Improving Network Performance by Efficiently Dealing with Short Control Messages in Fibre Channel SANs*

Xavier Molero, Federico Silla, Vicente Santonja, and José Duato

Departament d'Informàtica de Sistemes i Computadors,
Universitat Politècnica de València, Camí de Vera, 14. 46022 València, Spain
jmolero@disca.upv.es

Abstract. Traffic in a Storage Area Networks is bimodal, composed of long messages carrying several KBytes of data, and short messages containing control information (I/O commands). From the network point of view, latency of control messages is highly affected by the transmission of data messages, due to their length. As a consequence, it is necessary to establish management policies that benefit the transmission of short control messages, thus reducing the overall response time for I/O operations and increasing network throughput.
In this paper we propose several strategies for dealing with short control messages and analyze their impact on the performance of storage area networks. This analysis is carried out for a fully adaptive routing algorithm in the context of two different network topology environments: buildings and departments. Simulation results show that both I/O response time and network throughput may be improved when efficiently managing control messages.

1 Introduction and Motivation

I/O technology has been traditionally based on the server-storage architecture, where each storage device is connected to a single server, usually by means of a SCSI bus. However, this approach is now facing several problems such as unavailability of data due to server down times, bandwidth saturation during backups, and also a great number of limitations when implementing large configurations.

An emerging alternative to the traditional I/O architecture is based on the concept of SAN (*Storage Area Network*) [4,11]. A SAN is a high-speed network, similar to a LAN, that makes possible to connect servers to storage devices (see Figure 1). Thus, in the same way that LANs allow clients easy access to many servers, SANs provide access to storage devices from many servers. SANs allow storage to be shared among many servers without impacting LAN performance, which is improved because now it is free from the high overhead associated with

* This work was supported by the Spanish CICYT under grants TIC2000-1151-C07 and TIC2000-0472-C03-03

R. Sakellariou et al. (Eds.): Euro-Par 2001, LNCS 2150, pp. 901–911, 2001.

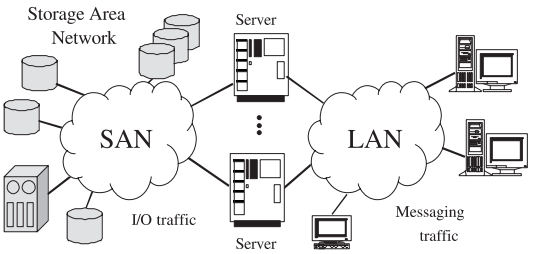


Fig. 1. A typical SAN environment

file retrieval, storage, and data backing up tasks. Moreover, data access is enhanced because file read/write and data migration are more effectively handled by a network that can be optimized for the demands of storage operations (e.g., high throughput and large packet data transfers). Most of current SANs are implemented using Fibre Channel technology [2,4] due to its suitability for storage networking.

Servers initiate the communication with disks by issuing a read or a write I/O operation. In both cases each I/O operation implies the transmission of two messages: a request and a response. In the case for read operations, the server first sends to the selected disk a request message containing the number of sectors to read, and the initial location of the data. Once the device has processed the request, it sends back the data to the server. Regarding write operations, servers initiate them by sending to the selected disk the data and the location where they must be stored. Once the data have been stored, the device returns an acknowledgment message to the server.

Thus, messages exchanged between servers and storage devices can be classified into two different types: short control messages (read requests or write acknowledgments) and long data messages. Usually, control messages are not longer than a few bytes, while typical data messages are a few KBytes long. Therefore, traffic in a storage area network becomes bimodal, being composed of long and short messages in the same percentage. From the network point of view, latency of control messages is highly affected by the transmission of data messages, which tend to monopolize physical links due to their length, making control messages to experience long queueing times, thus increasing the overall I/O response time. On the other hand, when a control message arrives at a switch it reserves an entire input buffer, which is usually large enough to store a whole data message or, at least, part of it. Due to the short length of control messages, reserving a whole buffer means wasting most of the buffering capacity, leading to a decrement in network throughput, and thus increasing the overall I/O response time.

