than tasks with lower priorities. In order to implement this idea in a distributed real-time system, communication of higher priority tasks should be able to overtake other communication. The *Responsive Link* does this at every node.

Annex C
(informative)

An Implementation of the *Responsive Link* interface

The electronic interface of the *Responsive Link* should be a type of LVDS (Low Voltage Differential Signal).

Cables that meet the requirements of the enhanced category 5, category 6, or category 7 of IEC 61156-5 should be used as the *Responsive Link* medium as shown in Figure C.1. And IEC 60603-7 connectors should be used as the *Responsive Link* connectors.

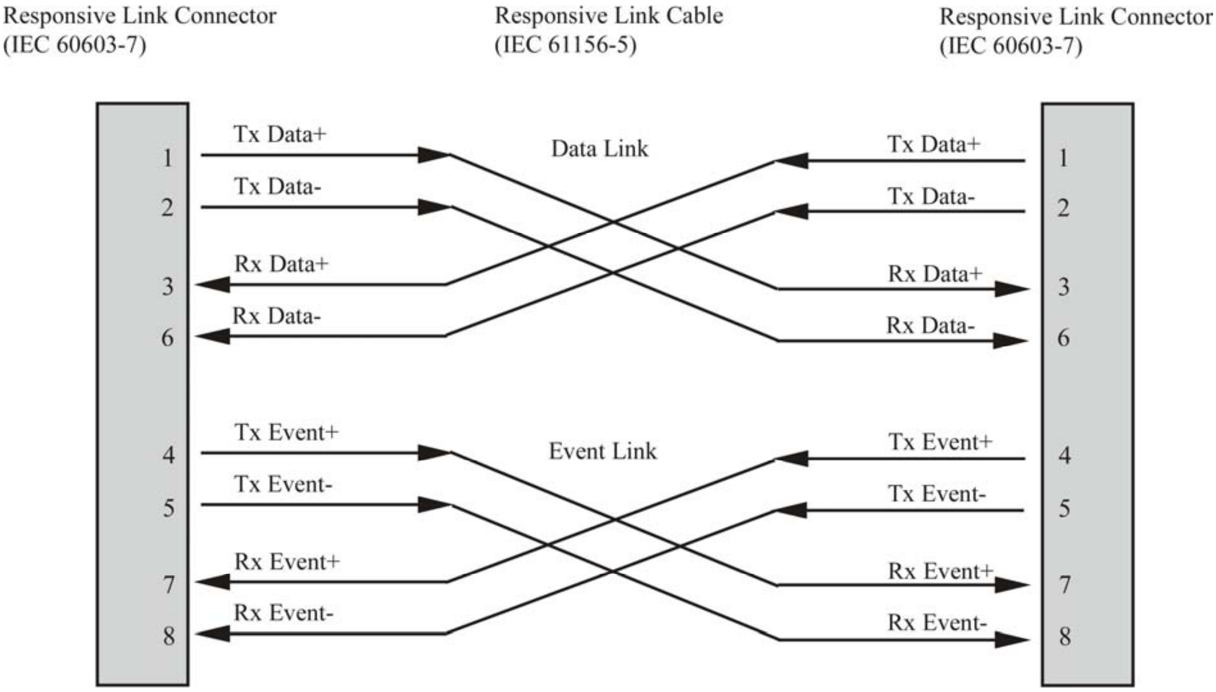

Figure C.1 - Responsive Link connector and cable

The maximum cable length of the *Responsive Link* changes according to the link speed as shown in Figure C.1.

Table C.1 - Maximum cable length

Clock rate (MHz)	100	200	400	800
Speed (Mbit/s)	66,7	133	267	533
Length (m)	20	15	10	5
Cable performance	Category 5e of IEC 61156-5	Category 5e of IEC 61156-5	Category 6 of IEC 61156-5	Category 7 of IEC 61156-5
Connector performance	IEC 60603-7-2	IEC 60603-7-2	IEC 60603-7-4	IEC 60603-7-7

Annex D
(informative)

D.1 An implementation of the *Responsive Link* switch

Responsive Link allows packets with higher priorities to overtake packets with lower priorities at each node whenever packet collisions occur.

Figure D.1 shows an implementation (a 5 by 5 network switch) of the *Responsive Link* switch with the packet overtaking function. Port 0 is connected to a local device, such as the node processor, and Ports (1 to 4) are connected to external ports. A packet arriving at an in-port without collision is transferred to an out-port specified by the routing table. When a collision occurs, i.e. two packets are routed to the same out-port simultaneously, the packet with the higher priority is served first and the other is stored in the overtaking buffer. After the in-packet with the higher priority is output, the in-packet with the lower priority is output to the same out-port.

