Stabilizing Path Modification of Power-Aware On/Off Interconnection Networks

Jose Miguel Montañana[1], Michihiro Koibuchi[1],
Hiroki Matsutani[2], and Hideharu Amano[3]

[1] National Institute of Informatics
2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo,
JAPAN 101-8430
koibuchi@nii.ac.jp

[2] The University of Tokyo
RCAST, The University of Tokyo, 4-6-1
Komaba, Meguro-ku, Tokyo JAPAN 153-8904
matutani@hal.rcast.u-tokyo.ac.jp

[3] Keio University
3-14-1, Hiyoshi, Kohoku-ku, Yokohama, JAPAN 223-8522
hunga@am.ics.keio.ac.jp

Abstract

Power saving is required for interconnects of modern PC clusters as well as the performance improvement. To reduce the power consumption of switches with maintaining the performance, on/off link regulations that activate and deactivate the links based on the traffic load have been widely developed in interconnection networks. Depending on which operation is selected, link activation or deactivation, the available network resources are changed, thus requiring paths to be reconfigured. To maintain deadlock freedom of packet transfers, connectivity, and performance during the path changes, we propose to apply dynamic reconfiguration techniques that process packet transfer uninterruptedly to power-aware on/off interconnection networks. The dynamic network reconfiguration techniques stabilize the update of paths that are quite crucial to use power-aware on/off link techniques in interconnects of PC clusters. We investigate the performance and behavior of network reconfiguration technique as soon as the link activation or deactivation occurs. Evaluation results show that the simple dynamic reconfiguration techniques slightly reduce the peak packet latency and reconfiguration time of the change compared with existing static reconfiguration in on/off interconnection networks. A reconfiguration technique called Double Scheme reduces by up to 95% the peak packet latency caused by the on/off link operation.

978-0-7695-4134-1/10 $26.00

1 Introduction

Lossless interconnection networks, such as InfiniBand[7] and Myrinet[13] have been widely developed for high-performance PC clusters. As computation power is increased, the network bandwidth is increased in the PC clusters. For example, high-throughput (non-blocking) commercial InfiniBand switches are now available, and its link bandwidth has rapidly increased, such as SDR (2.5Gbps)/DDR(5.0Gbps) and link speed x1,4,8, and 12. The power consumption of interconnects is increased as the link bandwidth is improved in PC clusters. Thus, the low-power techniques of interconnects have become one of the more important research topics for building PC clusters.

The power-aware routing techniques using DVFS (dynamic voltage and frequency scaling) have been discussed[18][20] in interconnection networks. In addition to the techniques using the DVFS, on/off link regulation techniques that activate and deactivates the links based on the traffic load have been developed and evaluated for the power saving of switches[1][9][19]. The links consume a large amount of power even if no data is transferred, and its power is almost constant regardless of the traffic load. For example, we measured that the link deactivation decreases approximately 1.2W and 2.1W per port of 1000Base-T in the Dell PowerConnect 5324 and 6248 Gigabit-Ethernet switches with totally 15W and 57W, respectively, when all ports are shutdown[9], and 0.9W per port with DDR, $\times 4$ speed in an InfiniBand 24-port switch, SFS7000D-SK9, with the totally 43W.

Depending on which operation is selected, link activation or deactivation, the available network resources are

changed, thus requiring paths to be reconfigured. The network reconfiguration that modifies paths with introducing no deadlocks of packets are required to use the on/off link regulation techniques, since the existing routing function is replaced by one that provides reach-ability between any pair of hosts during the changes. However, there has been little work on the network reconfiguration for on/off link regulations.

In this paper, we propose to apply a dynamic network reconfiguration technique in order to stabilize the path modification for on/off link regulations. During the changes, the dynamic network reconfiguration processes the packet transfer uninterruptedly, and the performance is slightly decreases in short term after the link status is changed.

The dynamic network reconfiguration is originally intended to increase the fault-tolerance, and there are its evaluation results when a link fails[11]. However, in the case of on/off link regulations, its behavior after the link is reactivated is not analyzed, but quite important, because it is a throughput-sensitive case when a traffic load is increased. Thus, we investigate its performance and behavior.

The rest of this paper is organized as follows. In Section 2, we briefly introduce the existing on/off link regulation techniques, and routing functions. In Sections 3, we propose to apply the dynamic network reconfiguration to the on/off link regulation techniques. In Section 4, we evaluate the dynamic network reconfiguration through the simulations under variable traffic loads. Our conclusions are in Section 5.

2 On/Off Link Regulations

2.1 Outline

On/off link regulation techniques reduce the power consumption of switches by deactivating links with maintaining the performance. All links are dynamically activated when the traffic load is high, while a large number of links are deactivated when the traffic load is low. Depending on which operation is independently selected, link activation or deactivation, the available network resources (switch and links) are changed. The existing routing function is thus replaced by one that provides network connectivity using the available network resources.

In the on/off link regulation techniques, the details of the procedure when a link is activated or deactivated is as follows.

Link Status: On to Off

1. Determine the path set that avoids the target link using a new routing algorithm.
2. Perform the network reconfiguration that updates the routing function of each path.
3. Inactivate the target link.

Link Status:Off to On

1. Determine the path set that uses the target link using a new routing algorithm.
2. Activate the target link.
3. Perform the network reconfiguration.

To avoid using deactivated links, existing fault-tolerant routing algorithms, and existing routing algorithms for irregular topologies can be applied as old and new routing function, and they provide deadlock freedom of packet transfers[19].

