RapidIO Network Enumeration Strategy Based on Minimum Isolation Block

Congyue Yin*, Xingming Zhang and Shuai Wei
National Digital Switching System Engineering & Technological Research Center
NDSC, Zhengzhou, China

*503637088@qq.com

Abstract. A enumeration strategy based on minimum isolation block is proposed aiming at the enumeration problem of RapidIO network. Firstly, the minimum isolation block is first described. Secondly, the module is divided into the routing network and a depth-first traversal search of the entire RapidIO network is performed. Thirdly, to compare the average operation time, power consumption value and energy consumption value of the RapidIO network before and after isolation better, the flow driven mechanism is introduced. The experimental results show that the method of minimum isolation block can avoid repeated enumeration, reduce the enumeration number of switches and make the performance of RapidIO network better.

1. Introduction
In recent years, with the rapid development of the Internet, microprocessors and memory, RapidIO has become increasingly demanding in the embedded device market as a system interconnection technology [1]. RapidIO industry associations are open, and many types of equipment vendors have all been supportive of the embedded market, and in this environment this openness is particularly well suited [2].

With the deepening of RapidIO research, the researchers are turning from the RapidIO protocol itself to the more complex RapidIO routing network system, and enumeration as a unique search method of the system is worthy of being studied. In [3], the enumeration process of RapidIO system network is given in detail. In [2], the corresponding enumeration instance is given, and the two are based on RapidIO, and consists of one switch, two system hosts, one High-speed I/O card and a DSP array combination of the framework of the system operating mechanism to gradually elaborate, but did not further describe how to optimize the RapidIO routing network. In [4], a method of networking for recursive enumeration of RapidIO networks is proposed. It adopts the path-weighting method based on depth-first. If only the depth-first enumeration algorithm is adopted, the optimal path can not be obtained This method does play a role in solving such problems, but for more complex ring topologies, star topology, star topology and other types of topologies are not necessarily applicable.

In order to solve the problem of RapidIO routing network enumeration, it is necessary to optimize the RapidIO routing and composition structure and simplify the complex routing network. The literature network of RapidIO fat tree structure in [5] is extended horizontally and vertically, and take the appropriate integration and optimization measures, the nature and structure of the same routing network combined into a module, said each module is the smallest isolation block, in order to reduce
the number of enumerations, while avoiding repeated enumeration. In order to better prove that the above ideas can optimize the RapidIO routing network, it will introduce the flow-driven mechanism, calculate the corresponding energy consumption according to the flow rate, and reduce the energy consumption before and after the route network simplifies the reliability of the above ideas.

2. Flow drive mechanism description and definition

Based on the upper section of the minimum isolation block description, in order to better define the minimum isolation block, the following figure will be used to illustrate the method to determine the minimum isolation block. Figure 1 shows a RapidIO routing topology, which includes a RapidIO host and three RapidIO devices, and the RapidIO switch group is an $2n \times 2n$ array. To the RapidIO host connected to the dashed box, for example, RapidIO switches 1, 2, 3, 4 the same structure and nature, we can combine the four switches to form a module, and said the module is the smallest isolation block, similarly, the rest of the RapidIO switches for similar isolation block division, and ultimately can be shown in Figure 2 RapidIO routing topology simplified map.

![Figure 1. RapidIO routing topology of the $2n \times 2n$ switch array.](image1)

![Figure 2. RapidIO simplified routing topology of the $2n \times 2n$ switch array.](image2)
The position of each RapidIO switch in the tree topology is set to coordinates \((i, j)\), and \(i, j \in \{0, 1, 2\}\), where the number of the switch \(i\) is represented in descending order, and the number of the switch \(j\) is represented in the order from left to right. When \(i > 0\), \(j\) within the range of the interval \([0, \frac{k^2}{2} - 1]\); When \(i = 0\), \(j\) within the range of the interval \([0, \frac{k^2}{4} - 1]\), which \(k\) represents the number of switches.