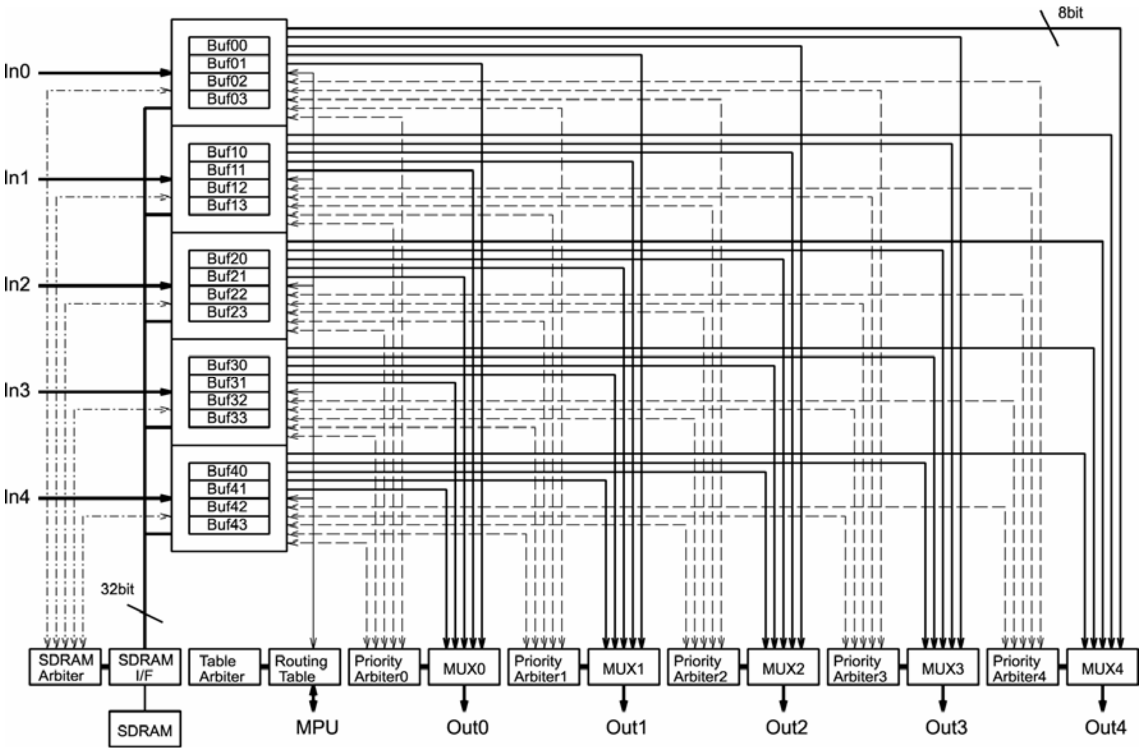

Figure D.1 - A Responsive Link switch

D.2 An implementation of overtaking buffers

Figure D.2 shows the input part of the *Responsive Link* switch shown in Figure D.1. The size of the overtaking buffer (BUFX) is equal to the packet size.

The header of a packet arriving at an input port is stored in an overtaking buffer. A routing table is then used to look up the header. The link strobes that indicate the output ports are obtained from the routing table. Each output port scans the overtaking buffers to find the packet with the highest priority.

If no collision occurs, then the input packet is cut through the output port. If a collision occurs, then the packet with the highest priority is cut through the output port. The packet with the lowest priority is stored in the overtaking buffer until the packets with higher priorities have been output. Then, the lower-priority stored packet (currently the highest-priority packet) is forwarded to the output port.

An SDRAM controller monitors the number of buffers. If remaining number of buffers becomes one, then it saves the next input packet to the SDRAM automatically. If the number of busy buffers is one, then the SDRAM controller automatically restores the saved packets to buffers in order of their priorities.

