On Deadlock Frequency during Dynamic Reconfiguration in NOWs *

Lorenzo Fernández[1], José M. García[1] and Rafael Casado[2]

[1] Universidad de Murcia. Murcia, SPAIN 30071. {lfmaimo,jmgarcia}@ditec.um.es
[2] Universidad de Castilla-La Mancha. Albacete, SPAIN 02071.
rcasado@info-ab.uclm.es

Abstract. NOWs executing multimedia and real-time applications need to handle dynamic changes in their irregular topologies. This may be carried out in two ways: statically and dynamically. In the former, user traffic is stopped, causing latencies to increase dramatically. In the latter, user traffic is not stopped but deadlocks may appear in the transition phase between the old and the new routing function. To solve this problem, dynamic reconfiguration methods based on deadlock avoidance have been proposed. However, another possibility not studied yet, is to use dynamic reconfiguration allowing deadlock formation with an efficient detection/recovery mechanism. It is necessary to know deadlock frequency during the reconfiguration process in order to assess the viability of this alternative proposal. In this paper, we show that deadlocks may become very infrequent with few virtual channels and the lost message problem can be reduced by using a simple misrouting technique.

1 Introduction

High performance networks of workstations (NOWs) have become a low-cost alternative to parallel computers, not only for high-performance scientific computing but mostly as a platform for high-end servers. A high performance NOW is built by interconnecting workstations by means of point-to-point links and switches. NOWs may suffer topological updates due to link failures, switches/hosts being turned on/off, link remapping, hot expansion, etc. Usually, the network itself detects the change and starts a reconfiguration process in order to obtain the new topology. The main problem in reconfigurable networks is to guarantee deadlock-free routing during the reconfiguration process. Another related problem is the number of lost messages during this process.

A simple way to avoid deadlocks while the reconfiguration process is being executed is *static reconfiguration*. In this approach, deadlock-free routing is guaranteed by stopping user traffic until reconfiguration finishes. In this case, there is no risk of deadlocks, but user messages have to wait for the reconfiguration to end, dramatically increasing their latency and making it inappropiate for multimedia and real-time applications with strict traffic requirements.

* This work has been supported in part by the spanish Ministry of Ciencia y Tecnología under project TIC2000-1151-C07-03

On the other hand, in *dynamic reconfiguration*, user messages can flow during the reconfiguration in a normal way. During the reconfiguration process deadlocks may appear, even when a deadlock-free routing function is being used. This is due to the coexistence in the network of at least two versions of the routing tables (the old ones and the new ones). To solve this important problem, there are two strategies: dynamic reconfiguration based on either a deadlock detection/recovery mechanism, or on a deadlock avoidance mechanism.

Deadlock-avoidance mechanisms are more dificult to implement and possibly, they may need more network resources than deadlock detection/recovery ones. However, the former are more efficient. To develop an alternative efficient deadlock detection/recovery mechanism during the dynamic reconfiguration process, it is necessary to guarantee that deadlocks are very infrequent. In this paper, our main objetive is to estimate deadlock frequency and the amount of lost messages. We have found that deadlocks are not very frequent for irregular networks with a low number of virtual channels (for 64 switches, the ratio deadlocked/delivered messages are below 2%). Moreover, the ratio lost/delivered messages is also maintained at a very low level (for 64 switches, below 0.2%).

This paper is organized as follows. In Sect. 2, we provide some background on high-performance networks, including static and dynamic reconfiguration. Section 3 shows the importance of choosing a good mechanism for detecting deadlocks and how often both deadlocks and lost messages occur during the reconfiguration process, even when misrouting is allowed. Section 4 depicts the evaluation results and finally, we end the paper by showing some conclusions as well as future work.

2 Background

Current high-speed LANs (Autonet [14], Myrinet [3], ServerNet [6] and InfiniBand [9]) may change their topology due to switches and host being turned on/off, link remapping, and component failures. A very important issue in these high-speed LANs is the deadlock problem. It is known that deadlocks occur more frequently when irregular networks are used [15]. There are two main strategies for deadlock handling: deadlock avoidance and deadlock recovery. Deadlock avoidance prevents deadlocks by restricting routing so that there may not be cyclic dependencies between channels [5]. Autonet implements up*/down* routing tables for this purpose. Myrinet implements a deterministic version of up*/down* based on source routing. Deadlock avoidance can be also achieved by providing some virtual channels and escape paths [7].

