A NOVEL STRATEGY FOR CONTROLLING HOT SPOT CONGESTION

Wing S. Ho and Derek L. Eager
Department of Computational Science
University of Saskatchewan,
Saskatoon, Saskatchewan, Canada S7N 0W0

ABSTRACT

In a shared memory multiprocessor system, contention for a particular memory location may lead to a substantial degradation in performance. In systems utilizing a buffered multistage interconnection network, such "hot spot" contention results in a phenomenon known as tree saturation, which adversely affects other traffic in the system. This paper proposes a novel strategy for eliminating tree saturation, with only moderate hardware cost. Simulation results are presented to show that this strategy successfully prevents degradation in the performance of "normal" non-hot spot traffic.

1. INTRODUCTION

Large scale, shared-memory multiprocessor systems are of considerable interest. Such systems commonly employ a multistage interconnection network to connect processor elements (PEs) to memory modules (MMs). Although congestion in such networks is typically minimal if traffic patterns are perfectly uniform, the presence of a "hot spot" (a memory location simultaneously favored by a number of PEs) may seriously degrade performance. As first shown by Pfister and Norton [6], a hot spot may induce "tree saturation", in which a tree of saturated buffer queues extends from the hot MM back to the accessing PEs. Tree saturation is particularly undesirable since it adversely affects "normal traffic" (traffic directed to non-hot MMs).

A substantial amount of previous work has been done that concerns, at least in part, strategies for minimizing the adverse effects of hot spots. *Pairwise hardware combining*, as proposed in the NYU Ultracomputer project [1], adopts a strategy of "combining" two requests directed to the same memory location that meet in a switch. Only a single request is forwarded to the relevant MM, and, when a reply from the MM returns, it is "split" into two replies corresponding to the two combined requests. The proposed strategy is termed "pairwise" combining since a request may be combined with at most one other request in any given switch.

Although pairwise hardware combining has been found able to prevent tree saturation in some contexts [6], Lee, Kruskal and Kuck found that it may be inadequate to cope with hot spot contention in very large scale systems [5]. They proposed *three-way hardware combining* in which at most three requests may be combined into a single request in any given switch. They found that three-way combining has performance close to that of *unbounded combining*, which allows an unlimited number of requests to be combined together in a single switch, and which is guaranteed to prevent tree saturation.

The use of a *software combining tree* has been proposed to redistribute hot spot accesses over a tree of data items that can be dispersed among many MMs [8]. This strategy has the advantage of avoiding the high cost and complexity of hardware combining, while achieving similar performance, but may not be as generally applicable.

Another approach to the hot spot problem is to eliminate tree saturation by eliminating all buffers, as is done in the circuit-switched BBN Butterfly™ [7]. An additional stage of switches is provided in the Butterfly network to provide two unique paths from each PE to each MM. When there is contention for a switch, one of the two requests is discarded and must be retransmitted using the alternate path while the other is allowed to proceed. Experimental results show that the network performs well under light load. However, performance degradation is experienced when the network is placed under heavy load due to the lack of buffers and the circuit switching operation.

This paper proposes a novel strategy for use in buffered packet-switched networks, that eliminates tree saturation, and thus also degradation in the performance of non-hot spot traffic, with only moderate hardware cost. Switch operation is similar to that in a conventional network, except that when two packets destined for *the same memory location* contend at a switch (i.e. simultaneously attempt to move into the same output port buffer queue), one is discarded. Simulation results are presented showing that the proposed "discard" strategy eliminates tree saturation by preventing monopolization of the network by hot spot traffic. Results for a conventional network and for a network utilizing pairwise hardware combining are presented for comparison purposes.

Section 2 of this paper describes in detail the strategies studied, as implemented in simulation models. Section 3 contains sample simulation results. Section 4 presents the conclusions.

2. STRATEGIES AND MODELS

The conventional network model utilizes what is termed here the "FIFO" strategy to emphasize its uncontrolled "first-in-first-out" approach to allocating switch buffer resources, and is described in Section 2.1. Sections 2.2 and 2.3 describe pairwise hardware combining, and the discard strategy, respectively, as implemented in simulation models.