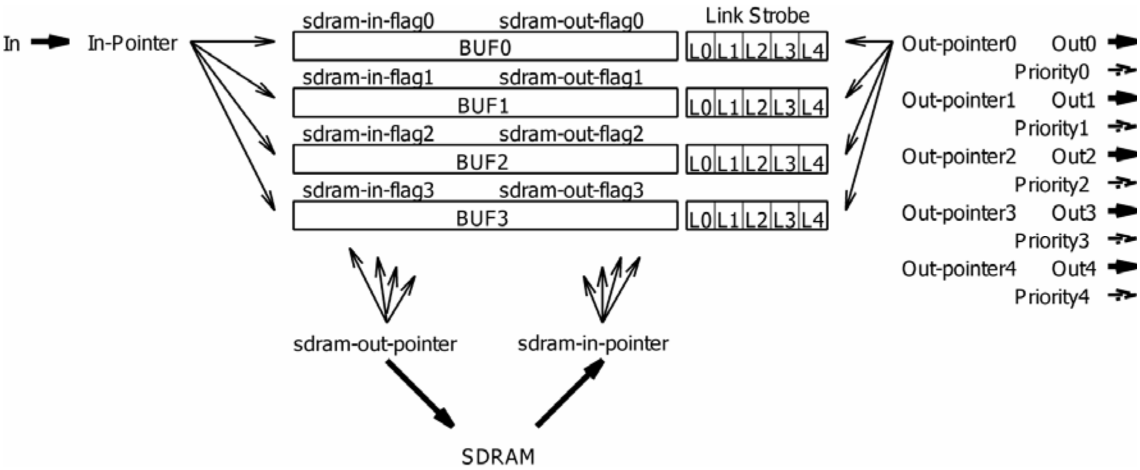

Figure D.2 - An overtaking buffer

Annex E
(informative)

E.1 An example of independent routing of data and event

Figure E.1 shows a typical arrangement of the *Responsive Link*, in which nodes are placed at every crossing of the grid and neighbouring nodes are connected by the *Responsive Link*. A source node sends a data packet with priority 0 to a destination node along the thick broken line. At the same time, the source node can send an event packet with priority 0 to the destination node along the thin dotted line. Event routes and data routes can be set independently, as shown in this figure.

E.2 An example of priority based routing

Figure E.1 also shows an example of priority-based routing. A data packet with priority 1 is transmitted along the thick black line. It is possible to set different routes for the same source and destination when the priorities are different. Similarly, an event packet with priority 3 can be routed along the thin black line. All data and event packets can be transmitted simultaneously.

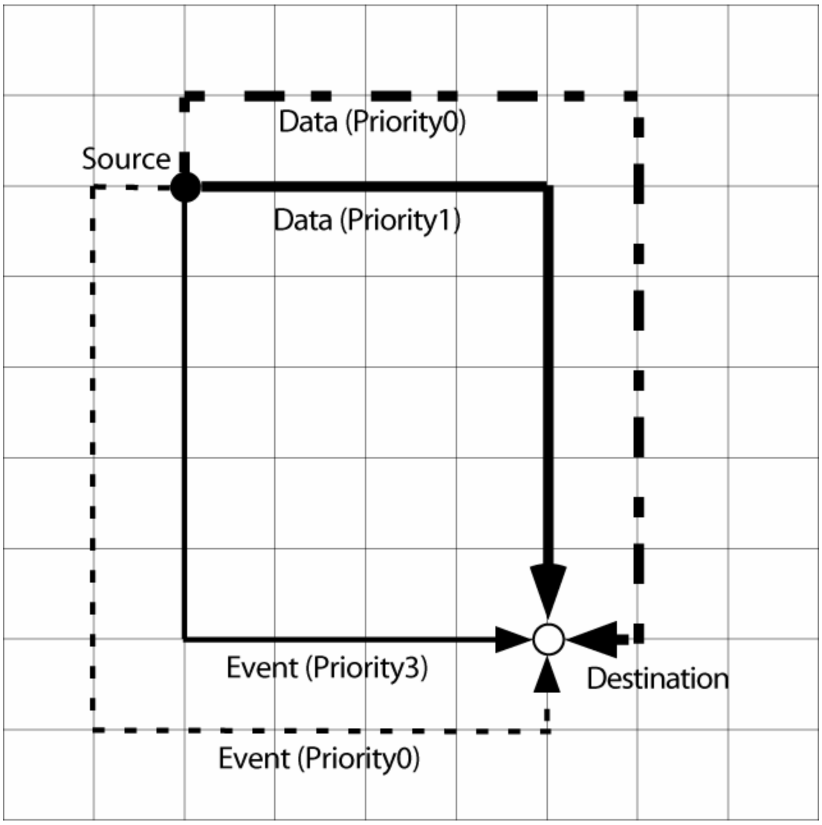

Figure E.1 - An example of routing