

# Serial RapidIO Robustness Enhancement Scheme

Yong Zhang<sup>1,2</sup>, Yong Wang<sup>1,2</sup>, Ping Zhang<sup>1,2</sup>, Kai Sun<sup>3</sup>

<sup>1</sup> Wireless Technology Innovation Institute, <sup>2</sup> Key Laboratory of Universal Wireless Communication, Ministry of Education Beijing University of Posts and Telecommunications, Beijing 100876, P. R. China

<sup>3</sup> College of Electronic Information Engineering, Inner Mongolia University, Hohhot 010021, P. R. China  
Email: xiazibin@gmail.com

**Abstract**—Robustness enhancement scheme in Serial RapidIO (SRIO) interconnect is proposed to overcome the performance degradation caused by noise and Electromagnetic Interference (EMI). The main idea of this scheme is the adaptive speed transition and mode conversion. Adaptive speed transition can improve average throughput and reduce delay in high Bit Error Rate (BER) environment. Mode conversion is to conquer frequent usage of feedback channel. Simulation shows that the scheme of combining adaptive speed transition with mode conversion leads great performance enhancement in SRIO network.

## I. INTRODUCTION

With the development of high performance embedded system, the demand of high-performance interconnect becomes critical, whereas conventional interconnects, such as Ethernet and Peripheral Component Interconnect (PCI), fail to meet such demand. RapidIO is one such emerging standard that provides high-performance interconnect for chip-to-chip, board-to-board, and chassis-to-chassis communications [1], [2]. But during the process of implementing SRIO between TMS320C6455, Virtex5 and Tsi578, we discovered that if the PCB fails to meet the explicit and implicit requirements of the SRIO specifications or the environment in which the SRIO works is tough, high BER will occur. Excessive packet losses will result in so many retransmissions, even congestions sometimes. To circumvent this problem, we define a Robust Transport Protocol (ROTP) to enhance the robustness of SRIO according to the characteristics of high speed switching signal.

## II. THE SRIO SPECIFICATION

RapidIO is a non-proprietary high-bandwidth system level interconnect. It is a packet-switched interconnect intended primarily as an intra-system interface at Gigabyte-per-second performance levels, and it has been optimized for embedded systems, particularly those which require multiple processing elements to cooperate [3], [4], [5]. Uses for the architecture can be found in connected microprocessors, memory, and memory mapped I/O devices that operate in networking equipment, memory subsystems, and general purpose computing.

RapidIO uses a three-layer architectural hierarchy, shown in Fig.1. The logic specifications, at the top of the hierarchy, define the overall protocol and packet formats. They provide the

This work was supported in part by Korean Electronics and Telecommunications Research Institute, China 863 Program with NO.2006AA01Z283, China Nature Science Fund and project with NO.60772113.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.  
CHINACOM 2010, August 25-27, Beijing, China  
Copyright © 2011 ICST 973-963-9799-97-4  
DOI 10.4108/chinacom.2010.55

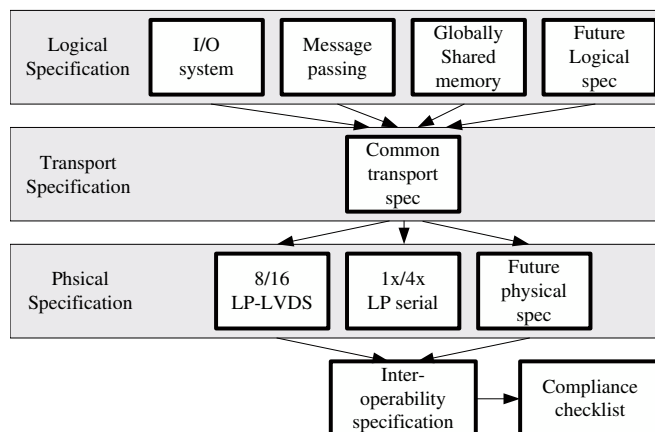


Fig. 1. RpaiddIO specification hierarchy.

information necessary for end points to initiate and complete transactions. The transport specification, on the middle layer of the hierarchy, defines the necessary route information for a packet to move from end point to end point. The physical layer specifications, at the bottom of hierarchy, contain the device level details, such as packet transport mechanisms, flow control, electrical characteristics, and low-level error management.

