MINC : Multistage Interconnection Network with Cache control mechanism

Toshihiro Hanawa Hideki Yasukawa Katsunobu Nishimura Hideharu Amano
Department of Computer Science
Keio University
3-14-1 Hiyoshi, Kohoku-ku, Yokohama, 223 JAPAN

Abstract

A novel approach to the cache coherent Multistage Interconnection Network (MIN) called the MINC (MIN with Cache control mechanism) is proposed. In the MINC, the directory is located only on the shared memory using the Reduced Hierarchical Bit-map Directory schemes (RHBDs). In the RHBD, the bit map directory is reduced and carried in the packet header for quick multicasting without accessing directory in each hierarchy. In order to reduce unnecessary packets caused by compacting the bit map in the RHBD, a small cache called the pruning cache is introduced in the switching element. Through the simulation, it appears that the pruning cache works most effectively when it is provided in every switching element of the first stage, and it reduces the congestion from 50% to 90% only with 4 entries.

Keywords: MIN, Coherent cache, Directory scheme, Multiprocessor, Congestion analysis

1 Introduction

Multistage Interconnection Networks (MINs)[2] have been well researched as an interconnection mechanism between processors and memory modules of medium scale parallel machines. Unlike the shared bus which allows to access one memory module at a time, the MIN permits to access to a number of memory modules simultaneously. However, in a large scale system, since the number of stage increases, the latency for accessing the shared memory also increases. As a result, the MIN becomes a bottleneck in the system performance.

For this solution, providing a private cache between a processor and the MIN is the most efficient approach to reduce the latency. However, it brings the coherent problem for the shared writable data. The snoop cache commonly used in shared bus connected multiprocessors cannot be used because of the benefit of the MIN, that is, multiple memory modules can be accessed simultaneously. The directory scheme based on the one-to-one message passing which is used in the CC-NUMA machines[10][3] is difficult to be implemented in the MIN connected multiprocessors. Instead of them, the approach which is supported by the pre-execution analysis with the compiler has been studied[14][4]. Although the required hardware cost is small, this approach is not suitable for general purpose usage by multiple users.

On the other hand, recent VLSI technology allows the switching element to provide the sophisticated structure and control. By making the best use of them, the MIN which has cache or its directory inside the switching elements has been proposed[12][13]. In this approach, the tree structure in the MIN is effectively utilized to maintain the coherent cache. However, traditional methods require a large memory outside the switching element. It introduces not only a large additional hardware but also the extra latency which required to access the memory outside the element.

Here, a novel approach to the cache coherent MIN is proposed. In this method, the directory is located only on the shared memory using the Reduced Hierarchical Bit-map Directory schemes (RHBDs). In the RHBD, the bit map directory is reduced and carried in the packet header for quick multicasting without accessing directory in each hierarchy(stage). In order to reduce unnecessary packets caused by compacting the bit map in the RHBD, a small cache called the pruning cache is introduced in the switching element. Since this cache is small enough to implement inside the switching element, the packet transferred through the MIN is unnecessary to access the memory outside the switching element. We call the MIN with these mechanisms the MINC (MIN with Cache control mechanism).

In Section 2, the hierarchical directory method and traditional approaches are described, and the problems are pointed out. The structure and control of

the MINC are introduced with the RHBD and pruning cache in Section 3. In Section 4, results of simulation studies are presented to demonstrate the efficiency of the MINC.

2 Hierarchical bit-map directory scheme

In order to maintain the coherent cache without using snoop cache, the directory is needed to indicate which processor has the copy of line in its private cache. For the CC-NUMAs with point-to-point network, several directory schemes including the full-map directory[11], limited pointer[3], chained directory[8], and dynamic pointer[10] are proposed and used. However, for multiprocessors with hierarchical network, the hierarchical bit-map directory scheme is advantageous. For the reason that, cache coherent messages can be efficiently multicast with making the best use of communication locality. Since every MIN provides the fat-tree structure, the hierarchical bit-map directory scheme can be naturally used.