In [6], Kim and Chien discusses different techniques to deal with bimodal traffic in the context of wormhole-routed networks. In this paper we propose several

ways of managing control messages and analyze their effect on the performance of Fibre Channel storage area networks.

The remainder of this paper is organized as follows. Section 2 presents our model of SANs. The different strategies proposed for dealing with control messages are described in Section 3. Section 4 describes the evaluation methodology. The effect of each strategy is evaluated in Section 5. Finally, Section 6 summarizes the conclusions from our study.

2 System Model

The basic switch architecture we have considered is shown in Figure 2. Each input port has an associated pool of buffers to store incoming messages. When a message arrives at a port, it starts being read into one of those buffers. The routing unit iteratively polls input ports, in a round-robin scheduling policy, for new messages that need to be routed. This unit can start routing a message as soon as the header information has arrived. If it finds new messages in a port, the router selects the first one and reads the header information, which contains the destination address. If the requested output link is busy, then the incoming message remains in the input buffer until it is successfully routed. An internal crossbar with as many input ports as input buffers allows multiple messages to be simultaneously forwarded without interference.

Fibre Channel uses the virtual cut-through switching mechanism [5]. In this case, when a message finds the output port busy, it is blocked in place, being completely stored at the input buffer of the current switch. Thus, message length is bounded by the input buffer size. This limitation can be easily overcome by splitting messages into packets. The flow control mechanism used in Fibre Channel to manage the transfer of data between two adjacent switches is based on credits.

Network topologies used in a SAN environment can be classified into two different categories: those that map an entire building and those intended to be used in a single room (laboratory, data center, computing site, etc.). Topologies in the first group can easily map to a multi-floor installation [9]. In these topologies most connections are between switches and devices in each floor. The

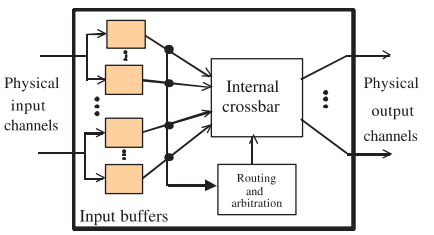


Fig. 2. Switch architecture

structure at each floor consists of a star configuration with center switches and arm switches. This structure is replicated as many times as floors are included in the SAN. Center switches are specifically devoted for connecting switches at different floors. Servers and storage devices are attached to the SAN by means of arm switches. We have used the tree topology as an example of toplogies used in departmental environments. Servers and disks are connected to the leaf switches. The rest of switches are devoted to communication between devices attached at different leaves.

Fibre Channel uses distributed routing, leaving routing decisions to each of the switches in the path from source to destination. In our study, we have considered the minimal adaptive routing algorithm [10].

3 Strategies for Managing Control Messages

In this section we present several strategies for managing short control messages. The main purpose of these policies is to improve response time of I/O operations by reducing network latency of control messages, at the same time network throughput is increased due to a better management of input buffers.

Routing with Higher Priority. Incoming messages are managed by the routing and arbitration unit, which polls input ports for messages to be routed in a round robin fashion. While control messages are waiting to be routed, output channels may be assigned to data messages located ahead in the polling path, thus diminishing the choices that control messages will have when being routed later, and increasing their queueing time. Therefore, if control messages were routed with a higher priority, they would be able to leave the switch sooner, decreasing I/O response time.

When routing depends on message priority, the routing and arbitration unit also polls input ports in a round robin scheme, but the scheduling policy is modified so that control messages arriving at a switch have higher priority. They will be routed before than the rest of messages at the switch. In each round, a maximum of one new control message per port can be routed. If a high priority control message has not been successfully routed, then it will be assigned a regular priority in following rounds. This will prevent starvation. In order to implement this scheme, the routing unit must be able to distinguish newly arrived control messages from those that have been unsuccessfully routed in previous rounds. This can be easily done by storing the message state in an one-bit register per input buffer.