This research has been supported in part by grants from the Natural Sciences and Engineering Research Council of Canada

2.1 Conventional Network Model

A buffered, packet-switched Omega network [4] is employed. It is assumed that a memory request will always fit within a single packet. Both a "forward" network handling traffic from the PEs to the MMs and a "return" network are required in a real system; only the forward network is modelled in the simulations since it is typically only this network in which congestion occurs. The return network is reflected in the results by adding an appropriate number of network cycles to the reported total network delays.

In an Omega network built with "*2 by 2*" switches (switches with two input and two output ports), there are log_2N "stages" of switches that are required to interconnect N PEs and N MMs. Each stage is composed of $N/2$ identical switches. With the FIFO strategy, a packet received by a switch is buffered at the end of one of two buffer queues, depending on which output port is desired, and forwarded to the connected switch in the next stage when the packet reaches the head of its queue.

When an attempt is made to simultaneously forward two packets from two different switches to the same output port of a switch in the next stage, a *contention* is said to have occurred. In this event, one packet is randomly selected and is buffered in the output port buffer queue if the buffer queue has space for at least one packet (it is assumed, however, that a buffer space does not become usable until the network cycle after that in which it is emptied). The other packet stays in its current location and waits for the next network cycle to attempt again to advance. Note that other variations of switch designs may allow both contending packets to be buffered at the output port if there is sufficient buffer space available.

2.2 The Pairwise Hardware Combining Strategy

With pairwise hardware combining, when a contention occurs, and if the two contending packets are directed to the same memory location, they are combined into one single packet and the combining is recorded in the *wait-buffer* of the receiving switch [1]. Moreover, if two packets directed to the same memory location are in the same output port queue in the same network cycle, they are also combined, as long as neither packet has been previously combined in that same switch. A combined packet can be combined with any other packet directed to the same memory location in the switches of later stages. In all other respects, our model of a combining network is identical to the conventional network model described in Section 2.1.

2.3 The Discard Strategy

The discard strategy is best thought of as a "congestion control method" that attempts to avoid monopolization of resources (i.e. buffer spaces and links) by hot spot traffic. When a contention occurs at a switch and the contending packets are destined for the same memory location, one packet is randomly chosen to be *discarded* while the other is forwarded to the appropriate output port buffer. Note that the source PE of the discarded packet must at some point retransmit the discarded packet. In all other respects, our model of a network utilizing the discard strategy is identical to the conventional network model described in Section 2.1.

Retransmission can be either *switch-initiated* or *PE-initiated*. In the former, a switch that discards a packet sends back a special packet to the source PE signaling that its packet was discarded. Upon receiving the special packet, the PE immediately retransmits its request. With PE-initiated retransmission, on the other hand, the source PE retransmits a packet if a reply has not arrived after the expiration of a pre-determined timeout period. Switch-initiated retransmission is employed in the simulation experiments.

3. PERFORMANCE RESULTS

This section presents results obtained from computer simulation of networks utilizing the FIFO, discard and combining strategies discussed in Section 2. Our experimental methodology is described first in Section 3.1. Results for a "base case" system are reported in Section 3.2.

3.1 Experimental Methodology

A series of time-driven simulation experiments were performed, in which the unit of time was the "network cycle time"; i.e., the time required for a packet to be transmitted from one switch to another. The "base case" system in these experiments consists of 1024 PEs and MMs interconnected by a packet-switched Omega network built with 2 by 2 switches. The network thus has 10 stages. Each switch includes 4 buffers per output port. A PE can have at most one memory request outstanding at a time; i.e., a PE cannot generate a new packet unless a reply for the previous one has been received. (Although this might seem to unduly restrict workload generation, the results that are shown later demonstrate that, in the presence of hot spot traffic, the network loads achieved are in fact sufficient to cause major performance degradations.) When a PE is not waiting for a reply, it generates a new packet in a given network cycle with an input probability that is termed the "offered load". It is assumed that an MM requires one cycle each to receive a packet, process it, and forward it to the first switch on the return path. However, the last step may be performed concurrently with either of the previous steps, for different packets.