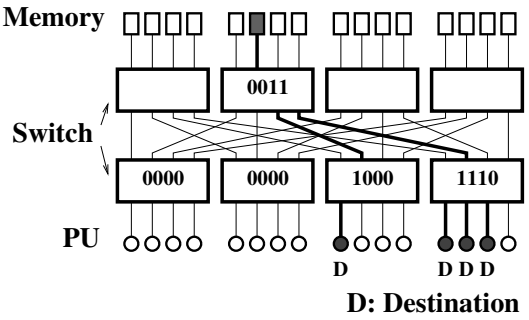

Figure 1: Hierarchical bit-map directory scheme on the MIN

Figure 1 illustrates the concept of hierarchical bit-map directory scheme on the MIN. Leaves of the tree correspond to processors and a message is first sent from the root of the tree. Each node can multicast a message to its multiple branches at a time. To specify the destination leaves precisely, an n-bit bit map is required for each node when an n-ary tree is used. Thus, for m height n-ary tree, total of $\sum_{k=1}^{m} n^k$bits are required for each entry. Although the required amount of memory is larger than that of the full-map directory scheme in which n^m bits are required (since there are n^m leaves), the message can be multicast using the tree structured path in the hierarchical bit-map directory scheme.

Approaches to cache coherent MIN have been done based on this tree structure and hierarchical bit-map directory scheme. In the MHN (Memory Hierarchy Network) scheme[12], a switching element holds both directory and data memory. The shared line moves between hierarchies to minimize the access latency. However, this method requires a large amount of memory in each switching element and it allows only single copy of shared read/write blocks in the private cache.

In the MIND (MIN with Directory) scheme[13], only directory is stored in the switching element. Hierarchical bit-map directory is implemented so as to minimize the network traffic for cache coherence. In order to multicast easily with a small hardware, MBN(Multistage Bus Network) [1] is proposed by replacing a switching element of the MIND with a bus. However, these methods require a large amount of memory for hierarchical directory outside the switching element. Since multicasting the cache coherent messages requires the directory access outside the switching elements, the latency of the message transfer is stretched.

3 MINC(MIN with Cache control mechanism)

3.1 Overview of the MINC

Here, a novel cache coherent MIN called the MINC is proposed. Like usual MIN connected multiprocessors, forward MIN and backward MIN are provided in the MINC as shown in Figure 2. Caches are provided between the MIN and Processing Units (PUs) which may have a local memory for private data. Any types of the MIN can be utilized as the forward MIN which carries address and writing data from processors to memory modules. The reading data and cache coherent messages are multicast through the backward MIN specialized for the MINC.

The MINC is based on the following two key ideas:

RHBD scheme The bit map of the hierarchical directory is reduced and equipped only in the main memory module. The coherent message is multicast based on the reduced bit map attached to the packet header. This technique is called the Reduced Hierarchical Bit-map Directory scheme (RHBD).

pruning cache In order to reduce unnecessary packets generated with the RHBD, a small cache mem-

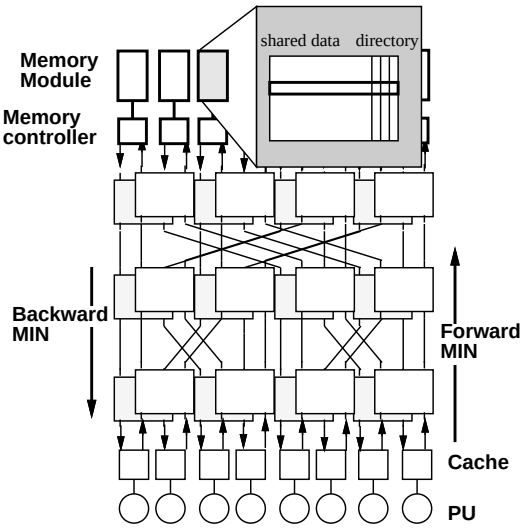

Figure 2: The structure of the MINC

ory called the bit map pruning cache is introduced inside the switching element in the backward MIN.

Here, the RHBD is introduced first, then, coherent mechanism of the MINC is described. Finally, the bit map pruning cache is proposed.

3.2 The RHBD

The basic concept of the Reduced Hierarchical Bit-map Directory scheme (RHBD)[7][9] was proposed for a massively parallel processor JUMP-1[6]. Although the RHBD for JUMP-1 is designed for a large scale cache coherent NUMA, this concept can be easily applied to the MIN connected multiprocessors.

In this scheme, the bit map is reduced using the following techniques:

- use the common bit map for all nodes of the same level of hierarchy, and
- a message is sent to all children of the node (thus, broadcasting) when the corresponding bit in the map is set.

The reduced directory is not stored in each hierarchy but stored only in the root. Message multicast is done according to the reduced bit map attached to the message header. Using this method, since multicast does not require to access the directory in each hierarchy, quick message transfers can be performed.