2.2 Path Modification

PC clusters usually employ lossless interconnection networks that include Ethernet with IEEE 802.3x link-level flow control, and they require deadlock-free packet transfers[15]. However, during the path change, the deadlock handle is difficult, because old and new routing functions work together in the network.

To avoid such a deadlock, the simple network reconfiguration limits the set of old and new routing functions so that they follow the same routing restriction for providing deadlock freedom. Figure 1 shows its example, and the old and new routing functions use the west-first turn model[6] that forbids the packet transfers from north/south/east to west direction in 2-D mesh in order to guarantee deadlock freedom. Even during updating a routing function, all packets will arrive at the destinations, because both old and new routing functions, that allow the same path set, send packets toward to the destination at intermediate routers. However, the network reconfiguration can only deactivate a limited link set, although each link should be dynamically and flexibly deactivated or activated according to the traffic load in order to make the best use of traffic access locality of parallel applications. In the example of the west-first turn model on 2-D mesh in Figure 1, all horizontal links, which are 50% of all links in the network, cannot be deactivated in order to maintain the network connectivity, because of its routing restriction.

To activate and deactivate all the links, a general network reconfiguration scheme can be used. A simple general network reconfiguration is STatic reconfiguration (ST). No packets routed according to the new routing function are allowed to be injected into the network while there still are packets in the network according to the old routing function[11]. The ST thus allows to use any combination

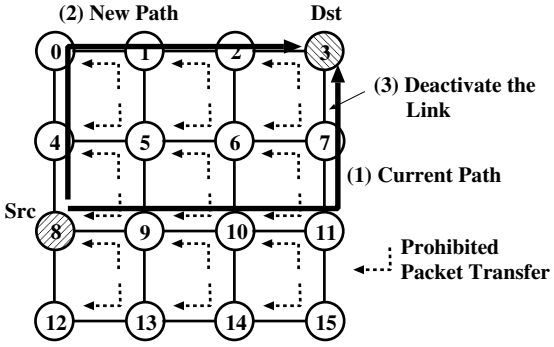


Figure 1. Simple Network Reconfiguration using West-First Turn Model

of old and new routing function which make the link set more activated or deactivated dynamically.

However, because of its serialization in performing reconfiguration, the ST leads to the large packet latency, and a large number of dropped packets which could be unacceptable for parallel applications.

ST can be implemented in two different ways:

- The injection of packets in the network is stopped, and the existing packets keep moving until the network is completely drained. After the new routing function is updated, the network traffic injection is resumed.
- Another implementation uses checkpoints. The injection is stopped in the same manner as previous way, however all the packets in the network are dropped by the routing update. The running applications continue from their last checkpoint after routing function is updated.

3 Dynamic Reconfiguration Mechanism for On/Off Link Regulations

In this section, we propose to apply general dynamic reconfiguration in order to avoid deadlocks and disconnection of the network during the topology change by the operation of the on/off link regulations.

3.1 Dynamic Reconfiguration

Reconfiguration techniques can be either static or dynamic. Static reconfiguration techniques require to completely stop the traffic in the network before any routing update as shown in the last section, thus the network is emptied[3, 12].Static reconfiguration largely impacts on packet latency, due to network down-time, i.e., halting packet injection, which could cause a strong performance degradation during the reconfiguration process.

On the other hand, in a dynamic reconfiguration the transition from one routing function to another is performed while the functional parts of the network are fully operational, i.e., no network down-time and not halting packet injection. The problem in this approach resides in the fact that, in general, two different and individually dead lock-free routing functions may be prone to deadlocks if they coexist in the network. This means that, in a dynamic reconfiguration, there will be a transition phase between the old and new routing functions where reconfiguration-induced deadlocks may occur. Another drawback when using dynamic reconfiguration is that it usually requires extra resources.

In the last decade, several dynamic reconfiguration mechanisms have been proposed. The Immunet mechanism [17] tries to minimize the impact of the reconfiguration process on the performance of the system, at the expense of providing a specific hardware support, which prevents it from being used on current commercial interconnects.

In [3], a Partial Progressive Reconfiguration (PPR) technique is proposed, allowing arbitrary networks to migrate between two instantiations of *up*/down** routing. The effect of load and network size on PPR performance is evaluated in [4]. Another approach is the *NetRec* scheme [14] which requires every switch to maintain information about switches some number of hops away. Yet another approach is the *Double Scheme* [16] (DS), which uses two sets of virtual channels in the network which act as two disjoint virtual network layers during the reconfiguration. The basic idea is first to drain one virtual network layer and reconfigure it while the other is fully up and running, then to drain and reconfigure the other virtual network layer while the first is up and running. A methodology for deriving new reconfiguration processes for any given pair of old and new routing function is given in [12]. An orthogonal approach which may be applicable on top of all of the above techniques is described in [10], where, for *up*/down** routing, only parts of the network (i.e., the "skyline") need to be reconfigured on a network change. Solid theoretical support on which dynamic reconfiguration design methodologies and techniques are proved deadlock-free can be found in [5].

The reconfiguration is originally designed for tolerating topological changes, for purposes as optimizing routing performance or increasing the fault tolerance of interconnection networks[11][2]. We use it for both cases where the network resources are increased or decreased by the operation of the on/off link regulations.