Deadlock recovery requires both deadlock detection and recovery mechanisms. The classic method for deadlock detection has been the timeout-based mechanism, that measures the inactivity time of blocked messages, leading to a high probability of false deadlock detection and a high dependency on message length. ICT (Inactive Channels Time) [12] and a very efficient improvement of ICT [10] can be found in literature as two alternative methods. ICT measures the time that channels requested by messages are inactive due to the fact that

the current messages occupying them remain blocked. A message is presumed to be deadlocked only if all of the alternative virtual output channels requested by that message contain blocked messages and have been inactive for a given period of time.

Recovery may be carried out in either a progressive or a regressive way. This depends on whether some resources are deallocated from a non-deadlocked message and reassigned to a deadlocked one for quick delivery [1, 8], or some resources are deallocated from deadlocked messages, usually discarding them.

2.1 Reconfiguration Mechanisms

Reconfiguration mechanisms in current high-speed LANs are based on static reconfiguration techniques. When a change in the topology occurs, a three-phases reconfiguration process (propagation, collection and distribution) is triggered. The node which detects the change cleans its buffers, stops accepting user messages and becomes the root node of the new topology. Then, it sends control messages to every neighbour announcing that a reconfiguration has started and including information about the new spanning tree that is beginning to be built. A node that receives such a control message must also join the spanning tree at its best position, clean its buffers, stop accepting user messages and propagate the reconfiguration process to its neighbours. The spanning tree built in this way is called *Propagation-Order Spanning Tree* (POST) and it may not be a minimal spanning tree. The collection phase begins when every node has joined the spanning tree. Then, each node sends the topological information about its sub-tree to its parent. When this information reaches the root node, it is known that the second phase has ended. At this point, the root knows the whole network topology. The last phase distributes the new topology through the spanning tree until all switches are aware of it. Next, user messages are allowed again.

A first proposal of dynamic reconfiguration based on deadlock avoidance, called PPR (Partial Progressive Reconfiguration) was presented in [4]. In this approach, routing tables are gradually asynchronously updated, in a controlled way, so that they remain deadlock-free after each partial update. User messages are not stopped, and this technique is valid for both regular and irregular topologies. Recently, other similar approaches [2, 11, 13] has been proposed in this context.

3 Deadlock Formation in Dynamic Reconfiguration

Our research work is focused on estimating the deadlock frequency during dynamic reconfiguration. The underlying dynamic reconfiguration method is a dynamic version of the static one implemented in Autonet. Deadlocks produced by routing table interaction may occur during dynamic reconfiguration. Messages that cross switches using only old routing tables, only new ones or a mixture of both can be found. Thus, although both old routing tables and new ones

are deadlock-free, a "mix-routed" message may break the restrictions, forming a cycle in the channel dependency graph.

Warnakulasuriya [15] showed that deadlock formation likelihood decreases as the number of virtual channels increases in a network. If enough virtual channels are provided, it is expected the number of deadlocks to remain low also in the course of the reconfiguration. Therefore, an appropiate number of virtual channels has to be selected in order to obtain a low number for deadlocks.

In order to implement deadlock recovery, it is necessary to detect deadlock formation. We are interested in distributed deadlock detection mechanisms that use only local information and few hardware resources. These mechanisms find every true deadlock, but they also detect false ones. ICT has been the mechanism selected because it detects much fewer false deadlocks than timeout and it is very easy to implement.

A regressive deadlock recovery technique has been used. When a deadlock is detected, the message responsible for the triggering is discarded and its buffer resources are released so that the rest of the messages can advance towards their destinations. The discarded message is supposed to be injected again into the network by an upper-level protocol.

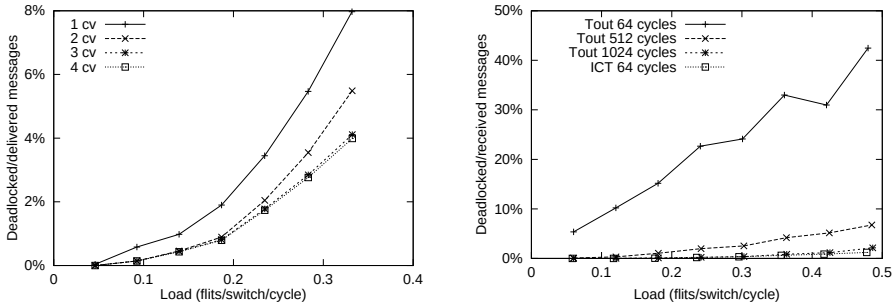
3.1 Lost Messages

Deadlocks are not the only problem faced during network reconfiguration. The loss of messages is another important question. When a switch is deactivated, crossing messages get lost.

