Distributing InfiniBand Forwarding Tables*

Aurelio Bermúdez, Rafael Casado, and Francisco J. Quiles

Department of Computer Science, University of Castilla-La Mancha
02071 Albacete, Spain
{aurelio.bermudez,rafael.casado,francisco.quiles}@uclm.es

Abstract. InfiniBand is an emerging technology both for communication between processing nodes and I/O devices, and for interprocessor communication. After the occurrence of a topology change, InfiniBand management entities collect the current topology, compute new forwarding tables, and upload them to routing devices. Traditional distribution techniques prevent deadlock but, at the same time, they affect negatively user traffic. In this paper, we propose two alternative deadlock-free mechanisms to distribute forwarding tables. These proposals adhere InfiniBand specification, can be easily implemented, and reduce significantly the impact of the distribution process.

1 Introduction

The InfiniBand Architecture (IBA) [7] defines a technology for interconnecting processor nodes (hosts) and I/O nodes to form a system area network. Hosts and I/O nodes are interconnected using an arbitrary (possibly irregular) switched point-to-point network, instead of using a shared bus. End nodes use channel adapters (CAs) to connect to the fabric. The network is composed of one or more subnets interconnected by routers. Each port within a subnet has a 16-bit local identifier (LID). Switches perform intra-subnet routing using the packet's destination LID included in the header of the packet. A forwarding table (FT) specifies which port forwards the packet.

IBA subnets are managed in an autonomous way. There is a subnet management mechanism capable of assimilating any topology change without external intervention, guaranteeing service availability. The specification defines various subnet management entities, describing their functions and the structure of the control packets used to exchange information among them. An entity called the subnet manager (SM) is in charge of discovering, configuring, activating, and maintaining the subnet. This entity exchanges subnet management packets (SMPs) with subnet management agents (SMAs) present in every device. Fig. 1(a) shows an example of irregular subnet including these management entities. In [2], we presented a completely functional prototype of a subnet management protocol which adheres to IBA specifications. This initial approach covers the detection of topology changes, device discovery, and

* This work was partly supported by the following projects: CICYT TIC2003-08154-C6-02, and JCCM PBC-02-008.

M. Danelutto, D. Laforenza, M. Vanneschi (Eds.): Euro-Par 2004, LNCS 3149, pp. 864–872, 2004.

computation and distribution of subnet routes. The discovery process was optimized in [3], and the computation of subnet routes was improved in [4].

In this work, we focus on the last step in assimilating a topology change, i.e., the distribution of switch forwarding tables. The SMPs for updating these tables are completely defined in the IBA specification. However, the update order is not detailed. Updating FTs in an uncontrolled way could generate deadlock situations [10]. The reason is that although the new and the previous sets of subnet routes are deadlock-free, the coexistence of both routing schemes during the distribution process is not necessarily deadlock-free.

In order to prevent deadlock situations during this process, static reconfiguration [5, 12] was assumed in our initial implementation. This means that user traffic is stopped while forwarding tables are being updated. As we showed in [2], this technique has negative effects over user traffic; in particular, a temporary lack of service and, consequently, a massive packet discarding during the period of time in which subnet ports are inactive.

The impact of static reconfiguration could be reduced by using dynamic reconfiguration techniques, such as *Partial Progressive Reconfiguration* [6], *Skyline* [8], and *Double Scheme* [11]. All these techniques allow the distribution of forwarding tables without stopping network traffic, guaranteeing deadlock-freedom during the process. However, the adaptation to IBA of these dynamic reconfiguration mechanisms is not trivial, due to it implies the modification of the current specification in some way (addition of new elements, the use of provided elements for different purposes, the assumption that SMAs perform special functions, etc.).

In this paper we present and analyze several alternatives that do not involve any change in the IBA specification. In particular, the idea is to relax the traditional static reconfiguration technique, either by reducing the amount of subnet ports that must be actually deactivated, or by preventing the use of certain transitions during the distribution of switch forwarding tables.

The remainder of this paper is organized as follows. First of all, Section 2 describes the way in which static reconfiguration is performed in IBA, and informally presents two optimized distribution processes. In Section 3 we comparatively analyze the different distribution techniques through several simulation results. Finally, Section 4 gives some conclusions and future work.