SRIO protocol exchanges packets and smaller quantities of link-specific information called control symbols and currently includes three frequency points: 1.25Gbps, 2.5Gbps, and 3.125Gbps. These fast switching signals generate a considerable amount of noise and radiation, which degrades system performance and creates EMI problems, since proper Printed Circuit Board (PCB) design for this interconnect requires highly experienced routing skills [6]. The most important characteristic of tough working environment and rough PCB is the lossy behavior, where BER increases as the working condition gets worse. As SRIO specification has limited contents to deal with such problems, robust transport algorithm/protocols are needed that can improve system performance.

Without any modifications to current SRIO specification, ROTP defines three Speed States and two modes of operation. Actually, the transmitter and receiver of ROTP can be shifted among different speed states in different modes. Robustness of SRIO can be significantly enhanced when ROTP used.

### III. ADAPTIVE SPEED TRANSITION

Our scheme redefines three frequency points of SRIO as shown in Table I. The transmitter and receiver can be shifted among different speed states. Performance of SRIO can be significantly enhanced with adaptive speed transition.

TABLE I  
FREQUENCY POINTS OF ROTP

Speed State	Speed Level	Baudrate	Environment Requirement
Low Speed	Low	1.25Gbps	Low
High Speed	High	2.5Gbps	High
Full Speed	Very High	3.125Gbps	Very High

Adaptive speed transition can be characterized as an interaction between two state machines, one transmitter machine and one receiver machine. Both the transmitter and the receiver have three speed states, which in many ways are related to each other. The transmission starts from the Low Speed, and tries to switch to highest possible state, or, stay in the Low Speed if the conditions are terrible.

Decisions about transitions between the various speed states are taken by the transmitter on the basis of:

- Packet-accepted control symbol (ACK) from the receiver.
- Packet-retry control symbol (NACK) from the receiver.

The receiver also starts in the Low Speed state and gradually transits to higher state according to the state machine of the transmitter.

Initially, while working in the Low Speed state, the receiver has not yet successfully received consecutive packets. Once consecutive packets have been accepted correctly, the receiver will inform the transmitter to directly jump to the Full Speed state by sending ACK control symbol. Then, the receiver changes its own speed state according to the speed state of the transmitter. Only upon repeated losses of packets will it inform the transmitter to go back to lower states by sending NACK. However, when that happens it first transits back to the High Speed state. There, reception of any number of consecutive packets is normally sufficiently to enable transition to the Full Speed state again.

During the transition of speed states, thresholds of adjacent speed states become the key of system performance tuning. For example, if the transitional threshold from the Low Speed to High Speed (L2H) is greater than the threshold from High Speed to Full Speed (H2F), the system then, is more likely to work in Full Speed under the condition that the BER is the same in different speed states. And it works the same way when the threshold from Full Speed to High Speed (F2H) is greater than the threshold from High Speed to Low Speed (H2L), the system has a better chance to work in Low Speed state as the working environment deteriorates.

### IV. MODE CONVERSION

The ROTP scheme has two modes of operation, called (O)ptimistic and (R)eliable mode. It is important to understand the difference between speed states, as described in the previous section, and modes. These abstractions are orthogonal

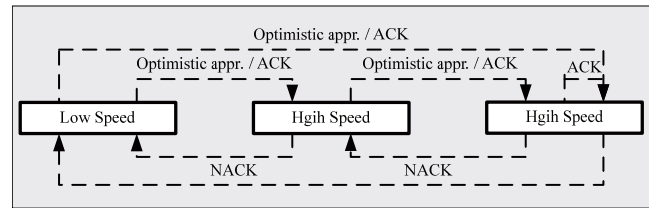


Fig. 2. Speed state machine in O-mode

to each other. The speed state abstraction is the same for all modes of operation, while the mode controls the logic of speed state transitions and what actions to perform in each speed state. The optimal mode to operate in depends on the characteristics of the environment of the ROTP, such as PCB routing quality, feedback abilities, error probabilities and distributions, etc. All ROTP implementations must implement and support both modes of operation.

#### A. O-mode

In O-mode, a feedback channel is used to send feedbacks from receiver to transmitter. O-mode aims to sparse usage of the feedback channel. The frequency of packet losses may be higher than for R-mode.

Fig.3 shows the state machine for the transmitter in O-mode.

The transition logic for speed states in O-mode is based on two principles: the optimistic approach principle and feedback.

Transition to a higher speed state in O-mode is carried out according to the optimistic approach principle. This means that the transmitter transits to a higher speed state when it is fairly confident that the receiver has received enough consecutive packets.