A single memory hot spot is assumed. It was desired to model a context in which not all PEs were contending for the hot spot; a more realistic situation was desired in which some PEs were generating only "normal" traffic. Rather than statically designating some PEs as "hot spot" PEs, instead the identity of those PEs accessing the hot spot was allowed to vary by simply fixing the total number of PEs that may have hot spot packets outstanding to 512. Whenever this number is less than 512, a new packet generated by a PE is destined for the single hot spot with an input probability h, that is termed the "hot spot rate". Otherwise, the PE is constrained to generate only non-hot spot "normal" traffic. The non-hot spot "normal" traffic is uniformly distributed over the 1024 MMs. For the discard strategy, it is conservatively assumed that two "normal" packets destined for the same MM reference the same location with probability $1/32$.

Although there are many parameters and assumptions in the base case model, many variations of this model have been simulated; additional results are found in [2]. In all cases, however, the results were found to be similar in nature to those for the base case, reported in Section 3.2.

The main performance metrics that are considered are the average total packet delay (in network cycles) and the achieved throughput. The throughput is reported as a percentage of that which would be achieved if no queuing occurred in the network or at the MMs. For the discard strategy, the time delay incurred by a discarded packet, including the time delay in notifying the source PE, is carried over to the newly retransmitted packet. Our computed network delay for the discard strategy hence also includes the time cost of packet discards.

3.2 Principle Performance Comparison

For each of the three strategies considered, Figure 1 shows the average total network delay in the base case system (including the fixed delay in the return network) versus the achieved throughput, for a number of values of the hot spot rate. As a curve in one of these graphs is followed from its lowest, left-most point, successive points correspond to increased offered loads. Note that the minimum delay in the base case system is 23 network cycles.

As shown in Figure 1(a), the hot spot rate has a major impact on the network delay experienced with the FIFO strategy. A higher hot spot rate results in more hot spot packets being queued in the interconnection network and worsens the tree saturation effect. Combining (Figure 1(b)) performs the best among the three strategies and retains average delays comparable to the minimum possible, except for high loads and high hot spot rates where some significant degradation is seen. However, Combining shares with the FIFO strategy the undesirable property that throughput may actually decline with increasing load.