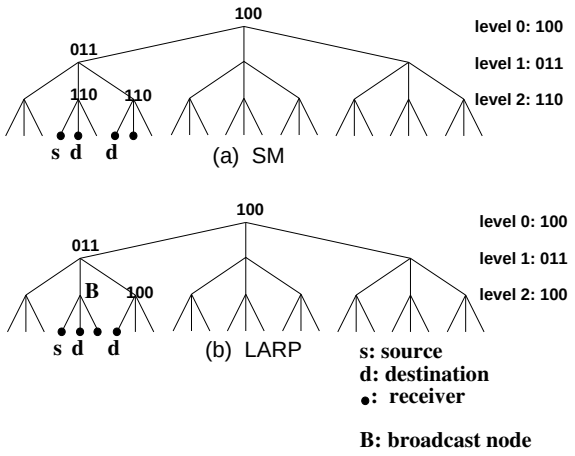

Figure 3: Hierarchical bit-map directory schemes

By the combination of the above two techniques, two schemes, the SM, and LARP are delivered:

SM scheme: Figure 3(a) shows an example of this scheme for the 3-ary tree. In this figure, 's' is the source node, and 'd' is the destination to which packet is sent. • indicates nodes which receive data. In the SM(Single Map) scheme, all nodes at a level use a unique bit map as shown in Figure 3(a), and thus, no broadcast is made (unless a bit map is all-1). This scheme is advantageous when the number of destination is not so large.

LARP scheme: In the LARP(Local Approximate Remote Precise) scheme, the bit map is used only nodes which is the root of the subtree far from the source node. Using this scheme, the bit map is used for remote nodes of the source node, while the message is broadcast in the remote subtree marked B. In this scheme, broadcast is done at nodes which are the root of the subtree including the source node while a unique bit map is used in other subtrees like the SM scheme. Note that the broadcast is started from children nodes whose parent starts the multicasting. In the example shown in Figure 3(b), since the multicast starts at the level 1 (The top node (level 0) only sends a message to a child.), the broadcast (marked **B**) starts at the level 2 to the subtree which includes the source node. In this case, the broadcast is done for local nodes of the source node, while the bit map is used for remote nodes. This scheme is advantageous when the mapping can make the best use of the locality of communication.

Unlike the hierarchical bit map directory method which requires $\sum_{k=1}^{m} n^k$ bits for each entry, the RHBD only requires nm bits for each entry. However, PUs which are marked with • but without 'd' receive unnecessary packets or messages for compacting of the bit map. Although unnecessary messages are just discarded in the cache controller of the PU, it may cause the congestion of the network.

3.3 The coherent control of the MINC

The directory is required for each cache line in the main memory modules, and managed by the memory controller. When the coherent message is required, the bit map is cut out from the directory, pushed into the packet header, and directly used as the multicast bit map.

A counter field which stores the number of PUs holding the copies of the cache line is required when update protocol is used. If only invalidation protocol is used, a bit flag indicating whether the line is shared with multiple PUs or not is required instead of the counter.

On the other hand, two status bits are required to the tag of cache provided between the MIN and PU. Like other cache protocols, the status bits represent:

- Valid/Invalid (V/I) and
- Shared/Private (S/P).

As shown in Figure 4, the cache coherent protocol of the MINC is based on the write through policy with three states I: Invalidate, P: Private, and S: Shared. Both invalidate type protocol (Figure 4a) and update type protocol (Figure 4b) are available. Although the update type protocol requires a lot of traffic, it will work efficiently for scientific computation which requires a lot of data multicast.

When the read access hits to the cache, the data is directly read out from the cache. For other cases, the cache, memory controller, and MINs behave as follows.

Write hit/miss-hit: If the write request hits to the cache, the cache is updated and the state is changed into 'P'. Since the write through policy is adopted, the address and data are transferred to the memory module through the forward MIN. After updating the memory module, the *invalidation message* or *data for update* is multicast using the backward MIN according to the bit map of the directory. When the invalidation protocol is used, all bits of the directory are reset and the counter field (flag) is cleared.