In this paper we use three different mechanisms, one static and two dynamic reconfiguration mechanisms. Although there are some dynamic reconfiguration mechanisms as introduced above, in this paper we consider the DoubleScheme (DS) [16], and the Simple Reconfiguration

(SR) [12] in the next subsection. This is because they are more flexible, in the sense that it can handle any topology and perform a transition between any pair of deadlock-free routing functions. However, they may require different support, like the presence of two sets of virtual channels for the DoubleScheme, and support control tokens for the SimpleReconfiguration.

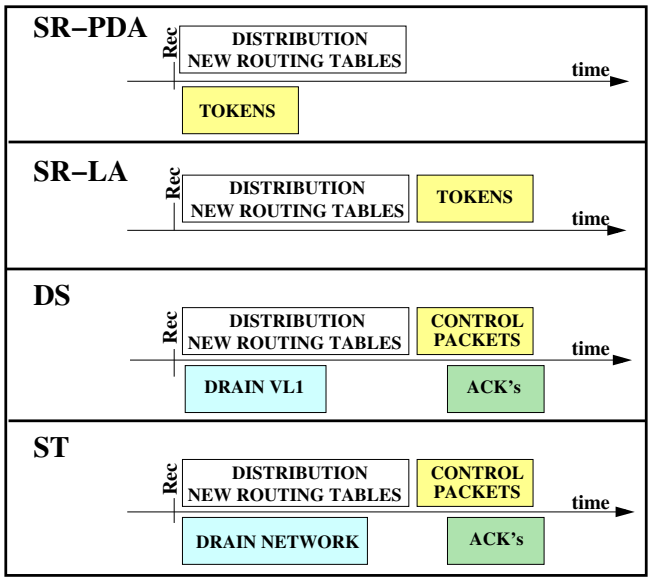
3.2 Implementation of Dynamic Network Reconfiguration

We have implemented two types of SR; the former updates the routing tables before starting the reconfiguration process, while the latter updates the routing tables simultaneously to the reconfiguration process, in this paper we refer to each one as SRLA and SRPDA, respectively [11].

For every reconfiguration mechanism implementation, we consider that there is a network manager (NM) located on an arbitrary end node that monitors the network for changes and orchestrates each reconfiguration process. NM communicates with every network device by sending control packets through a reserved control virtual channel. Here, we provide a description of each reconfiguration mechanism implementation:

- In SR-PDA (SR Packet Drop Aware), new routing tables are sent in parallel with a "reconfiguration" control broadcast packet from the NM to all end nodes and switches, signaling nodes to generate reconfiguration tokens.
- In SR-LA (SR Latency Aware), the NM firstly distributes and stores the new routing tables into a secondary location in the end nodes and switches before it broadcasts the reconfiguration control packet. The latter potentially would reduce average packet latency at the expense of possibly longer reconfiguration time.
- In the Double Scheme (DS), once a topology change is detected, the NM computes the new routing tables, and afterwards, it distributes them to all end nodes and switches. At the same time, it sends a "virtual channel drain" control packet to all switches and end nodes instructing them to drain one of the two data virtual channels (i.e., VC-1). Packets residing in that data virtual channel are transported through the other data virtual channel at each switch. Drainage thus occurs in parallel across all the network switches. When the virtual channel being drained (VC-1) is empty, the NM is notified. The NM then signals the end nodes and switches—via control packets—to start using both data virtual channels with the new routing function. The drained virtual channel is used as the escape path for any old packets in the other data virtual channel. Reconfiguration completes once all end nodes and switches are able to use both data virtual channels again.
- In ST, once the topology change is detected, the NM broadcasts a "network drain" control packet to all end nodes, instructing them to halt packet injection. At the same time, the NM starts computing the new routing tables. Once the tables are computed and distributed to all nodes, the network completely drain all data packets. Then, the NM sends control packets instructing the end nodes to resume packet injection.

Figure 2 summarizes the events that distinguish each of the reconfiguration schemes used in this evaluation.



Rec: Start of the reconfiguration process

Figure 2. Reconfiguration schemes

4 Evaluation

4.1 Simulation Environment

To evaluate the reconfiguration mechanism, we have developed a detailed simulator that allows us to model the network at the register transfer level. The simulator models a typical InfiniBand architecture (IBA) network[8].

Packets are routed at each switch by accessing the forwarding table. This table contains the output port to be used at the switch for each possible destination. The routing time at each switch will be set to 100 ns. This time includes the

time to access the forwarding tables, the crossbar arbiter time, and the time to set up the crossbar connections.

Switches can support up to 16 virtual lanes (VLs). VLs can be used to form separate virtual networks. We will use a non-multiplexed crossbar on each switch. This crossbar supplies separate ports for each VL. Buffers will be used both at the input and the output side of the crossbar. Buffer size will be fixed in both cases to 10 KB.

Links in InfiniBand are serial. In the simulator, the link injection rate will be fixed to the 1X configuration [8]. 1X cables have a link speed of 2.5 Gbps. Therefore, a bit can be injected every 0.4 ns. With 8/10 coding [8] a new byte can be injected into the link every 4 ns. We also model the fly time (time required by a bit to reach the opposite link side). We will model 20 m copper cables with a propagation delay of 5 ns/m. Therefore, the fly time will be set to 100 ns.

The IBA specification defines a credit-based flow control scheme for each virtual lane with independent buffer resources. A packet will be transmitted over the link if there is enough buffer space (credits of 64 bytes) to store the entire packet. IBA allows the definition of different MTU (Maximum Transfer Unit) values for packets ranging from 256 to 4096 bytes. Additionally, the virtual cut-through switching technique is used.