2 Improving the Distribution Process

2.1 Static Distribution

The static distribution of forwarding tables is performed in three sequential steps. First, all subnet ports are deactivated by the SM. In particular, the SM sends a *SubnSet(PortInfo)* SMP [7] to change the state of each port to *INITIALIZE*. In this state, the port can only receive and transmit SMPs, discarding all other packets received or presented to it for transmission. The next step is the sending of the FTs itself. This phase is performed using either *SubnSet(LinearForwardingTable)* SMPs or *SubnSet(RandomForwardingTable)* SMPs. Finally, after the SM verifies that all

subnet switches have received their tables, user traffic must be allowed again, by activating subnet ports. By means of *SubnSet(PortInfo)* SMPs, the SM sets the state of each port to *ACTIVE*. In this state, the port can transmit and receive all packet types.

SMPs used to perform the two first steps must employ directed (source) routing [7]. The reason is that new FTs have not been configured yet. Instead, SMPs for the activation phase can use either directed or LID (destination) routing.

2.2 Deactivation of Break Node Ports

This distribution technique assumes that the management mechanism uses the *up*/down** routing algorithm [12] to compute the set of subnet FTs. Up*/down* is a popular deadlock-free algorithm valid for any topology. It is based on a cycle-free assignment of direction to the operational links in the network. For each link, a direction is named *up* and the opposite one is named *down*. In this way, the network is configured as an acyclic directed graph with a single sink node. As an example, Fig. 1(b) shows a possible assignment of directions for the topology in Fig. 1(a). To avoid deadlocks, legal routes never use a link in the *up* direction after having used one in the *down* direction.

In a directed graph, a *break node* [6] is a node that is the source of two or more arcs. Break nodes prevent certain transitions (input port – output port) from being used by the packets crossing them. These restrictions are necessary to guarantee deadlock-freedom. In Fig. 1(b), transitions $6_{1\rightarrow 2}$ and $6_{2\rightarrow 1}$ (from port *1* to *2* and vice versa in node *6*) are forbidden. Similarly, transitions $10_{2\rightarrow 3}$ and $10_{3\rightarrow 2}$ are not allowed. In other words, the corresponding dependencies between channels are deactivated [6].

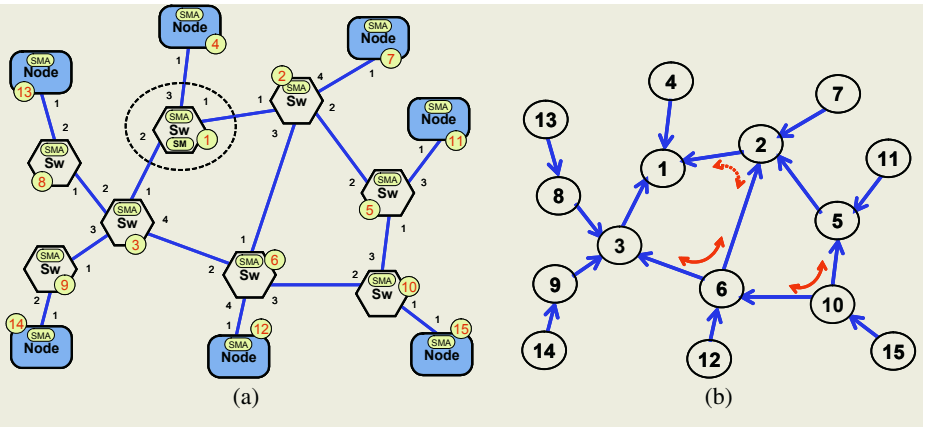


Fig. 1. (a) Example of irregular subnet topology composed of 8 switches and 7 end nodes. Circled numbers represent the LID assigned to each subnet device by the SM (located in node *1*) during the discovery process. Small numbers at the ends of the links represent switch and channel adapter port numbers. (b) A possible directed graph for this topology.

Deadlocks can appear during the distribution process only if the break node in a cycle changes its position. Let us suppose that the previous break node in the left cycle of Fig. 1(b) was the node labeled as *2*. That means that the direction assigned to the link connecting nodes *6* and *2* in the previous configuration was the opposite one of the currently assigned. Therefore, transitions $2_{1\rightarrow 3}$ and $2_{3\rightarrow 1}$ were not allowed. If the distribution process activates the dependencies for node *2* before the deactivation of the dependencies in node *6*, there is a potential deadlock in the cycle. The reason is that node *2* could route packets from *1* to *6* and, simultaneously, node *6* could route packets from *2* to *3*, closing the cycle. Deadlock could also appear in the opposite direction.

Obviously, a static distribution process prevents these situations, because of subnet ports are not activated until new FTs have been completely distributed. An optimized distribution mechanism could be based on the deactivation of only break node dependencies, instead of deactivating subnet ports, before the sending of tables. That would imply not to allow input port – output port transitions in break nodes. Unfortunately, we cannot program switch FTs to prevent these transitions. The reason is that IBA switches do not take into account the input port used by the packet to route it. This information is not stored in the tables.