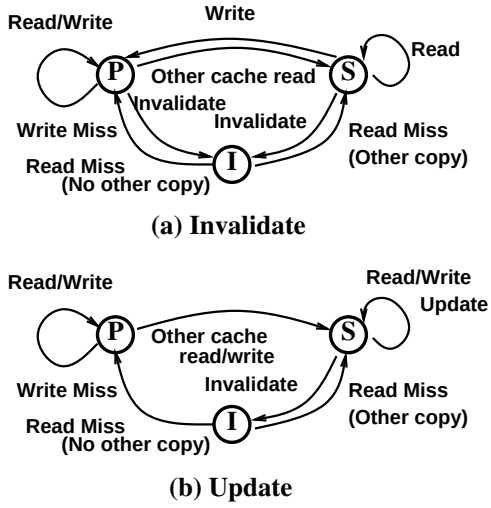

Figure 4: State transition graph of the MINC

Read miss-hit: If the invalidated (state I) line is not found in the cache, a line is selected and swapped out according to the **swapping out** operation described later. Then, the cache controller sends a *line request packet* to the memory module through the forward MIN. The memory controller receives requesting PU number in the line request packet, and sets the corresponding bit of the directory. At the same time, the controller reads the line and sends it to the requesting PU with *the line read-out packet* through the backward MIN. Here, the following operations are required depending on the value of the counter (flag).

- If the value of the counter is zero, no other copy is existing. In this case, the private bit in the header of the line read-out packet is set. A cache controller which receives the packet whose private bit is set changes the line state into 'P'.
- If the value of the counter is one, *the sharing message* is multicast after sending the line read-out packet also through the backward MIN according to the bit map. A cache controller which receives this packet changes the corresponding line state into 'S'.

In both cases, the counter is incremented if update protocol is used.

Swapping out: Since a bit in the directory may represent multiple copies in the RHBD, the bit in the directory must be reset carefully.

When a line is discarded for replacing with another line, *the discarding message* is sent to the memory module through the forward MIN in the update protocol. The memory controller which receives this message decrements the counter. When the value becomes zero, all bits of the directory is cleared. In the invalidation protocol, since the directory is frequently cleared by each data writing, the discarding message is unnecessary. In this case, the line for replacing is just discarded without any operation.

3.4 Bit map pruning cache

The major disadvantage of the RHBD is that it requires unnecessary message multicast since it only uses a single bit map or broadcasting in each level of the hierarchy. From the simple evaluation, the RHBD generates more than 100 times messages as really required messages in the worst case[9].

To cope with this problem, "the bit map pruning cache (P.C.)" is proposed in the switching element of the backward MIN. The bit map pruning cache is a small cache of the bit map referred by the line address of multicasting packets. When the line read-out packet is transferred through the backward MIN, the pruning cache is referred by the line address in the packet header. If the address is matched, the bit corresponding to the output label of the switching element is set as shown in Figure 5. Otherwise, an entry is discarded and the new entry is registered. After the registration, the bit corresponding to the output label of the switching element is set.

When the invalidate message or updating data is multicast, the bit map pruning cache is checked. If there is an entry which matches the address, the bit map of the pruning cache is utilized for multicasting, otherwise, multicast is done according to usual RHBD schemes. Using this mechanism, unnecessary packets are pruned according to the bit map in the cache. If this buffer is used in each hierarchy, the bit map which is the same as the original hierarchical bit map directory method is generated temporarily in each switching element of the backward MIN. In invalidation type protocol, the entry of the pruning cache is removed after multicasting of the invalidation message. Note that, the miss-hit of the pruning cache just causes unnecessary packets.

Since the bit map pruning cache must be implemented inside the switching element, the size is limited. If the entry of the cache is full at the registration, an entry is selected and removed. Two selection policies, the FIFO policy for the invalidation type protocol, and the LRU policy for updating protocol are used respectively.

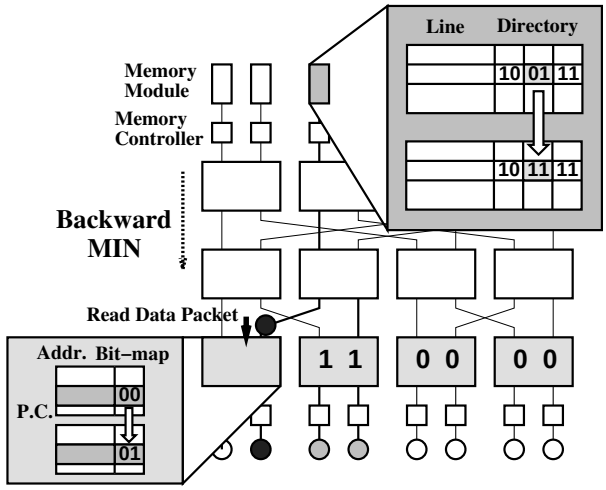

Figure 5: Bit map pruning cache (Only provided in the first stage)

For reducing the required hardware and latency caused by accessing the pruning cache, the bit map pruning cache can be omitted in some stages. From the simulation results shown later, it appears that the satisfactory effect is achieved with the pruning cache provided in switching elements of only a stage as shown in Figure 5.