Transmission with Higher Priority. The delay of control messages is mainly due to the fact that they must wait for a free output channel before being forwarded. If the message being transmitted through that channel is a data message, then the control message should wait for long. Control message queueing delay may be avoided if control messages were assigned the required output channel

once they have been routed, preempting the transmission of the data message owing the output link. Once the control message has been transmitted, then the switch can continue the transmission of the stopped data message. The transmission of a control message cannot be preempted by another control message.

In order to implement this scheme, the receiving switch must be able to correctly identify and extract the control messages that have been inserted into the data stream of a data message, so that they are stored in the proper buffers. This is easily overcome by using the header and the tail information associated with each message. When a control message preempts the transmission of a data message, the header of the control message helps the receiving switch to store the incoming message in a different buffer. At the same time, it would record the preemption, so that when the tail of the control message arrives, the receiving switch may continue storing the rest of the data message in the proper buffer.

Separating Input Buffers. Buffers at input ports may be divided into two disjoint sets: one for storing data messages and another one for control messages. As a consequence, the network would behave as divided into two independent subnetworks, one for data messages and another one for control messages. However, both kinds of messages still share the same physical links, routing units, and internal crossbars. Thus, long data messages can already interfere with control messages. Nevertheless, this solution avoids wasting input buffer capacity because buffers intended for control messages may be considerably reduced in size.

Message Packetization. When using message packetization, messages are decomposed into several packets of fixed length, which travel through the network independently from each other. Each packet carries its own routing information at its header.

From the network point of view, transmitting packets instead of longer messages avoids the monopolization of physical channels, allowing short control messages to advance to their destination faster. However, some disadvantages arise when using message packetization. One of them is the overhead introduced in the amount of information transmitted. It may also increase the amount of credit messages transmitted because of the flow control protocol. Also, the network must independently route each of the packets at the switches they traverse, thus highly increasing the number of cycles wasted in the routing process. The implementation of this strategy only needs modifying the software at the sender and the receiver ends: the sender must split the message into smaller packets, and the receiver must reassemble the received packets in order to get the original message.

4 Evaluation Methodology

In order to accurately model the SAN, we have used byte level simulation. Our simulator [7] has been implemented using the CSIM language [3]. Simulations

were run for a number of cycles high enough to obtain steady values of I/O response times. An initial transient period corresponding to the completion of 500 I/O operations has been considered. Then, simulations were run for a number of cycles high enough to obtain steady values of I/O operation times.

We have used synthetic traffic, considering that interarrival time for the generation of I/O operations is exponentially distributed and it is the same for all the servers attached to the storage network. Moreover, we have assumed that the destination disk of each I/O operation initiated by a server is randomly chosen among all the disks in the network. We have also assumed that the number of I/O read operations is similar to the number of I/O write operations.

Disk access time has been shown to be the dominant factor in the total I/O response time [8], thus becoming the bottleneck of the entire storage system. In order to stress the network, we have assumed that disks access data fast enough to avoid becoming a bottleneck.

Due to the lack of information about current Fibre Channel implementations, switch parameters have been taken similar to those in high-speed Myrinet networks [1]. In Myrinet networks, the time needed to route and forward the first byte of a message is 150 ns. Following bytes take 6.25 ns to traverse the switch. We have considered that each switch has eight 2048-byte buffers per input port devoted to store data messages.

Current Fibre Channel switches use full duplex serial links at 100 MBytes/s. However, higher bandwidths are under development nowadays. Thus in this study we have considered that links transfer data at 160 MBytes/s. Link length may range from a few meters until several kilometers when optical fiber is used. We have considered both 3 and 30 meter links.