The duration of the simulations are about 25 ms, and the results are collected after a transient state of 40,000 packets is completed.

For each simulation run, the workload (packet generation rate) is the same for all the end-nodes. The traffic rate keeps constant during the transient state. Then, the traffic rate linearly decreases during 1 ms to the 10%. It keeps the low value of the traffic rate during the next 1 ms, and it increases linearly again to the initial value.

Packet size is set to 58 bytes which includes the IBA packet header (20 bytes), the packet payload (32 bytes) and the IBA packet tail (6 bytes).

4.2 Simulation Results

4.2.1 Behaviour of Network Reconfiguration

We compare static reconfiguration (ST), Simple Reconfiguration (SRPDA, SRLA)[12], and Double Scheme (DS)[16] for various baseline traffic loads on 8×8 two-dimensional torus networks. Figures 3, 4, 5, and 6 show their behaviour under low, medium and high traffic rates. We show the breakdown of traffic; "CONTROL_VC" is used by the control packets for the routing update, and reconfiguration, while "Data VC_0 and 1" are used by data traffic.

In each baseline traffic rate, we have applied the different reconfiguration mechanisms twice for on/off link operations. At the former second half of the simulation, links are de-activated according to the reduction of the traffic rate. At the latter second half, links are re-activated according to the increase of the traffic rate.

Figure 3 shows that the ST introduces heavy packet latency during the on/off link operations even under low traffic, while Figures 4 and 5 show that the SRLA and SRPDA mitigate the packet latency. In addition, as shown in Figure 6, the DoubleScheme has almost no overhead during the on/off link operation. We can also said that the packet latency when the link is reactivated is smaller than that when the link is deactivated in all the reconfiguration schemes.

4.2.2 Reconfiguration Overhead and Maximum Latency

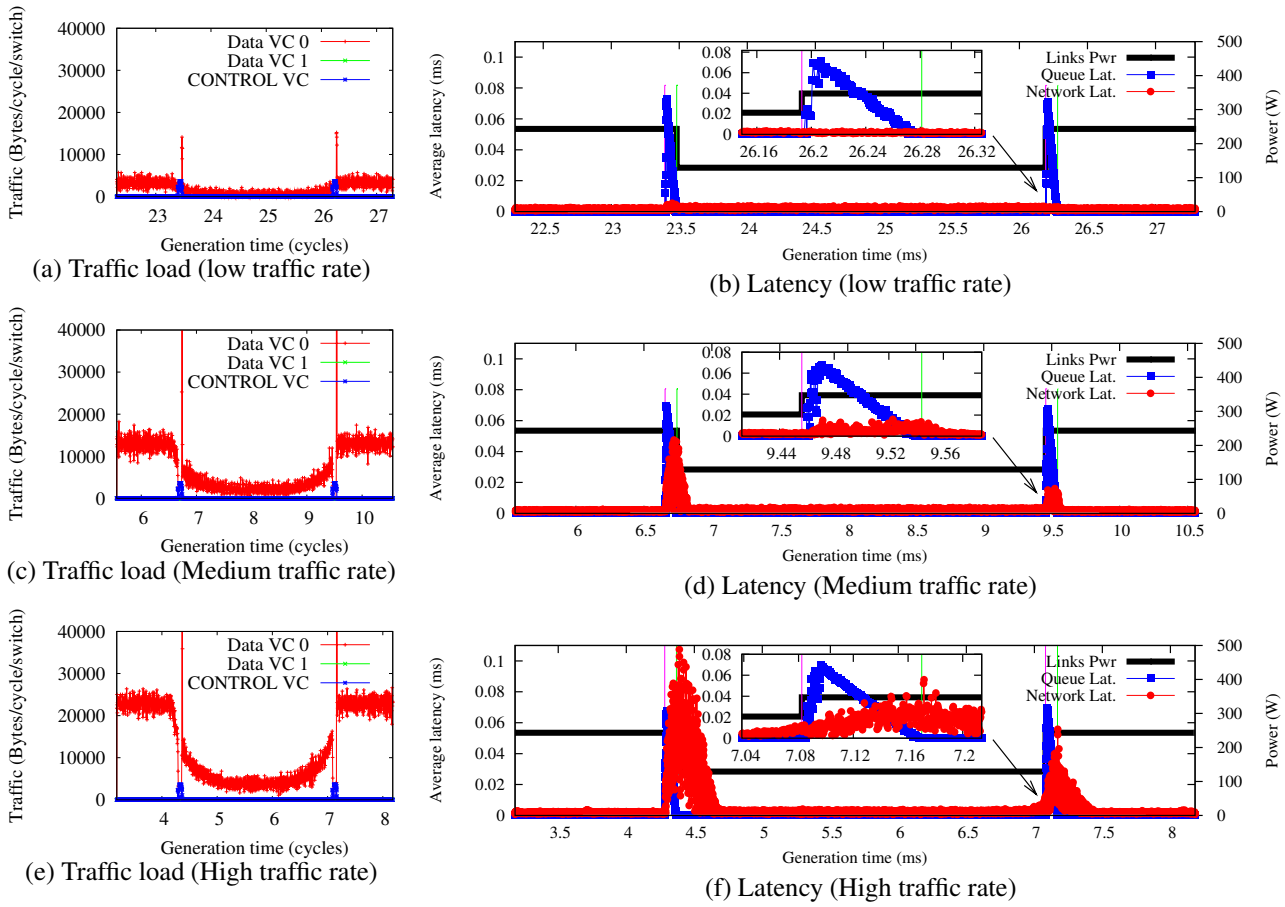
Figure 7.a shows the reconfiguration overhead of each network reconfiguration mechanism (bars filled with solid color), and the transient period where the packet latency increases after the reconfiguration finishes (bars filled with line pattern) on 8×8 two-dimensional torus network. Figure 7.b shows the maximum value of the latency caused by each reconfiguration process. Figures 7.a and 7.b show that DoubleScheme reduces by up to 95% the peak packet latency compared with STatic reconfiguration in on/off interconnection networks. Although SPRDA, SRLA are dynamic reconfiguration, their overhead cannot be ignored.