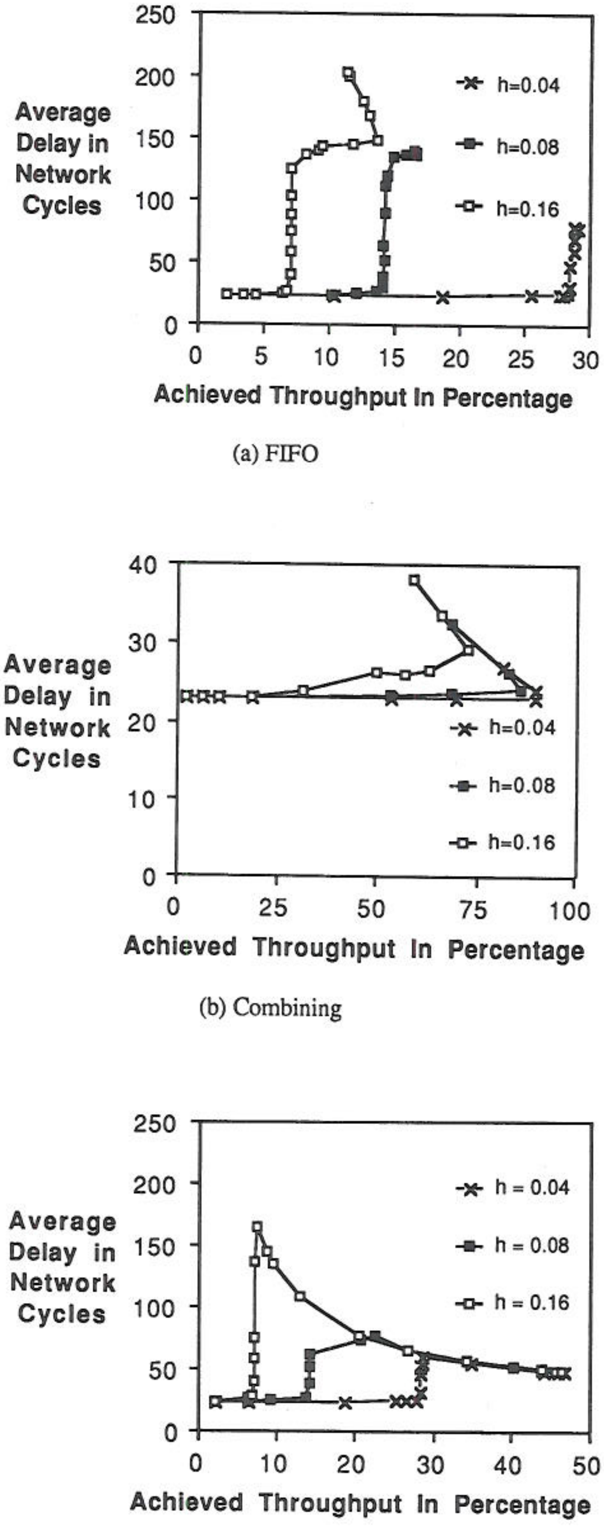
Under low load, the average delays with the discard strategy (Figure 1(c)) are nearly identical to those with the FIFO strategy, and increase sharply at about the same offered load (and achieved throughput) as that at which tree saturation first occurs in the FIFO strategy. However, as is seen in Figure 2(c), the average delay of normal traffic is near the minimum possible over the entire range of offered loads examined, unlike the case with the FIFO strategy. These average delays for normal traffic indicate that tree saturation does not occur under the discard strategy. As seen in Figure 1(c), the overall average delays with the discard strategy actually decrease, and higher throughputs are achieved, as the offered load is increased further, since performance is increasingly dominated by the (good) performance of the increasing volumes of normal traffic. Performance is relatively insensitive to the hot spot traffic rate under higher offered loads. Finally, note that the discard strategy provides much superior throughput capacity in comparison to the FIFO strategy and that the throughput never declines with increasing offered loads.

The results in Figure 1 indicate only "overall" network performance. The advantages of the discard strategy in comparison to the FIFO strategy are, however, best seen by considering "hot spot" and "normal" traffic separately, as is done in Figures 2 and 3. As shown in Figure 2(a), a very large average delay for normal traffic is incurred with the FIFO strategy. This is due to the tree saturation effect; normal traffic that has to cross the tree is slowed down to a rate dependent on the service rate of the single hot MM [3]. Figure 2(c) reveals that the average network delay of normal traffic in the discard strategy is close to that with the combining strategy (Figure 2(b)) which achieves a near-to-minimum delay.

It is important to realize that the greatly improved performance afforded to "normal" traffic with the discard strategy (in comparison to that with the FIFO strategy) is not gained at the expense of the throughput of the hot spot traffic. As shown in Figure 3(c), the hot MM is still kept as busy as possible. (The reported utilizations are so high with the combining strategy because, with combining, each packet processed by the hot MM may actually correspond to a number of memory requests. The "utilization" is defined as the rate of servicing memory requests multiplied by the packet service time at the MM.)

4. CONCLUSIONS

This paper has proposed a novel strategy for controlling hot spot congestion in multistage interconnection networks. This strategy is based on discarding one of two packets that contend for the same output port buffer queue in a switch, if both packets are destined for the same memory location. Extensive simulation results show that use of the proposed "discard" strategy yields much superior performance to that in a conventional network, when hot spot traffic is present. For non-hot spot "normal" traffic, performance is close to that achieved with the much more costly pairwise hardware combining strategy. The benefits for normal traffic are not achieved by unduly penalizing hot spot traffic; the hot MM is still kept fully utilized when the discard strategy is employed.