In this paper, we will build a model of the power consumption of the entire RapidIO tree network based on the power model of a single RapidIO switch. Equation (1) shows that the number of line cards, line card type, backplane type, and power consumption of the switch network interface are taken into account when establishing a single switch power model [6]. Equation (2) represents the composition of the RapidIO routing network energy consumption, which \(P_{i,j,k}^{port}\) represents the power consumption of the \(k\) port of the switch, which \(P_{i,j}^{fix}\) represents the fixed power consumption of the switch.

\[
\text{Power}_{\text{switch}} = \text{Power}_{\text{chassis}} + \text{num}_{\text{linecards}} \times \text{Power}_{\text{linecards}} \\
+ \sum_{k=0}^{\text{configs}} \text{numports}_{\text{config}} \times \text{Power}_{\text{config}} \times \text{utilizationFactor}
\]

\[
E = \sum_{i,j,k} \sum_{0 \leq j < \frac{k^2}{2}} (P_{i,j}^{fix} \cdot t_{i,j} + \sum_{k=0}^{\text{configs}} P_{i,j,k}^{port} \cdot t_{i,j,k}) \\
+ \sum_{i,j,k} \sum_{0 \leq j < \frac{k^2}{4}} (P_{i,j}^{fix} \cdot t_{i,j} + \sum_{k=0}^{\text{configs}} P_{i,j,k}^{port} \cdot t_{i,j,k})
\]

The RapidIO host transfer rate is the same as the switch port forwarding rate, denoted by \(r\). Assume that there are two streams in the RapidIO network topology, denoted as \(f_1\) and \(f_2\), where the size of \(f_1\) is recorded as \(m_1\), the size of \(f_2\) is recorded as \(m_2\), and \(m_1 > m_2\) [7].

The main purpose of the routing network energy model is to achieve the equation (3).

\[
\min E
\]

In the routing network energy consumption model, the following equation can be used as restrictions.

\[
y_{i,j,k} = \sum_{x} \frac{S_{i,j,k}(x) \cdot f(x)}{r_{i,j,k}}
\]

\[
\sum_{i \in \rho} \sum_{j \in \rho} S_{i,j,k}(x) \leq 2, x \in \rho
\]

\[
\text{Path} = \left\{(i, j, k, x) \in \left[\begin{array}{c} 0 \leq j < \frac{k^2}{2} \cap S_{i,j,k}(x) = 1 \end{array}\right], x \in \rho\right\}
\]

\[
t_{i,j,k}^{'} = \max\{y_{i,j,k} / i, j, k \in \text{Path}, x \in \rho\}
\]

\[
t_{i,j,k} = \max\{t_{i,j,k}^{'} / x \in \rho\}
\]
\[ t_{i,j} = \max\{t_{i,k}, k \in K\} \]  

(9)

3. Core algorithm design

3.1. Depth-first Traversal Algorithm

In the RapidIO routing network topology, the depth of the first traversal algorithm is generally used, the specific algorithm flow shown in Table 1.

**Table 1.** RAPIDIO depth first traversal algorithm flow.

| Depth-first traversal algorithm |
|-------------------------------|
| 0. Set the switch SWITCH M connected to the host HOST as the starting point. |
| 1. Make sure that the switch M terminal is connected to the switch SWITCH N or the terminal EndPoint. |
| 2. If the switch SWITCH N is connected, it is judged whether the switch N has been accessed. |
| 3. If switch N is not accessed, mark N as visited and continue the depth-first search from point N. |
| 4. If switch N has been accessed, only the connection is established with switch N. |
| 5. If the terminal EndPoint is connected, the terminal will establish a relationship with the switch M. |
| 6. If any of the neighboring switch nodes around SWITCH have been accessed, they are returned in reverse order from the last visited node until the initial vertex. |
| 7. Terminate the algorithm when all SWITCH vertices have been accessed, or from any SWITCH vertices that have been accessed, and can not reach the SWITCH vertices that have not been accessed. |