For the purpose of this work, user traffic is not stopped; thus, in dynamic reconfiguration there are three categories of lost messages: (a) those messages addressed to a non-existent switch in the network, which will be dropped when meeting their destination; (b) those messages traveling towards an existent switch but following a deadlock-free path that crosses a deactivated switch; and (c) those messages that are routed using contradictory information that breaks the up*/down* rules. These messages cannot continue on their way satisfying the restrictions on routing tables for remaining deadlock-free, so they are discarded. Lost messages can be managed by a higher-level protocol, and then, resent, discarded or even reinjected by means of in-transit buffers [8], but, in any case, it is desirable to reduce their number.

3.2 Reducing the number of deadlocked and lost messages

A first approach for reducing the deadlock frequency, is to re-route a deadlocked message using any free channel with enough free buffer space, in order to avoid discarding it. In small networks, such a misrouted message will probably soon be involved in another deadlock. However, as network size grows it is expected that the likelihood of a new deadlock being formed will decrease. In addition, the greater adaptivity provided by virtual channels is expected to help avoid new deadlocks. If there are no free channels, the message is discarded as mentioned above. This misrouting is only used during the reconfiguration process because



(a) (b)

Fig. 1. (a) Deadlocked/delivered messages vs. accepted traffic when varying the number of adaptive virtual channels used. (b) Comparing ICT vs. timeout. Network with 32 switches and packet length of 512 bytes

there is no possibility of a deadlock occurring after it finishes. Livelock cannot occur because reconfiguration always ends.

On the other hand, this technique could be also useful for reducing the number of lost messages. When a message is about to be lost, it can be applied misrouting to avoid losing it. Probably, this could also increase the number of deadlocked messages because a misrouted message augments its probability of involvement in a deadlock configuration. Therefore, it is interesting to obtain some experimental results about the use of misrouting in this contex.

4 Evaluation

Our network simulator has the characteristics described in Sect. 3, i.e. dynamic version of the reconfiguration method of Autonet, ICT with message discarding as deadlock recovery policy and misrouting. Our switch architecture implements virtual cut-through and virtual channels, with a partially multiplexed crossbar. There is just one escape channel and the rest are adaptive ones. Several irregular topologies have been randomly generated containing 16, 24, 32 and 64 switches, with one host per switch. Packet length is 512 bytes, destination distribution is uniform and deadlock detection mechanism is ICT.

One by one, each switch is deactivated while the rest remain activated. This triggers as many one-deactivation reconfigurations as switches the network has. For 64-switch network, it was deactivated only a random subset of switches instead of the whole set. Simulations were made progressively increasing the applied load until the network saturation was achieved. The measures were taken from the time the reconfiguration begins, until the time the last switch finishes its reconfiguration process. Inserting a new node behaves in the same way than a deactivation, but with much few lost packets.

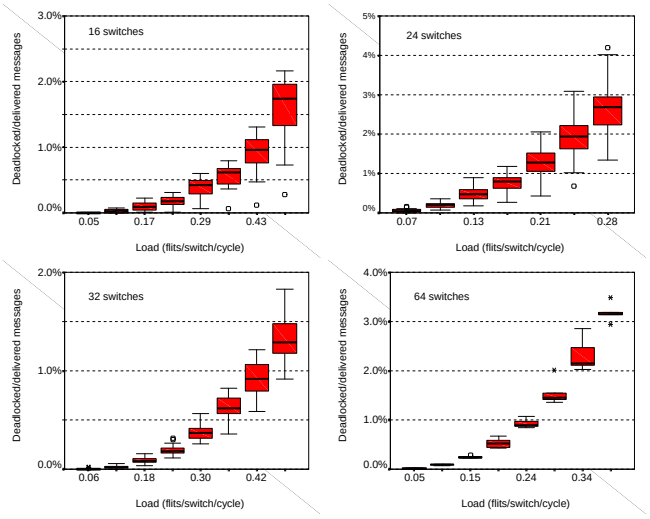


Fig. 2. Deadlock frequency versus delivered traffic

Firstly, an evaluation of the number of virtual channels needed to maintain the deadlock frequency low will be carried out. Figure 1(a) shows that three adaptive virtual channels are enough to maintain low the number of deadlocked messages, thus, we choose to use three adaptive channels plus an escape one.

Concerning deadlock detection, timeout and ICT were tested. This is depicted in Fig. 1(b). It can be noted that ICT needs a lower threshold than timeout. Moreover, due to this, ICT detects deadlocks before timeout does. Thus, ICT was chosen as the deadlock detection mechanism for the rest of our evaluations.