The O-mode makes use of feedback from receiver to transmitter for transitions in the backward direction and for optional improved forward transition.

NACKs will be sent from the receiver to the transmitter when the receiver found that some packets have been lost. Upon reception of NACKs the transmitter transits back to the lower speed state.

In addition to NACKs, ACKs may also be used in the O-mode. Upon reception of an ACK for consecutive packets, the transmitter knows that the receiver has received consecutive packets and the transition to a higher speed state can be carried out immediately. Transitions between receiver speed states are decided by the transitions of speed state in transmitter.

#### B. R-mode

The R-mode differs in many ways from the O-mode. The most important difference is a more intensive usage of the feedback channel and a stricter logic at both the transmitter and the receiver that prevents loss of packet between transmitter and receiver. Feedback is sent to acknowledge all packets. R-mode aims to maximize robustness against loss propagation and damage propagation. It may have a lower probability of packet loss and a lower BER than O-mode, but a larger number of NACK may be delivered.

The transition logic for transmitter states in R-mode has much in common with the logic of the O-mode except the optimistic approach principle, and the transition logic is based on two principles: the secure reference principle and NACKs.

The upwards transition is determined by the secure reference principle. The transmitter bases its confidence only on acknowledgments received from the receiver. This ensures that the synchronization between the transmitter and receiver will never be lost due to packet losses.

Downward transitions are triggered by NACKs. Note that NACKs should rarely occur in R-mode because of the secure reference used.

### C. Mode transitions

This subsection describes how the transitions are performed together with exceptions for the transmission and reception functionality during transitions.

#### a) Parameters for mode transitions

The transmitter and receiver maintain a variable whose value is the current mode for that context. The value of the variable controls, for the context in question, which control symbol to use, which actions to be taken, etc.

Parameters for the transmitter side:

**T\_MODE:** Possible values for the T\_MODE parameter are (O)ptimistic and (R)eliable. T\_MODE must be initialized to R.

**T\_TRANS:** Possible values for the T\_TRANS parameter are (P)ending and (D)one. T\_TRANS must be initialized to D. When T\_TRANS is P, it is required

- that mode information is included in packets sent, at least periodically,
- that the transmitter not transit to the Full Speed state,
- that new mode transition requests should be ignored.

Parameters for the receiver side:

**R\_MODE:** Same as that of T\_MODE.

**R\_TRANS:** Possible values for the R\_TRANS parameter are (I)ntialized, (P)ending and (D)one. R\_TRANS must be initialized to D. A mode transition can be initiated only when R\_TRANS is D.

#### b) Mode transition from O-mode to R-mode

An ACK(R) or a NACK(R) is sent to initiate the mode transition. The transition procedure is described in Fig.3.

#### c) Mode transition from R-mode to O-mode

Either the ACK(O) or the NACK(O) is used to initiate the transition from R-mode to O-mode and the transmitter must always run in the High Speed state during transition. The transition procedure is described in Fig.4.

During the transition of different mode, threshold plays a vital role in the process. For example, as the transitional threshold from R-mode to O-mode (R2O) gets higher, system is more likely to work in R-mode. And it works the same way when the threshold from O-mode to R-mode (O2R) increases.

## V. CONTROL SYMBOL DEFINITIONS OF ROTP

Our scheme essentially extends the current SRIO specifications by adding extra control symbols, which are about adaptive speed transition and mode conversion.