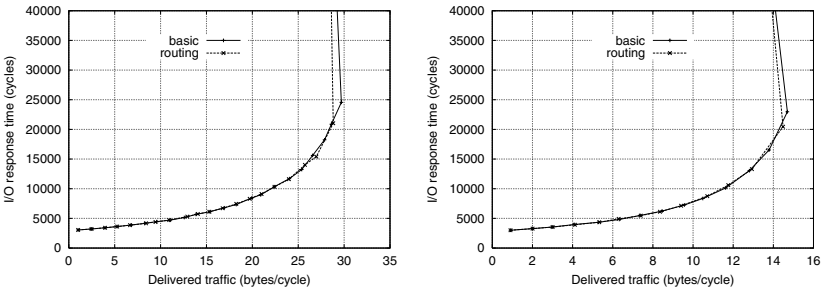
Our case study for the analysis of the floors topologies is a 5-floor topology where each floor has 2 center switches ("floors+" topology). Four arm switches provide connectivity at each floor between the vertical backbone on one hand and servers and disks on the other. Each of the arm switches are connected to a single server and 5 disks. Thus, our storage area network is composed by 20 servers and 100 disks. Links connecting servers or disks to the corresponding arm switch are 3 meters long, while links in the backbone are 30 meters long.

In the case for departmental topologies ("tree" topology), our storage area network is composed of 10 servers and 50 disks. All links in the network are 3 meters long.

5 Performance Analysis

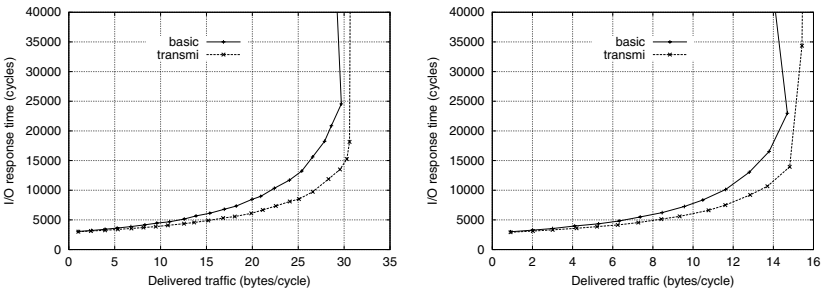
This section analyzes the effect of the strategies proposed in Section 3 on network performance, which is measured by means of the response time of I/O operations and the delivered traffic.

Figure 3 displays the simulation results for the first management policy where incoming control messages are assigned a higher priority in order to be routed first. These results, referred to as "routing", are compared with the performance achieved by the basic switch architecture in Figure 2 and referred as "basic".



(a) floors+ topology (b) tree topology

Fig. 3. Effect of routing control messages with higher priority



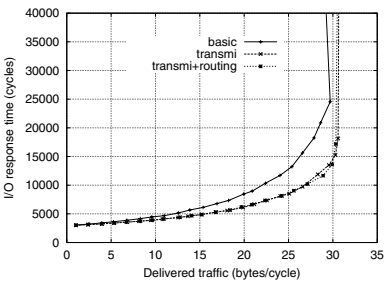
(a) floors+ topology (b) tree topology

Fig. 4. Effect of transmitting control messages with higher priority

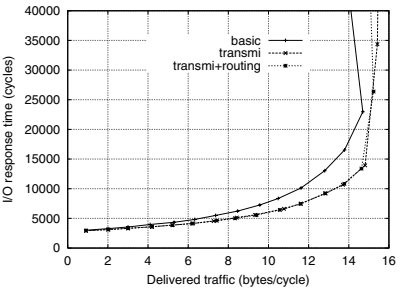
As can be seen, the effect on performance of this strategy is negligible. The reason is that routing incoming control messages first has no advantage if the required output channel is busy. In this case, once the newly arrived control message has been unsuccessfully routed, its priority is reduced, and therefore, it is handled as a regular message in following routing cycles. Note that only when a control message arrives at a switch and the required output link is available, this policy would provide some benefit. Unfortunately, these circumstances are not common.