5 Conclusions

Power saving is required for interconnects of modern PC clusters as well as the performance improvement. To reduce the power consumption of switches in PC clusters with maintaining the performance, on/off link regulations that activate and deactivate the links based on the traffic load has been widely developed[1][9][19]. Depending on which operation is selected, link activation or deactivation, the available network resources are changed, thus requiring paths to be reconfigured.

To maintain deadlock freedom of packet transfers, connectivity, and performance during the path changes, we propose to apply dynamic reconfiguration techniques where packet transfer is processed uninterruptedly. We investigate the impact of network reconfiguration technique on performance and behavior as soon as links activation or deactivation occurs.

From the evaluation results we can conclude that the dynamic reconfiguration should be applied for on/off link regulations in order to stabilize the path update, since it allows migration between different network configurations without interrupt the network traffic. In particular, the DoubleScheme reduces by up to 95% the peak packet latency caused by the change compared with existing static reconfiguration in power-aware on/off interconnection networks.



(a) Traffic load (low traffic rate)

(b) Latency (low traffic rate)

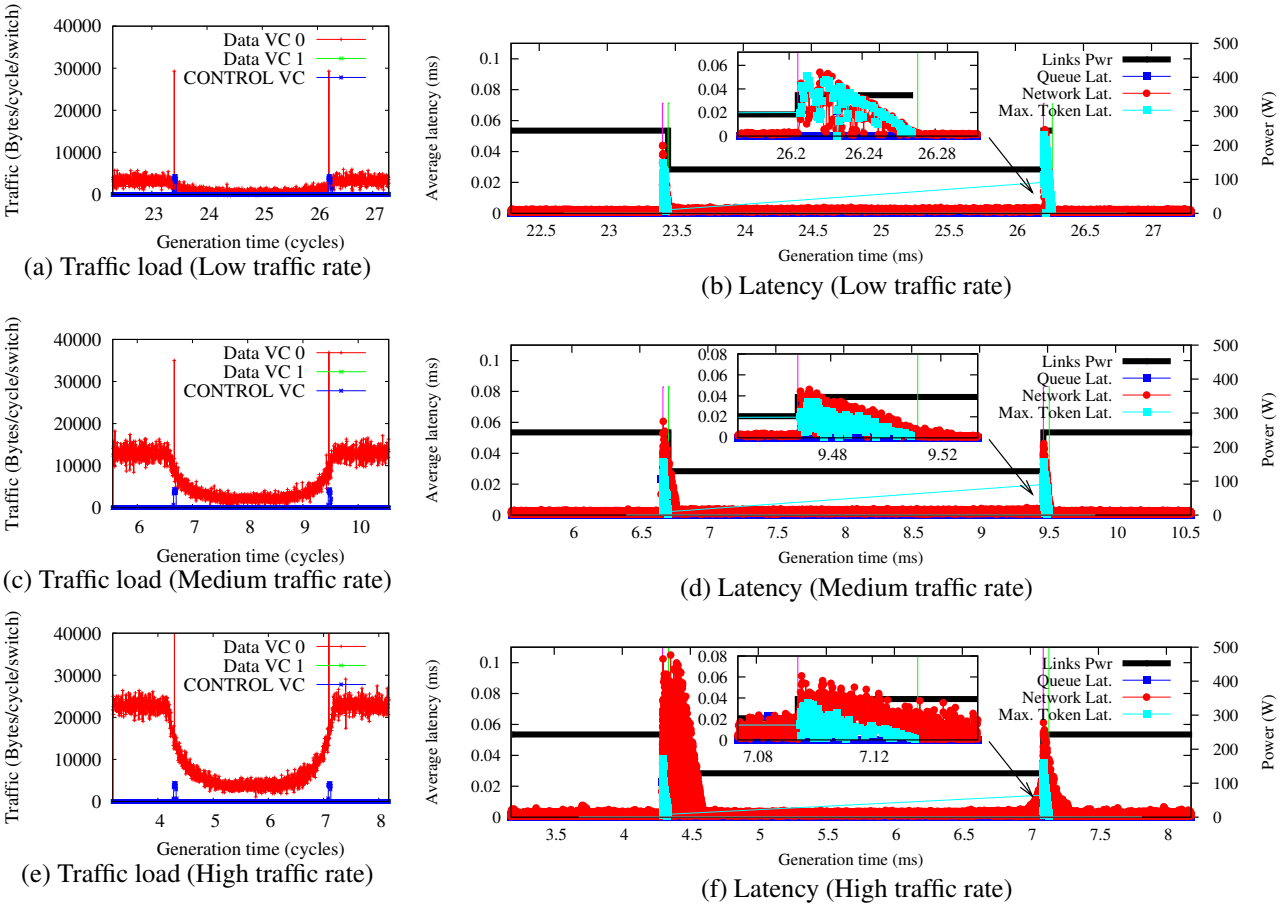
(c) Traffic load (Medium traffic rate)