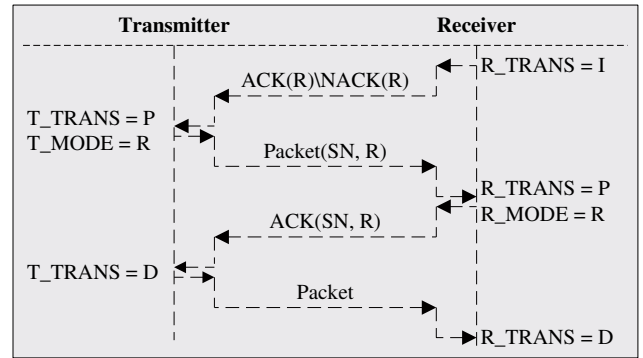


Fig. 3. Mode transition from O-mode to R-mode

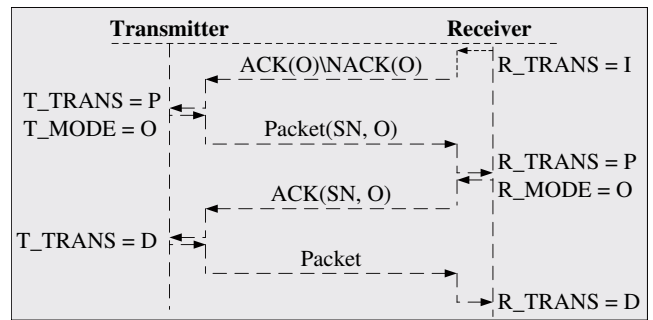


Fig. 4. Mode transition from R-mode to O-mode

Control symbols are the message elements used by end points connected by a SRIO link to manage all aspects of the SRIO link operation. Control symbols can carry out two functions, as shown in Fig.5, one encoded in the stype0 are 'status' functions that indicate status about the end point transmitting the control symbol. The functions encoded in stype1 are requests to the receiving end point. The fields parameter0 and parameter1 are used by the functions encoded in the stype0 field. The cmd field is a modifier for the functions encoded in the stype1 field.

In this section we explain the definition of control symbols in our scheme, which complete all the functions we mentioned above. These extra control symbols have been defined by extending the SRIO control symbols currently used. The transitions between speed states are initialized by transmitter and implemented by hardware. The mode conversion is launched by receiver and completed by software. The status of speed state can be acquired by sending speed state inquiring control symbol (encoded in stype1), then the receiver should reply

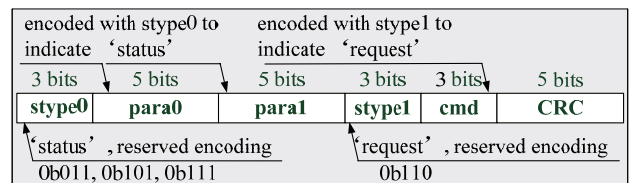


Fig. 5. Control symbol format

with corresponding speed state control symbol (encoded in stype0). Control symbol converting link mode (encoded in stype1) can be generated when the receiver wants to jump into another mode.

### 1) Extension of stype0 control symbol

Stype0 carries 'status' functions that convey status of the end point transmitting the control symbol. Adaptive speed transition redefines reserved encoding 0b011 of stype0 as speed state control symbol, in this case, parameter0 stands for packet\_ackID which means the ackID of the packet being ACKed and parameter1 stands for the status of speed state. Mode conversion uses reserved encoding 0b101 of styp0, when styp0 equals 0b101, parameter0 represents packet\_ackID and parameter1 represents current mode state, shown as Table II.

TABLE II  
PARAMETER1 ENCODING FOR MODE CONVERSION

Parameter	Value	Mode	Trans
parameter1[0-4]	0bxxxx0	R-mode	-
	0bxxxx1	O-mode	-
	0bxx00x	-	(D)one
	0bxx01x	-	(P)ending
	0bxx10x	-	(I)ntialized

Stype1 carries requests to receiver from transmitter and ROTP uses the reserved encoding 0b110. Field cmd contains the command and two commands are defined: speed state inquiring control symbol and mode conversion control symbol. The speed state inquiring control symbol causes receiver to return the status of its own speed state. When an end point accepts a mode conversion control symbol, it means its corresponding receiver wants to change to another mode.

## VI. SIMULATION RESULTS

We simulated our scheme using Opnet. We mainly focused on the effect of adaptive speed transition upon the average throughput and average delay, we also focused on the effect of mode conversion upon the amount of feedbacks. The RapidIO communication fabric consisted of one transmitter and one receiver. Average packet length is 1650 bits and packet interval time is 1 *us*. We assumed that the PCB layer does not satisfy all the physical requirements laid out by the SRIO specifications, so the BER of SRIO link is relatively high. BER of Low Speed is  $10^{-9}$  and BER of High Speed is  $10^{-8}$ , BER of Full Speed varies from 0 to  $10^{-4}$ .

The algorithm used by SRIO link to compute the number of bit errors in a packet utilizes the formula below [7].

$$P_k = p^k \cdot (1 - p)^{N-k} \cdot \binom{N}{k} \quad (1)$$

$P_k$  is the probability of exactly  $k$  bit errors occurring in a segment of length  $N$ ,  $p^k$  is BER. The number of bit errors  $k$  in packet of  $N$  bits with a bit error probability  $p$  follows a binomial distribution.