In Fig. 2 deadlocked/delivered messages ratio versus accepted traffic is depicted for networks of 16, 24, 32 and 64 switches. In order to obtain a comprehensive view of the deadlock behaviour, a box-and-whiskers [1] graph of the mean latencies was used.

When the network has a light traffic load, deadlock frequency remains low in all tested networks (e.g. below 1%-2%). The closer to saturation the network is, the more frequent deadlocks become. In addition, we have found that the ratio of deadlocked/delivered messages does not depend on deactivating a particular node. However, there are some particular cases, such as disconnecting a leaf node, which results in an almost up*/down* graph inversion. In those cases, deadlocks are more frequent because there is a large amount of conflicting routing information between old and new routing tables. Conversely, when a node near the root is deactivated, the new routing table has much common information with the old one, producing much fewer deadlocks.

[1] This statistical graph draws a box containing 50% of the sample, marking the median with a line. The whiskers that stick out of the box represent the rest of the sample. Rare cases are shown by means of small circles and asterisks.

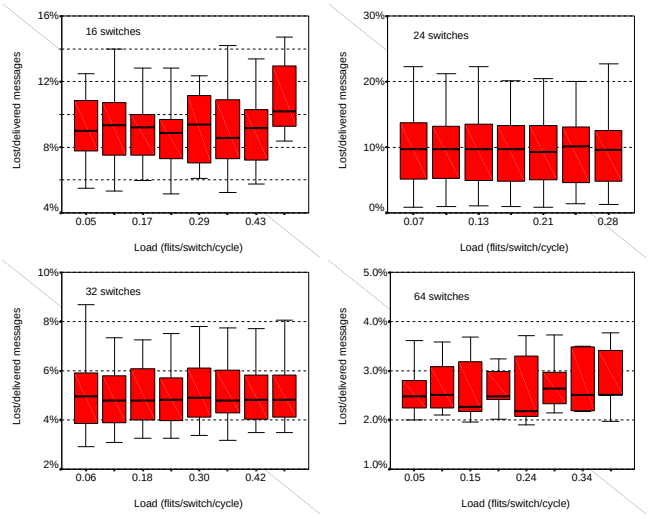


Fig. 3. Lost message frequency versus delivered traffic

With respect to lost messages, the ratio lost/received messages strongly depends on the particular disconnected switch whereas it does not depend on the load. A set of Box-and-whiskers graphs summarize lost message frequency when every switch, each in turn, is deactivated and the traffic is increased until network saturation. The parameters of the network are the same as Fig. 2 and it is shown in Fig. 3.

For leaf nodes, there are few up*/down* paths that use it. Thus, few messages will get lost. However, when the switch that disappears is close to the root node, a great number of messages will probably try to cross it. As a result, many messages will get lost. Figure 4 shows the results of simulating the misrouting mechanism with the same networks as early experiments. With low load, deadlock frequency is higher than it was without misrouting, but this frequency is aproximately the same when the network is close to saturation. A misrouted message stays into the network longer than a well-routed one, thus augmenting both load and deadlock frequency. As far as lost messages are concerned, Fig. 4 shows that the ratio lost/delivered messages considerably decreases for every network size. The larger the size, the greater the enhancement; for example, a 32-switch network reduces this ratio from a maximum of approximately 9% to 2%. Indeed, in 64-switch network the frequency drops from a maximum of 3.75% to 0.3% approximately.

5 Conclusions and Future Work

In this paper, we have shown that using a low number of virtual channels (only four), deadlocks during the dynamic reconfiguration become infrequent in high speed local area networks (NOWs). In order to obtain both lower and upper