4 Performance Evaluation

4.1 Simulation Model

The network traffic and congestion caused by the coherent messages are evaluated with a simple probabilistic model. The effect of the RHBD and pruning cache in the MINC are sensitive to the message traffic pattern. Here, the following two traffic models are used for deciding the accessing address:

- Nonlocal: a processor accesses to every memory module with the same probability.
- Local: the destination memory module of a packet from a processor p follows a normal distribution whose center is p and standard deviation $SD = 1$. Thus, a processor p accesses memory module p or neighbors more frequently than other memory modules.

The control model of the forward/backward MIN is based on the SSS(Simple Serial Synchronized)

MIN[5].In this control model, packets are transferred through a few bit serial lines synchronized with a common frame clock from input buffers prepared between PUs and the MIN. Since each switching element stores only one bit (or a few bits) of the packet, the SSS-MIN behaves like a set of shift registers with the switching capability.

When a conflict occurs, one of the conflicting packets is routed to the incorrect direction since the SSS-MIN provides no packets buffer in each switching element. The conflict bit in the routing tag is set when the conflicting packet is routed to the incorrect direction. Such a packet is treated as a dead packet, and never interferes other packets. The conflicting packet is inserted again in the next frame from the input buffer.

Although the MINC can be implemented on the traditional MIN which provides the packet buffer inside the switching element, the control mechanism for multicast becomes much simple in the SSS-MIN. In the backward MIN, the bit map of the directory is copied to the controller when a coherent message is multicast. Since multicast message may conflict with other messages, only bits corresponding to the successful message is reset, and the multicast is repeated until all bits are cleared.

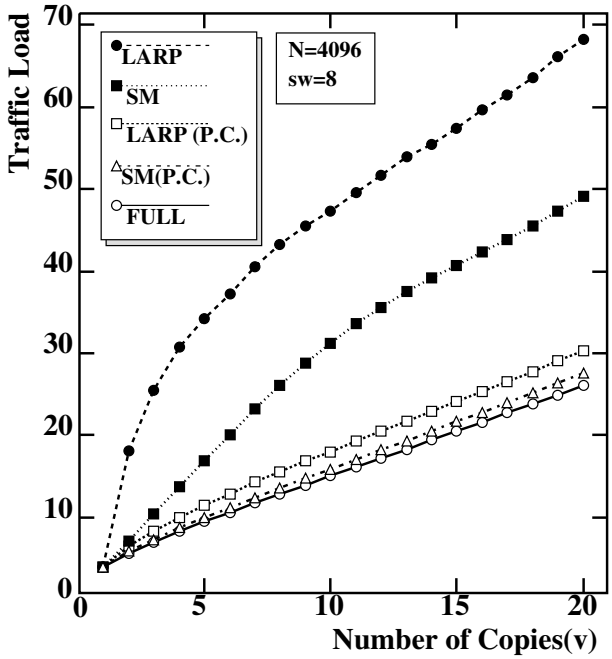

Figure 6: Traffic load versus the number of copies in the system

4.2 Traffic load

First, the number of packets transferred in the backward MIN for coherent control is evaluated. Here, we call the total number of packets which are transferred on links between switching elements in the backward MIN the *traffic load*. If a large number of unnecessary packets is generated with the RHBD, the traffic load increases and the congestion of the backward MIN will degrade the performance.

Figure 6 shows the traffic load versus the number of copies (the number of PUs which share a cache line). The results of four schemes, the LARP, SM, LARP with the pruning cache (LARP/P.C.), and SM with the pruning cache (SM/P.C.) are shown in the figure with ideal case (IDEAL) which includes no unnecessary packets. In this case, the bit map pruning cache is only provided in switching elements of the first stage (the nearest stage to the PUs). The *Local* traffic model is used on the system with 4096 PUs. The backward MIN is 4-stage baseline network with 8×8 switching elements.