(d) Latency (Medium traffic rate)

(e) Traffic load (High traffic rate)

(f) Latency (High traffic rate)

Figure 3. Behavior of network reconfigurations by de-activating and re-activating link using the ST



(a) Traffic load (Low traffic rate)

(b) Latency (Low traffic rate)

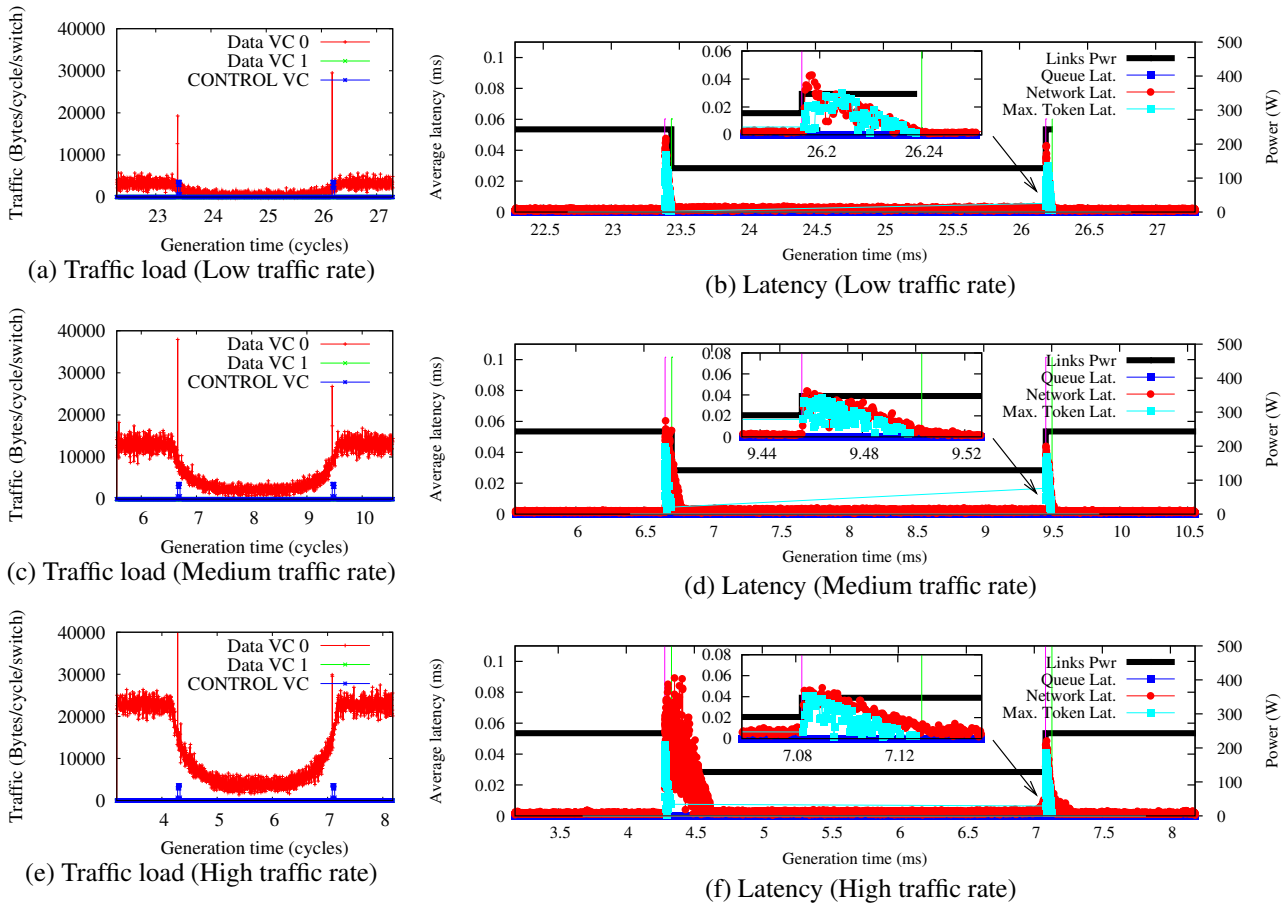
(c) Traffic load (Medium traffic rate)

(d) Latency (Medium traffic rate)

(e) Traffic load (High traffic rate)

(f) Latency (High traffic rate)

Figure 4. Behavior of network reconfigurations by de-activating and re-activating link using the SRLA



(a) Traffic load (Low traffic rate)

(b) Latency (Low traffic rate)