As an intermediate step, we could derive a deadlock-free distribution mechanism that only deactivates the ports of the switches that will act as break nodes in the new configuration. Moreover, note that it is not necessary to deactivate all ports in those nodes. Instead, it is enough to select those ports associated with *up* links, and deactivate all of them, except one. Thus, we ensure that forbidden transitions will not be used by any packet. For the directed graph in Fig. 1(b), the distribution process only must deactivate port 6_1 or 6_2. Similarly, it is only necessary to deactivate port 10_2 or 10_3. In this way, we are allowing the use of many subnet routes during the distribution of tables. As in the static distribution process, there is a third step, after the distribution of tables, in which break node ports are activated.

2.3 Deactivation of Break Node Dependencies

The previous mechanism is easy to implement and, as we will see, it reduces the negative effects of the "pure" static distribution process. However, there are still a lot of subnet routes that could be used during the sending of tables, without introducing potential deadlock situations. For example, let us suppose that, in Fig. 1(a), port 10_2 has been selected for deactivation. In this situation, we are not allowing those packets generated or destined to node *10* that must use the link connecting *10* and *6*.

We can improve the distribution mechanism by only preventing a few input port – output port combinations, allowing the rest of routing options. To do that, we can conveniently program the set of subnet SL to VL mapping tables. The IBA specification itself suggests that these tables can be used to avoid routing deadlocks [7].

IBA uses a virtual lane (VL) based mechanism to create multiple virtual links within a single physical link. Each port could implement 1, 2, 4, 8, or 15 data VLs. As a packet traverses the subnet, each link along the path can support a different

number of VLs. Therefore, it is not possible to transmit a packet only using a VL specified in its header. Instead, IBA uses a more abstract criterion, based on the concept of service level (SL). Each switch has a SL to VL mapping table to establish, for each pair input port – output port, a correspondence between the service level of the packet (a number from 0 to 15) and the VLs supported by the output port assigned to the packet.

Packets are discarded if the SL to VL mapping table returns the value VL15. This is the value that we are going to use to prevent forbidden transitions. The resulting distribution mechanism does not require deactivating any subnet port. Instead, it replaces the initial deactivation step with the sending of SL to VL mapping tables to new break nodes. In particular, it is necessary to send a SL to VL mapping table for each forbidden input port – output port combination in each break node. In this table, all entries contain the value VL15. Therefore, packets trying these transitions will be automatically discarded.

For the example in Fig. 1(b), it is necessary to configure four SL to VL mapping tables, in order to deactivate forbidden dependencies in *6* and *10*. The sending of SL to VL mapping tables is performed by the SM by using directed routed *SubnSet(SLtoVLMappingTable)* SMPs.

3 Performance Evaluation

All the results presented in this work have been obtained through simulation. Before showing and analyzing them, we briefly describe the simulation methodology.

3.1 Simulation Methodology

Our model embodies key physical and link layer features of IBA, allowing the simulation of various IBA-compliant network designs. Also, it incorporates the subnet management entities and packets defined in the specification. To develop it, we have used the OPNET Modeler [9] simulation software. The current IBA model is composed of copper links (supporting different data rates), 4-port fully demultiplexed switches, and end nodes containing a channel adapter (hosts). Several physical and link layer details are described in [1, 2, 3, 4].

We have evaluated randomly generated irregular subnets with 8, 16, 24, and 32 switches, assuming that there is at least a host connected to each switch, if a port is available. Also, not all switch ports are connected. Each subnet switch supports a linear forwarding table (LFT) with 1,024 entries.

We have considered a packet maximum transfer unit (MTU) of 256 bytes (the minimum MTU value allowed by the IBA specification). The packet generation rate is Poisson, and it is expressed in packets/sec/node. Traffic sources also use a uniform distribution to obtain the packet destination (among all the active hosts) and the SL value (from 0 to 15). The traffic load applied is different for each subnet topology. We have selected low load values (25% of saturation rate), in order to prevent network saturation during the analysis.