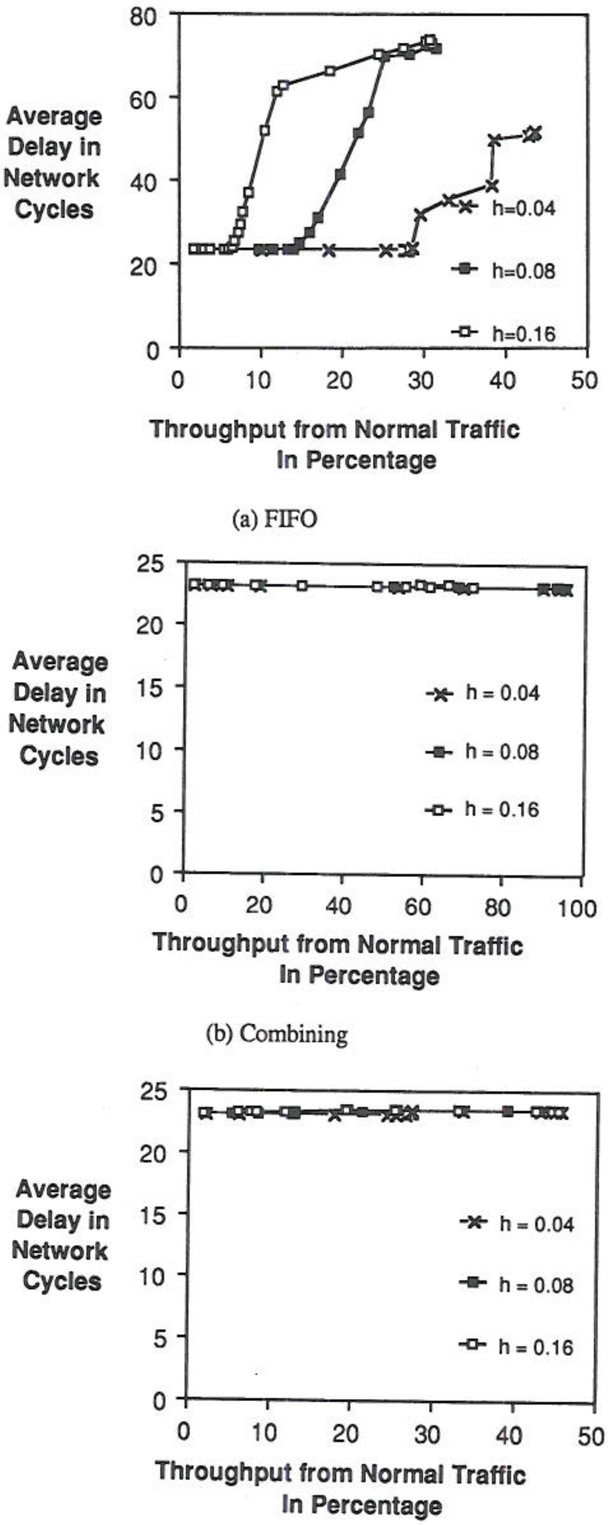


(a) FIFO

(b) Combining

(c) Discard

Figure 1.
Delay Throughput Profiles (1024 PEs and MMs, a Buffer Queue Size of 4)