The effect on network performance of the second management strategy is shown in Figure 4. This policy allows preempting the transmission of data messages in order to transmit control messages once they are routed. Results for this management scheme are referred to as "transmi". As can be seen, the average I/O response time is improved In the case for the floors+ topology, an improvement in I/O response time near 40% is achieved when the network is near saturation. For tree, the improvement is about 30%.

Improvement in I/O response time is due to an important reduction in the latency of control messages, which do not have to wait until the completion of the transmission of data messages. However, improving the transmission of

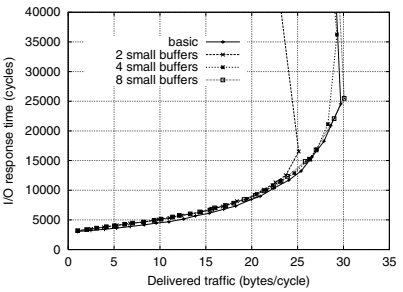


(a) floors+ topology

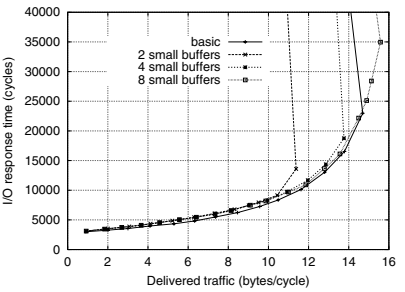


(b) tree topology

Fig. 5. Combined effects of both transmission and routing with higher priority



(a) floors+ topology

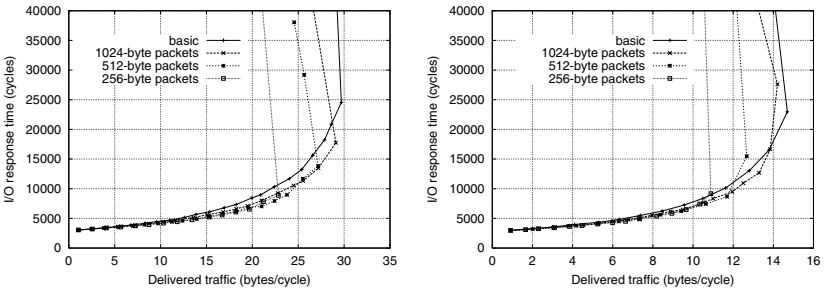


(b) tree topology

Fig. 6. Effect of separating input buffers for both data and control messages

short control messages has a negative effect on the transmission of long data messages, whose latency is increased. This makes that network throughput is not noticeably increased, because the contribution of data messages to the overall network throughput is much more important than the contribution of control messages, due to their smaller size. Therefore, the main effect of transmitting control messages with a higher priority is a reduction in I/O response time.

When control messages are allowed to preempt the transmission of data messages, the network latency achieved by the former is the minimum one if we do not consider the delay introduced by the routing unit due to its round-robin scheduling policy. This delay may be avoided if this management strategy is combined with the first one. In this case, when a control message arrives at a switch, it would be immediately routed and then it would preempt, if necessary, the transmission of the data message being forwarded through the selected output channel. We have analyzed the effect on performance of this new strategy. Figure 5 shows results for the topologies under study. As can be seen, for high loads, the combination of both strategies reports a slight benefit, mainly near network saturation.



(a) floors+ topology (b) tree topology

Fig. 7. Effect of message packetization

Regarding the implementation of the third strategy for managing control messages, switch buffering capacity should be kept similar for all the configurations analyzed. This would provide a fear comparison of the results. The basic 8-port architecture has a storage capacity of eight 2048-byte buffers per port, providing an overall capacity of 128 KBytes. Therefore, the switch architecture implementing this scheme should provide a similar capacity. More concretely, we have assumed eight buffers devoted to data messages, and a variable number of small buffers intended to store control messages, ranging from two to eight buffers. This will result only in an additional maximum buffering capacity of 1% for the larger configuration.