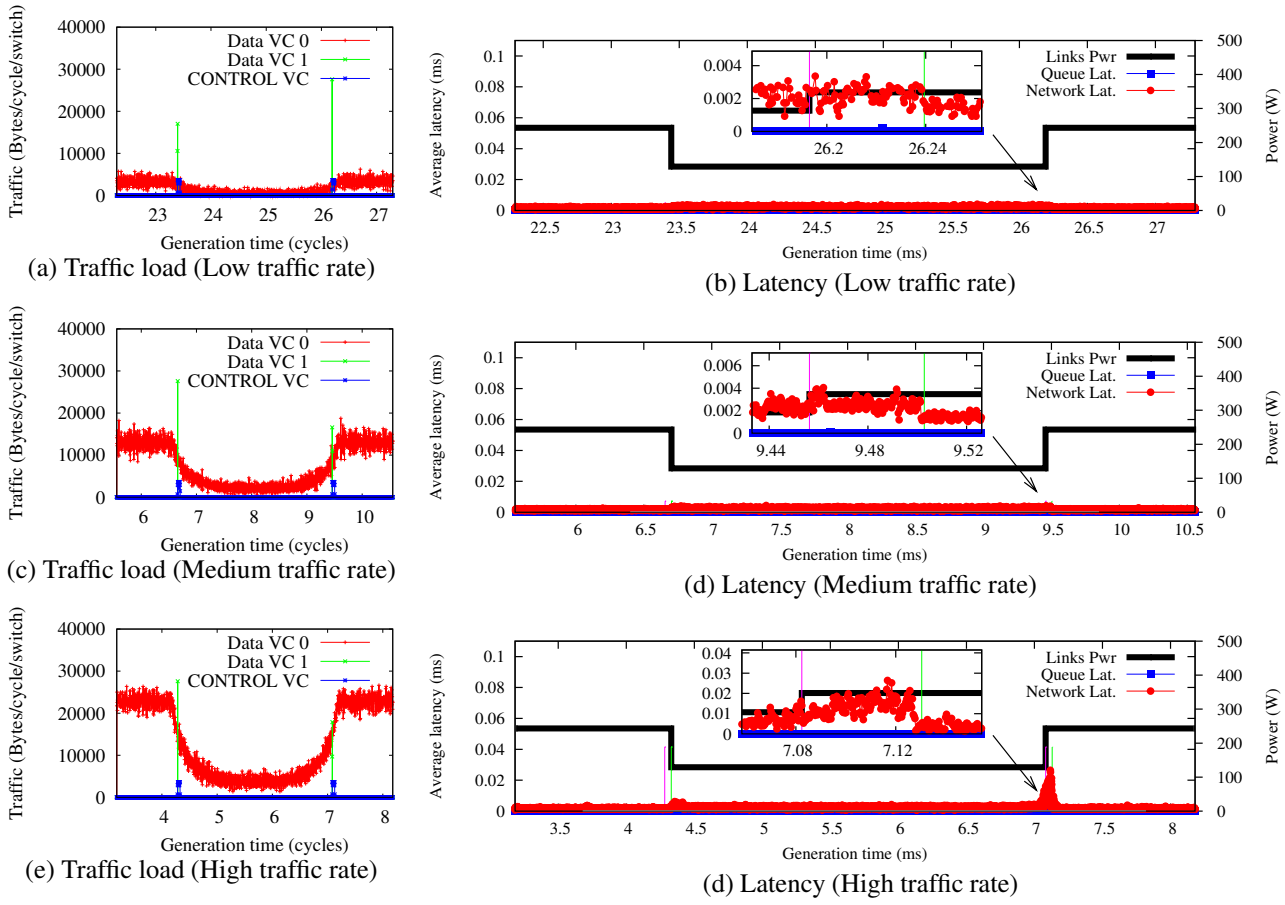
(c) Traffic load (Medium traffic rate)

(d) Latency (Medium traffic rate)

(e) Traffic load (High traffic rate)

(f) Latency (High traffic rate)

Figure 5. Behavior of network reconfigurations by de-activating and re-activating link using the SR-PDA



(a) Traffic load (Low traffic rate)

(b) Latency (Low traffic rate)

(c) Traffic load (Medium traffic rate)

(d) Latency (Medium traffic rate)

(e) Traffic load (High traffic rate)

(d) Latency (High traffic rate)

Figure 6. Behavior of network reconfigurations by de-activating and re-activating link using DS

Acknowledgments

This work was partially supported by JST CREST (ULP-HPC: Ultra Low-Power, High-Performance Computing via Modeling and Optimization of Next Generation HPC Technologies).

References

[1] M. Alonso, J. M. Martinez, V. Santonja, P. Lopez, and J. Duato. Power Saving in Regular Interconnection Networks Built with High-Degree Switches. In *International Parallel and Distributed Processing Symposium*, 2005.

[2] A. Bermúdez, R. Casado, F. Quiles, and T. Pinkston. Evaluation of a subnet management mechanism for infiniband networks. In *Proc. 32nd Intl Conf. Parallel Processing*, pages 117–124, 2003.

[3] R. Casado, A. Bermúdez, J. Duato, F. J. Quiles, and J. L. Sáchez. A protocol for deadlock-free dynamic reconfiguration in high speed local area networks . *IEEE Transactions on Parallel and Distributed Systems*, 12:115–132, 2001.

[4] R. Casado, A. Bermúdez, F. J. Quiles, J. L. Sánchez, and J. Duato. Performance evaluation of dynamic reconfiguration in high-speed local area networks. In *Proceedings of the Sixth International Symposium on High-Performance Computer Architecture*, 2000.

[5] J. Duato, O. Lysne, R. Pang, and T. M. Pinkston. Part I: A theory for deadlock-free dynamic network reconfiguration. *IEEE Transactions on Parallel Distributed Systems*, 16(5):412–427, 2005.

[6] C. J. Glass and L. M. Ni. The Turn Model for Adaptive Routing. *Proceedings of International Symposium on Computer Architecture*, pages 278–287, 1992.