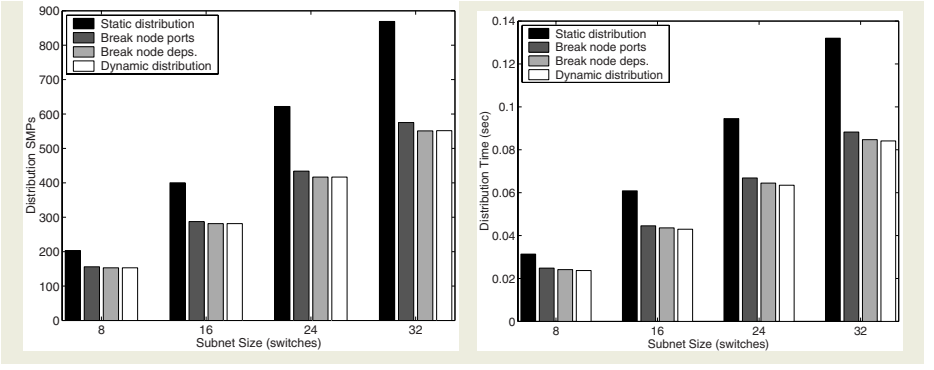


Fig. 2. Control packets and time required to distribute switch forwarding tables as a function of subnet size. The change consists of the addition of an individual switch. Results for switch removal (not shown here) are very similar.

For each simulation run, after a transient period we have programmed a topology change, consisting of the addition or removal of a switch. The experiment is repeated for each switch in the subnet, and average values are shown in the plots. Traps support is disabled, and the period of time between consecutive sweepings has been tuned according to the subnet topology. The simulation is stopped once the topology change has been completely assimilated.

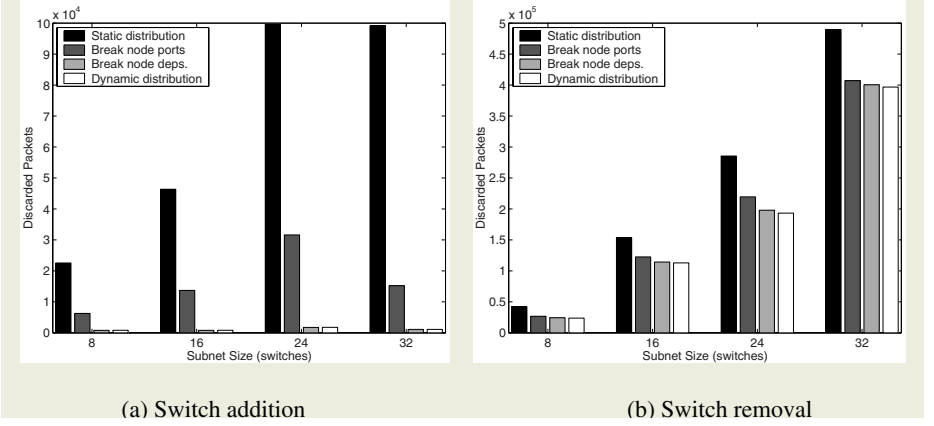
3.2 Simulation Results

In this section we analyze the behavior of the distribution techniques presented and their impact over application traffic. For the sake of comparison, we have added to some plots a series showing the results for a basic dynamic distribution process. This process consists of only one step, in which FTs are directly distributed to subnet switches, without deactivating ports or dependencies. Therefore, it is deadlock-prone.

Fig. 2 shows the amount of SMPs and the time required by the subnet management mechanism to update switch FTs after the occurrence of a topology change, as a function of the distribution mechanism used and the subnet size. Results also consider both the deactivation and activation phases (when applicable).

We can see that the technique based on the deactivation of break node ports achieves an important reduction of both parameters. Also, we can observe an additional reduction for the distribution mechanism based on the deactivation of break node dependencies. In fact, results for dynamic distribution are almost identical.

Fig. 3 shows the amount of packets discarded as a function of the distribution mechanism used, the type of topology change, and the subnet size. Note that, in case of switch removal, there is a big amount of packets that are discarded due to many subnet routes disappear, at least until new tables have been distributed and alternative routes are provided. Independently of the distribution technique applied, this massive discarding is inevitable, because of IBA routes are deterministic.



(a) Switch addition (b) Switch removal

Fig. 3. Number of packets discarded during the change assimilation as a function of subnet size. Note that vertical scales are different.

In Fig. 3(a) we can appreciate a considerable improvement if we only deactivate break node ports, due mainly to the number of packets dropped by deactivated ports decreases significantly. The number of discarded packets decreases even more for the deactivation of break node dependencies. Packets discarded by deactivated ports have been replaced by packets dropped by the SL to VL mapping process. As happens before, results provided by this technique are very similar to dynamic distribution.

Note that there are other minor causes for packet discarding. For example, during the transient period, there are packets that can not be routed because of the corresponding DLID is not found in a FT, or because of the output port returned by the table coincides temporary with the packet's input port.