Figure 6 shows the I/O response time versus delivered traffic when 2, 4, and 8 additional buffers are considered, as well as results for the basic switch architecture. As can be seen, when the number of input buffers intended for storing control messages is very low, network throughput is reduced. The reason for this is that the maximum amount of control messages present in the network is limited by the amount of small input buffers. Thus, I/O response time is increased because control messages must wait for a free small buffer before entering the network.

As the number of additional small buffers is increased, control messages reduce their queueing time, and therefore network throughput is improved. When eight additional buffers are intended for control messages, network throughput is slightly increased with respect to the basic switch architecture because of the larger overall buffering capacity available in the network. As more additional small buffers are attached to each input port, the improvement in performance becomes smaller. Finally, it must be taken into account that increasing the number of small buffers will increase the size of the internal crossbar, thus leading to a much more complex design of the switch architecture.

Figure 7 shows the effect of using packetization. Results when messages are split into 256, 512, and 1024-byte packets are plotted, as well as results for the initial architecture. As can be seen, a reduction in the I/O response time is achieved when messages are packetized, independently of the analyzed topol-

ogy. However, network performance is also reduced, being the reduction more noticeable as packet size diminishes.

Finally, message packetization may be combined with the strategy that routes control messages with higher priority, in order to forward control messages sooner. We perform some simulations for such a combination, and conclude that routing control messages with higher priority provides no additional benefit.

6 Conclusions

The main insights provided by our analysis show that SAN performance may be improved if control messages are efficiently handled. More concretely, I/O response time may be lowered at the same time that network throughput is slightly increased when control messages are allowed to preempt the transmission of data messages. In the case for I/O response time, the actual improvement factor depends on network topology, ranging from 10% to 40%. Maximum throughput increment is about 5%.

Regarding the rest of management policies analyzed: routing control messages with higher priority, additional buffers intended for storing control messages, and packetizing data messages, they provide no noticeable benefit, despite of the fact that some of them are costly, as is the case for additional input buffers devoted for control messages.

References

1. N. J. Boden, D. Cohen, R. E. Felderman, A. E. Kulawik, C. L. Seitz, J. N. Seizovic and W. Su, Myrinet - A gigabit per second local area network, *IEEE Micro*, pp. 29–36, February 1995. 906
2. T. Clark, *Designing storage area networks: a practical reference for implementing fibre channel SANs*, Addison Wesley, 1999. 902
3. *User's guide: CSIM18 Simulation Engine (C version)*, Mesquite Software, Inc. 905
4. M. Farley, *Building storage area networks*, McGraw-Hill, 2000. 901, 902
5. P. Kermani and L. Kleinrock, Virtual cut-through: a new computer communication switching technique, *Computer Networks*, vol. 3, pp. 267–286, 1979. 903
6. J. H. Kim and A. A. Chien, Network Performance Under Bimodal Traffic Loads, *Journal of Parallel and Distributed Computing*, vol. 28, no. 1, pp. 43–64, 1995. 902
7. X. Molero, F. Silla, V. Santonja and J. Duato. Modeling and simulation of storage area networks, *Proceedings of the 8th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*. IEEE Computer Society, August 2000. 905
8. X. Molero, F. Silla, V. Santonja and J. Duato. Performance analysis of storage area networks using high-speed LAN interconnects, *Proceedings of the 8th International Conference on Networks*. IEEE Computer Society, September 2000. 906
9. S. S. Owicki and A. R. Karlin. Factors in the performance of the AN1 computer network, Digital SRC research report 88, June 1992. 903
10. F. Silla and J. Duato, High-performance routing in networks of workstations with irregular topology, *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, no. 7, pp. 699–719, July 2000. 904

11. D. Tang, *Storage area networking: the network behind the server*, Gadzoox Microsystems Inc., 1997. 901