3.2. The Minimum Isolation Block Generation in Routing Topology

According to the above contents, the fat tree structure of RapidIO in [5] is extended. In general, the structure of fat tree consists of the core layer, the convergence layer and the edge layer from top to bottom, where the portion of the core layer does not change, and the layer of the converging layer and the edge layer extends longitudinally to obtain a layer as shown in Fig 3. RapidIO fat tree topology expansion map. In order to reduce unnecessary energy consumption, there are 16 link-related RapidIO switches are open, the rest of the switches are closed. The link \( f_1 \) is from the host \( h_0 \) to \( h_4 \) and the link \( f_2 \) is from the host \( h_2 \) to \( h_6 \). In the case of link \( f_1 \), in general, to achieve access to the path from the host \( h_0 \) to \( h_4 \), you need to follow the depth of the previous traversal algorithm to complete the various RapidIO switch nodes between the mutual search and enumeration. However, according to the previous section, the RapidIO switches \((0, 0), (0, 1), (1, 0)\) and \((1, 1)\) can be combined into a minimum isolation block, the switches \((2, 0), (2, 1), (3, 0)\) and \((3, 1)\) form the second smallest isolation block, and the rest of the RapidIO switches are the same. And then in the local, respectively, search in each of the minimum isolation block depth of the first priority traversal; on the whole, but also to the isolated block as a unit, search in the isolation between the depths of the first priority traversal. It can greatly reduce the number of enums RapidIO switches, but also to avoid repeated enumeration.
3.3. The Flow Driven Energy Consumption Algorithm

The purpose of this algorithm is to evaluate the performance improvement of the system by calculating the energy consumption value.

**Table 2.** Flow driven energy consumption algorithm flow.

| Flow-driven energy consumption algorithm |
|------------------------------------------|
| **Input:** FT: RapidIO routing network structure  |
| TR: flow matrix  |
| **Output:** PATH: the path to each traffic allocation  |
| E: total energy consumption of the RapidIO routing network  |
| P: total power consumption of the RapidIO routing network  |
| TIME_AVE: average working time of the RapidIO switch  |

0. Topology()
1. Traffic()
2. while(TR)
3. {
4. for $i \in$ TR
5. FindPath(FT,src,des,P)
6. E +=Energy_Cost()
7. }
8. end while

3.4. The Flow Path Allocation Algorithm

In the flow path allocation issue, we mark the assigned flow and the number of flow through each port. When assigning a path to a new flow, the algorithm selects the path based on the status of each port. If the number of flow flowing through each port is less than the maximum number of flow on the port, the path will be assigned to that flow.
| Table 3. Flow path allocation algorithm flow. |
|------------------------------------------------|
| Flow path allocation algorithm               |
| 0. FindPath(FT, src, des, P)                  |
| 1. {                                           |
| 2. switch_port_flag[][]                      |
| 3. traffic_flag[][]                           |
| 4. flow_num                                  |
| 5. if src/POD==des/POD                        |
| 6. for the port the flow passes && switch_port_flag[][] <flow_num |
| 7. write the subscript information of the port to P |
| 8. else                                       |
| 9. for the port the flow passes && switch_port_flag[][] <flow_num |
| 10. write the subscript information of the port to P |
| 11. traffic_flag[src][des]=1                  |
| 12. switch_port_flag[] +=1                    |
| 13. }                                         |

4. Experiments and results analysis

4.1. Experimental Setup
The experiment is in the VxWorks embedded operating system to complete the simulation. The simulation experiment will be carried out with reference to the following actual contents: The RapidIO embedded interconnects system will use the MPC8548 processor [8] as the host device at the endpoint. The switching device will use the Tsi578 chip [9]. So far, the RapidIO protocol has been updated to version 4.0, but in the actual operating environment is still using the 2.0 version of the RapidIO protocol. The experiment will be based on Figure 3. Set the RapidIO default interface rate to 2.5GB/s. The flow size meets the average distribution of 64MB index. RapidIO switch fixed power consumption is 100W. The switch port frequency can be configured to 1.25GB, 2.5GB and 3.125GB [10]. The port is set to 1 × mode.

4.2. Analysis of Experimental Results
In order to better verify the optimization of the minimum isolation block to RapidIO routing network, this paper will start from the aspects of RapidIO routing network awareness flow and RapidIO host awareness flow, compare the overall energy consumption, power consumption and switch of RapidIO routing network before and after use Average working hours. The so-called routing network stream of consciousness, that is, the maximum transmission rate of a single traffic is limited by the available bandwidth of the network, and the so-called host stream of consciousness, that is, the maximum transmission rate of a single traffic is limited by the host forwarding rate [8].