Finally, Fig. 4 shows the effects of the distribution mechanisms over user traffic. Note that the figure only shows a detail of the distribution phase, instead of the complete change assimilation process. For all plots, the X-axis represents the simulation time. The first plot shows the aggregate amount of SMPs exchanged by the management entities. The big step in this plot corresponds exactly with the distribution of subnet ta-

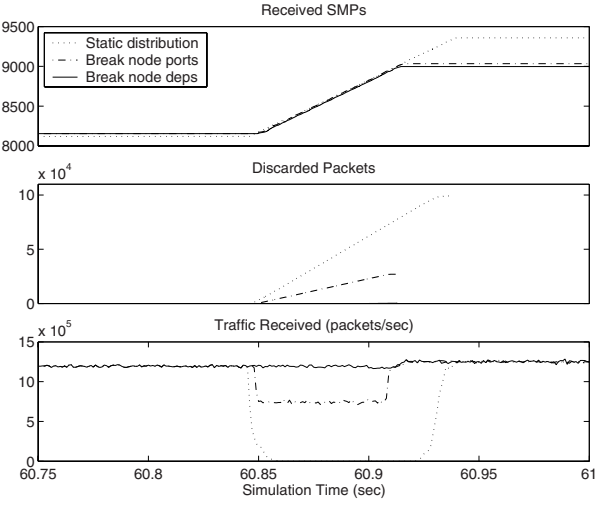


Fig. 4. Detail of the distribution process for an irregular subnet composed of 24 switches and 22 hosts and a change consisting of the addition of a switch (at time 60.1 sec.).

bles. Second plot represents the aggregate amount of discarded packets during the simulation. As the change considered is a switch addition, the period of time during packets are discarded coincides with the distribution process. The last plot shows instantaneous network throughput.

In the top plots we can observe (again) a reduction in the amount of distribution SMPs and time, and in the number of packets discarded, when the proposed distribution mechanisms are used. Bottom plot shows the way in which the optimized techniques improve network throughput during the process. Static reconfiguration has a very negative effect over instantaneous throughput. We can appreciate that the proposed distribution techniques improve considerably this behavior. In fact, when the mechanism based on the deactivation of break node dependencies is used, the gap in this plot completely disappears.

4 Conclusions

We have presented two deadlock-free mechanisms to distribute InfiniBand subnet forwarding tables. These mechanisms have been directly derived from the traditional static reconfiguration process. However, they have a much better behavior. We have seen that the impact of the distribution process over application traffic is practically avoided. The reason is that the proposals allow the use of many subnet routes during the period of time in which forwarding tables are being distributed. The main advantage of these proposals is that they respect the InfiniBand specification and, therefore, they can be easily implemented over real systems.

References

1. Bermúdez, A., Casado, R., Quiles, F.J., Pinkston, T.M., Duato, J.: Modeling InfiniBand with OPNET. Workshop on Novel Uses of System Area Networks, February 2003
2. Bermúdez, A., Casado, R., Quiles, F.J., Pinkston, T.M., Duato, J.: Evaluation of a subnet management mechanism for InfiniBand networks. In Proc. IEEE Int. Conference on Parallel Processing, October 2003
3. Bermúdez, A., Casado, R., Quiles, F.J., Pinkston, T.M., Duato, J.: On the InfiniBand subnet discovery process". In Proc. IEEE Int. Conference on Cluster Computing, December 2003
4. Bermúdez, A., Casado, R., Quiles, F.J., Duato, J.: Use of provisional routes to speed-up change assimilation in InfiniBand networks. In Proc. Workshop on Communication Architecture for Clusters, April 2004
5. Boden, N.J., et al.: Myrinet: A gigabit per second LAN. IEEE Micro, February 1995
6 Casado, R, Bermúdez, A., Quiles, F.J., Sánchez, J.L., Duato, J.: A protocol for deadlock-free dynamic reconfiguration in high-speed local area networks. IEEE Transactions on Parallel and Distributed Systems, vol. 12, no. 2, February 2001
7. InfiniBand Architecture Specification (1.1), November 2002. http://www.infinibandta.com/
8. Lysne, O, Duato, J.: Fast dynamic reconfiguration in Irregular networks. In Proc. Int. Conference on Parallel Processing, August 2000

9. OPNET Technologies, Inc. http://www.opnet.com/
10. Pinkston, T.M., Pang, R., Duato, J.: Deadlock-free dynamic reconfiguration schemes for increased network dependability. IEEE Transactions on Parallel and Distributed Systems, vol. 14, no. 6, June 2003
11. Pinkston, T.M., Zafar, B., Duato, J.: A method for applying Double Scheme dynamic reconfiguration over InfiniBand. In Proc. Int. Conference on Parallel and Distributed Processing Techniques and Applications, June 2003
12. Schroeder, M.D., et al.: Autonet: a high-speed, self-configuring local area network using point-to-point links. IEEE Journal on Selected Areas in Communications, vol. 9, no. 8, October 1991