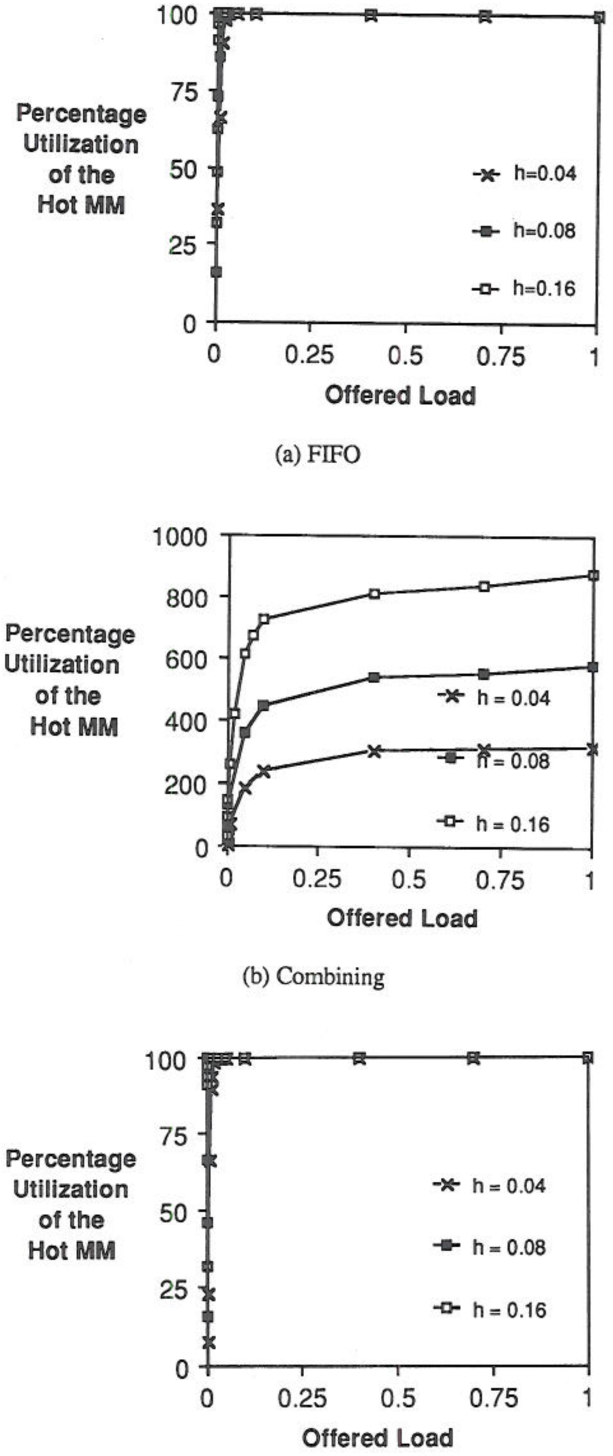


(a) FIFO

(b) Combining

(c) Discard

Figure 2.
Delay Throughput Profiles for Normal Traffic (1024 PEs and MMs, a Buffer Queue Size of 4)



(a) FIFO

(b) Combining

(c) Discard

Figure 3.
Utilization of Hot MM Versus Offered Load (1024 PEs and MMs, a Buffer Queue Size of 4)

REFERENCES

[1] A. Gottlieb, R. Grishman, C. P. Kruskal, K. P. McAuliffe, L. Rudolph and M. Snir. The NYU Ultracomputer - Designing an MIMD Shared Memory Parallel Computer. *IEEE Transactions on Computers 32, 2 (Feb. 1983)*, 175 - 189.

[2] W.S. Ho. Congestion Control in Shared Memory Multiprocessor Systems. *M.Sc. Thesis, Research Report 89-1, University of Saskatchewan, Jan. 1989.*

[3] M. Kumar and G. F. Pfister. The Onset of Hot Spot Contention. *Proceedings of the 1986 International Conference on Parallel Processing (Aug. 1986)*, 28 - 34.

[4] D. H. Lawrie. Access and Alignment of Data in an Array Processor. *IEEE Transactions on Computers C-24, 12 (Dec. 1975)*, 1145 - 1155.

[5] G. Lee, C. Kruskal and D. J. Kuck. The Effectiveness of Combining in Shared Memory Parallel Computers in the Presence of "Hot Spots". *Proceedings of the 1986 International Conference on Parallel Processing (Aug. 1986)*, 35 - 41.

[6] G. F. Pfister and V. A. Norton. "Hot Spot" Contention and Combining in Multistage Interconnection Networks. *Proceedings of the 1985 International Conference on Parallel Processing (Aug. 1985)*, 790 - 797.

[7] R. H. Thomas. Behavior of the Butterfly Parallel Processor in the Presence of Memory Hot Spots. *Proceedings of the 1986 International Conference on Parallel Processing (Aug. 1986)*, 46 - 50.

[8] P. Yew, N. Tzeng and D. Lawrie. Distributing Hot-Spot Addressing in Large-Scale Multiprocessors. *Proceedings of the 1986 International Conference on Parallel Processing (Aug. 1986)*, 51 - 58.