Figure 6 shows that the traffic load of the LARP is worse than that of the SM even with the pruning cache (LARP/P.C.). The traffic load is especially large with a large number of copies. From this evaluation, it appears that the SM is superior to the LARP in the MINC. Unlike the interconnection network used in JUMP-1[9], the backward MIN of the MINC includes complete 8-ary trees. In such trees, depending on the location in the tree, some messages should have transferred through the very root of the tree just to share the data with the neighboring node. Broadcast used in the LARP generates a large number of unnecessary packets in such cases.

The bit map pruning cache much improves the traffic load in both cases. The traffic load of the SM with the pruning cache is close to that of the ideal case.

4.3 The congestion of the backward MIN

In order to evaluate the performance considering packets conflict, the behavior of a switching element, memory module, and PUs are simulated in detail. Table 1 shows parameters used in the simulation. Since a large execution time is required for detail simulation, the target system with 256 PUs is evaluated. In this case, the MIN is the 3-stage baseline network with 4×4 switching elements. Considering the implementation, the size of the bit map pruning cache (PCsz) is set to be 4. Other parameters are fixed as shown in Table 1.

In the SSS-MIN, if there is a lot of conflicts, multi-

Packet generation	
Pr	The probability that each processor issues a packet per one flame (0.2 – 0.35)
r	The ratio of the reading access (0.8)
System parameters	
N	System size(256)
sw	Switch size(4x4)
PCst	The location of the bit map pruning cache (0)
PCsz	The size of the pruning cache (4)
Memory parameters	
GM	The size of the shared memory size[bytes](64M)
BS	The size of the cache line size[bytes](16)

Table 1: Simulation parameters

cast must be retried many times. Thus, the congestion of the MIN is represented with the average number of frames required for the completion of a multicast. Here, the stage number is in the ascending order from the nearest one to the processor. Here, the S-M scheme is only evaluated since it appears that it is superior to the LPRA with the traffic load analysis. Two possible protocols: invalidate (inv.) and update (upd.) are evaluated under two traffic patterns. The traffic patterns are mixtures of the above mentioned traffic models:

- global dominant (GD) includes 90% *non-local* mode traffic and 10% *local* mode traffic, and
- local dominant (LD) includes 10% *non-local* mode traffic and 90% *local* mode traffic.

Figure 7 shows the average number of required frames versus the packet generation probability by a PU. With a large probability, the number of frames increases. The congestion becomes severe especially when the probability increases beyond 0.35. However, this probability is unrealistic in systems with the reasonable amount of private cache. This figure also demonstrates the effect of the bit map pruning cache. The small pruning cache (4 entries) improves the number of required frames from 60% to 80%.

From this figure, it appears that the congestion of the backward MIN is severe when the update protocol is used. While the bit map of the directory is cleared when data is written into the corresponding line in the invalidation protocol, the bit map remains until all copies are swapped out in the update protocol. Therefore, the number of multicast required for the update

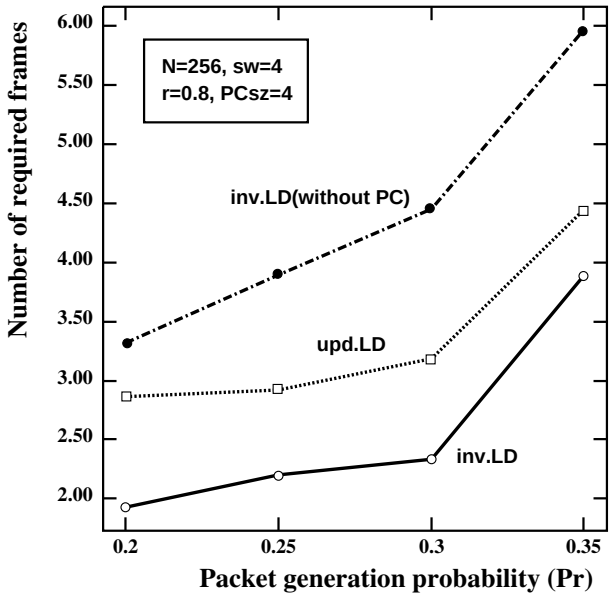

Figure 7: Average number of required flames versus packet generation probability

protocol is larger than that of the invalidation protocol. From this evaluation, the invalidation protocol seems to be better than the update protocol. However, the update protocol will be advantageous in scientific calculations which require a large number of data multicast. The trace driven or execution driven simulation based on the practical application programs is required for the further comparison.