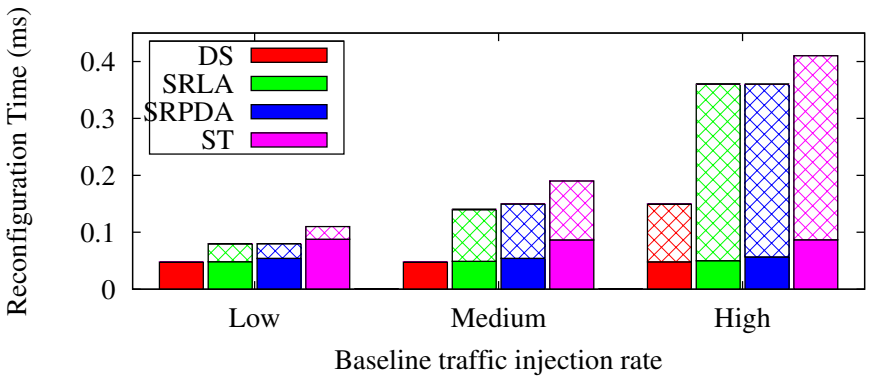
[7] InfiniBand Trade Association. `http://www.infinibandta.org/`.

[8] InfiniBand Trade Association™. *InfiniBand Architecture specification release 1.2*, October 2004.

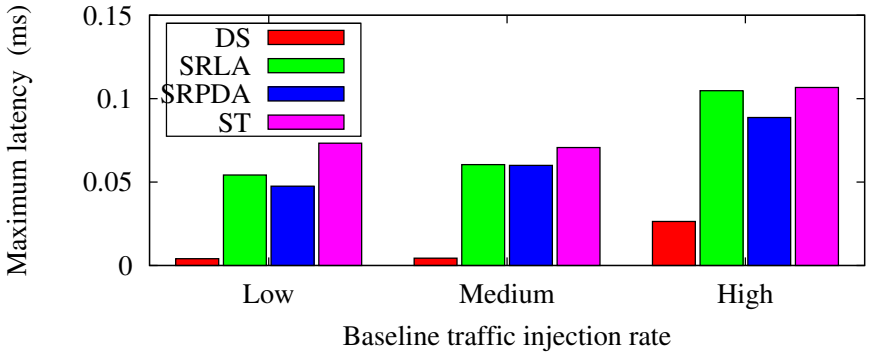
[9] M. Koibuchi, T. Otsuka, H. Matsutani, and H. Amano. An On/Off Link Activation Method for Low-Power Ethernet in PC Clusters. In *International Parallel and Distributed Processing Symposium*, 2009.

[10] O. Lysne and J. Duato. Fast dynamic reconfiguration in irregular networks. In IEEE Computer Society, editor, *Proceedings of the 2000 International Conference of Parallel Processing*, pages 449–458, Toronto (Canada), 2000.

[11] O. Lysne, J. M. Montañana, J. Flich, J. Duato, T. M. Pinkston, and T. Skeie. An Efficient and Deadlock-free Net-



(a)



(b)

Figure 7. (a) Reconfiguration overhead and (b) maximum packet latency of each network reconfiguration mechanism

work Reconfiguration Protocol. *IEEE Transactions on Computers*, 57(6):762–779, June 2008.

[12] O. Lysne, J. M. Montañana, T. M. Pinkston, J. Duato, T. Skeie, and J. Flich. Simple Deadlock-Free Dynamic Network Reconfiguration. In *Proceedings of the 11th International Conference on High Performance Computing (HiPC)*, Bangalore (India), 19-22 December 2004.

[13] Myricom. `http://www.myri.com/`.

[14] N. Natchev, D. Avresky, and V. Shurbanov. Dynamic reconfiguration in high-speed computer clusters. In *Proceedings of the International Conference on Cluster Computing*, pages 380–387, Los Alamitos (USA), 2001. IEEE Computer Society.

[15] T. Otsuka, M. Koibuchi, T. Kudoh, and H. Amano. A Switch-tagged VLAN Routing Methodology for PC Clusters with Ethernet. In *Proc. of the 2006 International Conference on Parallel Processing (ICPP-06)*, pages 479–486, Aug. 2006.

[16] T. M. Pinkston, R. Pang, and J. Duato. Deadlock-free dynamic reconfiguration schemes for increased network dependability. *IEEE Transactions on Parallel and Distributed Systems*, 14(8):780–794, August 2003.

[17] V. Puente, J. A. Gregorio, F. Vallejo, and R. Beivide. Immunet: A Cheap and Robust Fault-Tolerant Packet Routing Mechanism. In *Proceedings of the* 31th *Annual International Symposium on Computer Architecture*, 2004.

[18] L. Shang, L.-S. Peh, and N. K. Jha. Dynamic Voltage Scaling with Links for Power Optimization of Interconnection Networks. In *Proceedings of the International Symposium on High-Performance Computer Architecture*, pages 79–90, Jan. 2003.

[19] V. Soteriou and L.-S. Peh. Exploring the Design Space of Self-Regulating Power-Aware On/Off Interconnection Networks. *IEEE Transactions on Parallel and Distributed Systems*, 18(3):393–408, Mar. 2007.

[20] J. M. Stine and N. P. Carter. Comparing Adaptive Routing and Dynamic Voltage Scaling for Link Power Reduction. *IEEE Computer Architecture Letters*, 3(1):14–17, Jan. 2004.