(a) RapidIO Routing Network Awareness Flow
RapidIO host forwarding rate and switch transfer rate are set to 2.5Gbps. The experimental results shown in Figure 4. Experiments show that the average energy consumption of the RapidIO routing network is reduced by about 9.7% and the average power consumption is reduced by about 8.6% after using the isolation block. The average working time of the RapidIO switch is not much different before and after the isolation, but it is also reduced.
(b) RapidIO Host Awareness Flow

RapidIO host forwarding rate is set to 2.5Gbps, the switch transfer rate is set to 3.125Gbps. The experimental results shown in Figure 5. Experiments show that, after using the isolation block, RapidIO host awareness of the overall energy consumption of the overall reduced by about 7.9%, power consumption decreased by about 12.3% on average, and RapidIO switch the average working time before and after isolation is almost the same, the difference is small, but also slightly reduced.

5. Conclusion

In this paper, the concept of minimum isolation block is proposed for the problem of RapidIO routing network enumeration. Based on this, the flow driving mechanism is introduced to evaluate the performance of RapidIO routing network. Although the mathematical model based on the isolation block is established and the performance evaluation of the model is carried out with flow, the traffic congestion may be caused by the merging of the ports at the interface of the isolation block. However,
the main purpose of this paper is the use of the idea of the minimum isolation block, the final reduction of the number of instances of the RapidIO switch and the avoidance of repeated enumeration when the RapidIO routing network performs a depth-first traversal. Flow driving is only a mechanism of judgment that proves the reliability of thought, so the problem does not have any effect, but it can be used as a further direction for future work.

Acknowledgements
This work is partially supported by the National Science Technology Major Project (2016ZX01012101), the National Natural Science Foundation of China (61572520, 61521003). Any questions please contact Yin Congyue with 15237124973 or 503637088@qq.com.

References
[1] S. Baymani, K. Alexopoulos, and S. Valat. "Exploring RapidIO technology within a DAQ system event building network," IEEE Transactions on Nuclear Science, IEEE, Sept. 2017, pp.1-4, doi: 10.1109/RTC.2016.7543124.
[2] S. Fuller, RapidIO: The Embedded System Interconnect. John Wiley & Sons, 2005.
[3] RapidIO Interconnect Specification Annex1: Software / System Bring Up Specification, Rev. 4.0, http://www.rapidio.org, 2017.06.
[4] H L. Zhong, G C. Li, and G. Gao. “A recursive enumeration method for RapidIO networks”. China, Apr. 2014, 103746910.
[5] L. Shu, J. Hao, and Y F. Song. "Optimal many-to-many personalized concurrent communication in RapidIO-based fat-trees." Ieee / acis International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel / distributed Computing IEEE Computer Society, June.2016, pp.343-350, doi: 10.1109/SNPD.2016.7515923.
[6] Z H. Guo, S F. Hui, and Y. Xu. "Dynamic flow scheduling for Power-efficient Data Center Networks." IEEE / ACM, International Symposium on Quality of Service, IEEE, June. 2016, pp.11200-11214, doi: 10.1109/ACCESS.2017.2716345.
[7] Y F. Shang, D. Li, and M W. Xu. "Greening data center networks with flow preemption and energy-aware routing." Local & Metropolitan Area Networks, IEEE, Apr. 2013, pp.1-6, doi: 10.1109/LANMAN.2013.6528281.
[8] L. Wang and J. GONG. “The realization of serial RapidIO bus driver under VxWorks”, Mechanical and Electrical Product Development and Innovation, vol.28(6), Dec. 2015, pp.117-119, doi: 10.3969/j.issn.1002-6673.2015.06.042.
[9] Y L. Xian, X M. Ji, and X Y. Wang. “RapidIO interconnect technology based on TSI578”, Electronics and Packaging, vol.11, Dec. 2014, pp:26-30, doi: 10.3969/j.issn.1681-1070.2014.11.007.
[10] A. KLILOU, S. BELKOUCH, and A. ELMAIZI. "Case studies of data traffic management on a high-performance computing system based on multi-DSPs and Serial RapidIO interconnect." International Conference on Information Technology for Organizations Development, IEEE, Apr. 2016, pp.1-6, doi: 10.1109/IT4OD.2016.7479324.