First, we simulate the impact of adaptive speed transition on both SRIO's average throughput and delay. Being said, the four

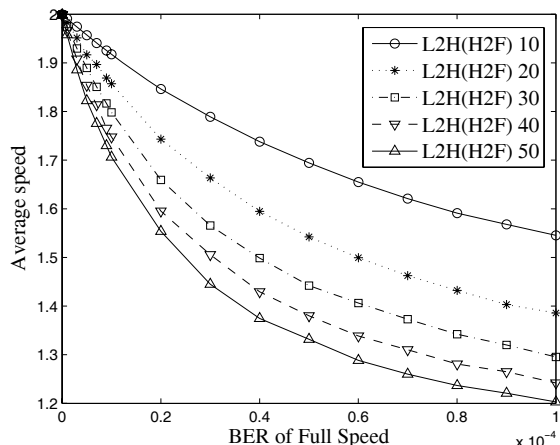


Fig. 6. Average speed in different cases

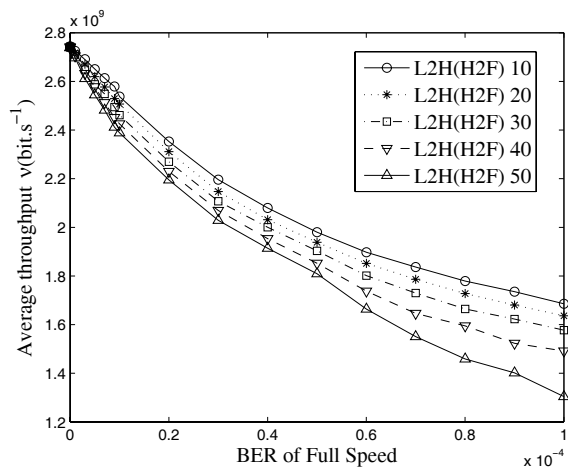


Fig. 7. Average throughput in different cases

thresholds between different speed states are the key of system performance. At this time, we fix the down shift threshold (F2H, H2L) to 1, which means once packets loss happens, the speed state goes down accordingly. We also assume the up shift thresholds (L2H, H2F) are all the same and scenarios with different values are simulated.

Fig.6 shows system average speed rate under different upshift threshold, where Low Speed is set to 0, High Speed and Full Speed state are set to 1 and 2 respectively. Greater system speed means longer time in higher speed state. For example, the speed rate is 2 when BER is 0, which means system runs in Full Speed at all time. As BER increases, system begins switching between these three states, so the average speed decreases in respond. The higher the upshift threshold is, the harder it is for system to switch to higher speed state. But on the other hand, system is more flexible and more adaptive to the real environment if all thresholds are small.

Fig.7 is the system throughput corresponding to Fig.6, whereas Fig.8 is the system delay. Average system throughput