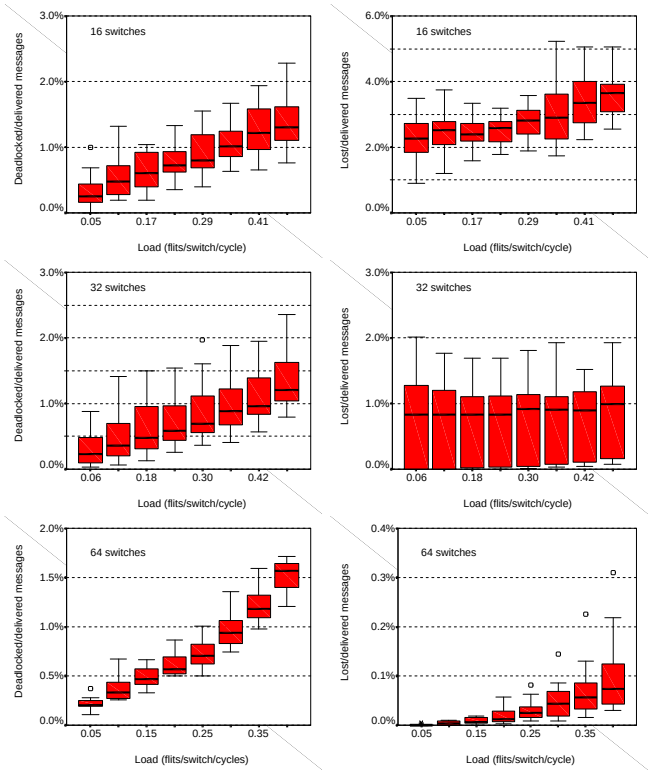


Fig. 4. Deadlock frequency and lost/delivered message ratio when misrouting is allowed

boundaries we have carried out several evaluations for each switch being deactivated one by one, while the rest remain activated. This fact, in addition to the relatively short duration of the reconfiguration process, could lead us to consider a deadlock detection/recovery approach as a suitable proposal. But the percentage of lost messages become an important problem. It only depends on the particular switch that is disconnected and does not depend on the traffic load. However, it has been shown that a simple misrouting technique provides good results in both the ratio of deadlocked and lost messages.

Future work include to carry out a deeper analysis of deadlock frequency, and to make a comparison between dynamic reconfiguration methods with deadlock avoidance and the one developed in this paper, based on deadlock detection/recovery.

6 Acknowledgements

The authors would like to thank Jose Duato, Juan Peinador and Francisco Alfaro for providing insightful comments that greatly improved this paper.

References

1. Anjan K. V. and T. M. Pinkston, "An efficient fully adaptive deadlock recovery scheme: DISHA," *Proceedings of the 9th International Parallel Processing Symposium*, April 1995.
2. D. R. Avresky, N. Natchev and V. Shubarnov, "Dynamic reconfiguration in high-speed computer networks." *IEEE Workshop on Embedded Fault Tolerant Systems*, Washington D.C., September 21-22, 2000.
3. N. J. Boden, D. Cohen, R. E. Felderman, A. E. Kulawik, C. L. Seitz, J. Seizovic and W. Su, "Myrinet - A gigabit per second local area network," *IEEE Micro*, pp. 29-36, February 1995.
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." *Sixth International Symposium on High Performance Computer Architecture (HPCA-6)*, Toulouse, France. January 10-12, 2000.
5. W. J. Dally and C. L. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," *IEEE Trans. on Computers*, vol. C-36, no. 5, pp.547-553, May 1987.
6. D. García and W. Watson. "ServerNet II," *Proceedings of the 1997 Parallel Computer, Routing, and Communication Workshop*, pp. 119-135, June 1997.
7. J. Duato, "A new theory of deadlock-free adaptive routing in wormhole networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 4, no. 4, 466-475, April 1993.
8. J. Flich, M. P. Malumbres, P. López and J. Duato, "Improving routing performance in Myrinet networks," in *International Parallel and Distributed Processing Symposium (IPDPS 2000)*, May 2000.
9. "InfiniBand architecture specification. Volume 1. Release 1.0," Infiniband Trade Association, October 2000. http://www.infinibandta.org
10. P. López, J. M. Martínez and J. Duato, "A very efficient distributed deadlock detection mechanism for wormhole networks," *Proceedings of the Fourth International Symposium on High-Performance Computer Architecture (HPCA-4)*, IEEE Computer Society Press (1998) 57-66.
11. O. Lysne and J. Duato, "Fast dynamic reconfiguration in irregular networks," *International Conference on Parallel Processing (ICPP 2000)*, August 2000.
12. J. M. Martínez, P. López, J. Duato and T. M. Pinkston, "Software-based deadlock recovery technique for true fully adaptive routing in wormhole networks," *Proceedings of the 1997 International Conference on Parallel Processing (ICPP'97)*, IEEE Computer Society Press (1997) 182-189.
13. R. Pang, T. Pinkston and J.Duato, "The double scheme: Deadlock-free dynamic reconfiguration of cut-through networks," *International Conference on Parallel Processing (ICPP 2000)*, August 2000.
14. M. D. Schroeder et al, "Autonet: a high-speed, self-configuring local area network using point-to-point links," *IEEE Journal on Selected Areas in Communications*, 9(8):1318-1335, October 1991.
15. S. Warnakulasuriya and T. M. Pinkston, "Characterization of deadlocks in interconnection networks," *Proc. of the 11th International Parallel Processing Symposium (IPPS'97)*, April 1997.