Table 2 shows the hit ratio of the bit map pruning cache provided in switching elements of the first stage. Needless to say, the hit ratio under the local dominant traffic is higher than that of the global dominant one. Under the local dominant traffic, a high hit ratio is achieved only with 4 entries. In the invalidation protocol, the pruning cache is cleared when the invalidation message is multicast. Since only active entry is held in the pruning cache, the hit ratio of the invalidation protocol is higher than that of the update protocol.

inv.LD	inv.GD	upd.LD	upd.GD
77.87%	48.52%	54.20%	34.22%

Table 2: Pruning cache hit ratio

5 Conclusion

A new cache coherent MIN called the MINC is proposed. By the combination of the RHBD and bit map pruning cache, each switching element can multicast coherent messages quickly without accessing outside memory nor generating large number of unnecessary packets.

Through the simulation, it appears that the SM (Single Map) scheme of the RHBD is suitable for the MINC. The pruning cache works most effectively when it is provided in every switching element of the first stage, and it reduces the congestion from 50% to 90% only with 4 entries. Although the update protocol requires larger number of messages than that of the invalidation protocol, it will be advantageous in scientific calculations which requires a large number of data multicast. The trace driven or execution driven simulation based on the practical application programs is required for the further comparison.

References

[1] L.N. Bhuyan, A.K. Nanda, and T. Askar. Performance and Reliability of the Multistage Bus Network. In *Proc. of ICPP*, pages I–26–I–33, 1994.

[2] G. Broomell and J.R. Heath. Classification Categories and Historical Development of Circuit Switching Topologies. In *ACM Computing Surveys, Vol.15, No.2, June*, Jun. 1983.

[3] D. Chaiken and A. Agarwal. Software-Extended Coherent Shared Memory: Performance and Cost. In *Proc. The 21st ISCA*, pages 314–324, 1994.

[4] H. Cheong and A.V. Veidenbaum. A Cache Coherence Scheme with Fast-Selective Invalidation. In *Proc. of 15th ISCA*, pages 299–307, 1988.

[5] Amano H., Zhou L., and Gaye K. SSS(Simple Serial Synchronized)-MIN: a novel multistage interconnection architecture for multiprocessors. In *Proc. of the IFIP 12th World Computer Congress, Vol.I*, pages 571–577, Sept. 1992.

[6] K. Hiraki, H. Amano, M. Kuga, T. Sueyoshi, T. Kudoh, H. Nakashima, H. Nakajo, H. Matsuda, T. Matsumoto, and S. Mori. Overview of the JUMP-1, an MPP Prototype for General-Purpose Parallel Computations. In *Proc. IEEE International Symposium on Parallel Architectures, Algorithms and Networks*, pages 427–434, 1994.

[7] H.Matsumoto and T.Hiraki. The shared memory architecture on the massively parallel processor. In *Technical report of IEICE, CPSY 92-36*, pages 47–55, 1992.

[8] D.V. James, A. T. Laundrie, S. Gjessing, and G. S. Sohi. Distributed-Directory Scheme: Scalable Coherent Interface. *IEEE Computer*, 23(6):74–77, 1990.

[9] T. Kudoh, H. Amano, T. Matsumoto, K. Hiraki, Y. Yang, K. Nishimura, K. Yoshimura, and Y. Fukushima. Hierarchical bit-map directory schemes on the RDT interconnection network for a massively parallel processor JUMP-1. In *Proc. International Conference on Parallel Processing*, 1995.

[10] J. Kuskin and al. et. The Stanford FLASH Multiprocessor. In *Proc. The 21st ISCA*, pages 302–313, 1994.

[11] D. Lenoski et al. The Stanford DASH Multiprocessor. *IEEE Computer*, 25(3):63–79, 1992.

[12] H.E. Mizrahi, J.L. Baer, E.D. Lazowska, and Zahorjan J. Introducing Memory into the Switch Elements of Multiprocesssor Interconnection Networks. In *Proc. of 16th ISCA*, pages 158–166, 1989.

[13] A.K. Nanda and L.N. Bhuyan. Design and Analysis of Cache Coherent Multistage Interconnection Networks. *IEEE Trans. on Computers*, 42(4):458–470, 1993.

[14] A.V. Veidenbaum. A Compiler-Assisted Cache Coherence Solution for Multiprocessors. In *Proc. ICPP*, pages 1026–1036, 1986.