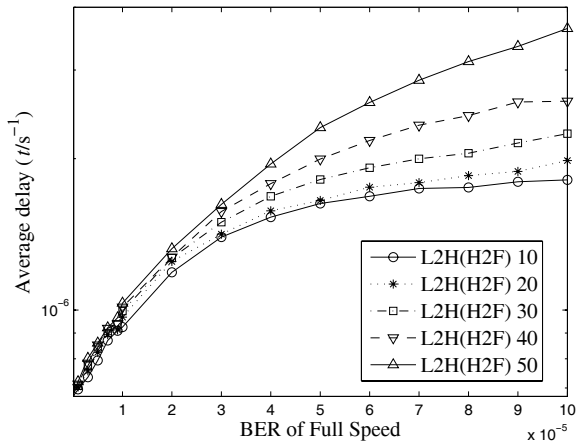


Fig. 8. Average delay in different cases

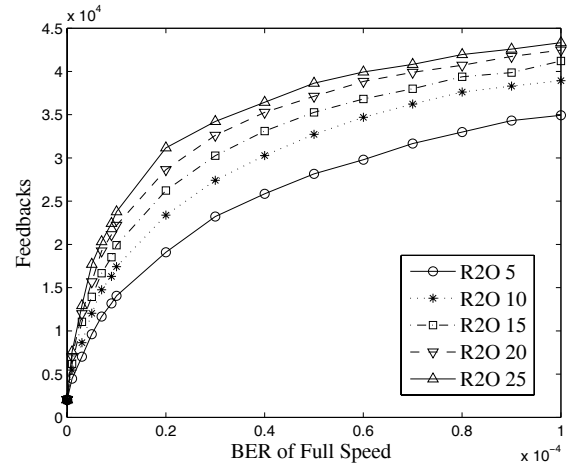


Fig. 10. Average feedbacks in different cases

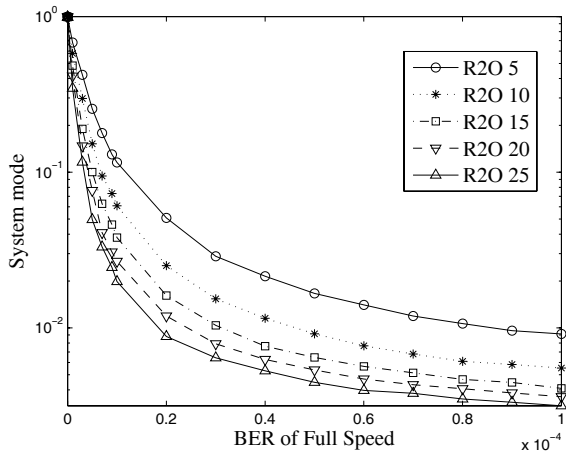


Fig. 9. System mode in different cases

in adaptive speed transition cases are much higher for adaptive speed transition can adjust the baudrate intelligently according to the environment. Working with fixed baudrate under high BER needs much more retransmission, so average throughput in this case is lower and latency is larger.

Then we simulate the impact of mode conversion on SRIO feedback channel. As said before, the threshold R2O and O2R are critical point to system performance. Here we simulate scenarios with different R2O values. In this simulation, once packet losses are detected, the receiver leaves O-mode, which means setting O2R to 1. After R2O continuous packets, the receiver re-enters O-mode.

With different threshold R2O, how mode conversion occurs according to above scheme is shown in Fig.9, where R-mode and O-mode are assigned to be 1 and 2 respectively. Larger system mode indicates longer time in O-mode. For example, system mode is 1 when BER is 0, which means system runs in O-mode at all time. As BER increases, system begins shifting between two modes, so system mode decreases in respond.

Fig.10 is the corresponding average system feedbacks of

Fig.9. Mode conversion can, base on real environment, reduce usage of the feedback channel as much as possible without compromising system throughput. As a result, the greater the threshold R2O is, the more difficult it is for system to enter O-mode, then the amount of feedbacks will increase.

## VII. CONCLUSIONS

In order to solve the problem we encountered in real project, we employ adaptive speed transition and mode conversion to improve the intelligence and robustness of SRIO. Simulation results indicate that ROTP can overcome the severe existing circumstances by dynamically shifting among different speed states and modes. Further research will focus on implementing our scheme in real project and make a proper evaluation of it.

## REFERENCES

- [1] D. H. Martin McKenny, Julian A.B Dines, "Transporting multiple classes of traffic over a generic routing device - an investigation into the performance of the rapidio interconnect architecture," in *Networks, 2003. ICON2003. The 11th IEEE International Conference on*. Sydney: IEEE, 28 Sept.-1 Oct. 2003, pp. 39-44.
- [2] H.-b. W. Q.-z. Zhang, Juan; Su, "Research and implement of serial rapidio based on mul-dsp," in *Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on*. Wuhan: IEEE, 11-13 Dec 2009, pp. 1-4.
- [3] D. Bueno, A. Leko, C. Conger, I. Troxel, and A. D. George, "Simulative analysis of the rapidio embedded interconnect architecture for real-time, network-intensive applications," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*. Tampa: IEEE, 16-18 Nov. 2004, pp. 710-717.
- [4] X. Zhang, M. Gao, and G. Liu, "A scalable heterogeneous multi-processor signal processing system based on the rapidio interconnect," in *Proc. International Symposium on Intelligent Information Technology Application Workshops IITAW '08*. Zhang Jiajie: IEEE, 21-22 Dec. 2008, pp. 761-764.
- [5] X. Zhang, G. Liu, and M. Gao, "A high-performance scalable computing system for real-time signal processing applications," in *Proc. Congress on Image and Signal Processing CISP '08*. Sanya: IEEE, 27-30 May 2008, pp. 556-560.
- [6] B. Young, "Enhanced lvds for signaling on the rapidiominterconnect architecture," in *Electrical Performance of Electronic Packaging, 2000, IEEE Conference on*. New Jersey: IEEE, 23-25 Oct. 2000, pp. 17-20.
- [7] *Network Simulator*, Opnet Technologies, 2003. [Online]. Available: <http://www.opnet.com>