Effects of Communication Characteristics on Task Mapping Quality on a 2-D Mesh with Wormhole Routing

S.-Y. Lee
Department of Electrical and Computer Engineering
Auburn University
Auburn, AL 36849
sylee@eng.auburn.edu

ABSTRACT

Given that computational load is well balanced, task mapping is mainly concerned with reducing communication overhead. How much communication time can be reduced by optimizing allocation of tasks on a multiprocessor system would be dependent on several factors. In this study, dependency of improvement by task mapping on communication pattern is investigated on a mesh with wormhole routing. Communication pattern may be described by a set of parameters such as the total number of messages, the number of sources, the number of destinations, and distribution of message size. Through extensive simulation, it has been shown that the communication pattern has a significant effect on reduction in communication overhead that can be achieved by task mapping. Effects of individual communication parameters have been analyzed.

1. INTRODUCTION

As more parallel computing systems, tightly-coupled multiprocessors or networks of workstations, are made commercially available, how those systems can be efficiently used for various applications has become an important issue. A specific issue of *task mapping* concerns with assigning partitioned tasks onto processing elements (PE's). Task mapping has been extensively studied by many researchers for a long time [1][2][3][4][5]. However, as a whole, it is still an open problem. The term, task mapping, is to be distinguished from *task scheduling* which determines the order in which a given set of tasks are to be executed on a single or multiple PE's. Task mapping (as used in this paper) allocates tasks, which are communicating concurrently and are to be executed at the same time, onto multiple PE's. With task partitioning fixed, task mapping mainly affects communication overhead. As the ratio of communication time to computation time in parallel or distributed computing increases, it is required that a mapping scheme take communication overhead into account. This increased ratio is due to a larger number of PE's now available for an application, a larger size of shared data involved in many recently-emerging applications, the ever-increasing processor speed, etc.

One may attempt to reduce communication overhead by optimizing "when PE's send out messages", i.e., scheduling communication with a given (spatial) mapping. However, the achievable improvement can be limited since a spatial assignment (mapping) of communicating tasks is fixed. That is, mapping would help scheduling to achieve a better result (a smaller communication overhead). Also, this approach would require more information to deal with such as timing information. Furthermore, it can be more sensitive to dynamic variation of communication.

Communication pattern among PE's in a multiprocessor system depends on where communicating tasks are assigned and shared data are allocated. Depending on the relative positions of communicating tasks, the resulting communication overhead may vary significantly. In some cases, *random* mapping may work just as well as any elaborated mapping. Therefore, whether one should concern about mapping at all and what should be optimized in mapping are to be addressed.

There have been significant research efforts in optimizing communication in order to minimize communication overhead including [7][8]. Various communication patterns such as multicasting, broadcasting, all-to-all broadcasting, etc. were investigated. Hambrusch et. al. [9] considered "S-to-P" broadcasting problem on meshes, where S is the number of source PE's and P is the total number of PE's in a mesh, $1 \leq S \leq P$. A set of particular source distribution patterns were examined.

In this paper, how much communication overhead can be reduced by optimizing locations of communicating tasks (task mapping) on a 2-D mesh with wormhole routing depending on the characteristics of communication pattern is analyzed for general (or random) communication patterns. It is well known that communication time for a pair of PE's is *almost* independent of the distance between them in a wormhole routing network while it is proportional to the distance in a packet switching or message switching network [10][11]. Nevertheless, it should not be "extrapolated" to: communication overhead in a wormhole routing multiprocessor system does not depend on what PE's communicating tasks are mapped onto. Conflict among messages is dependent on locations of communicating tasks while communication

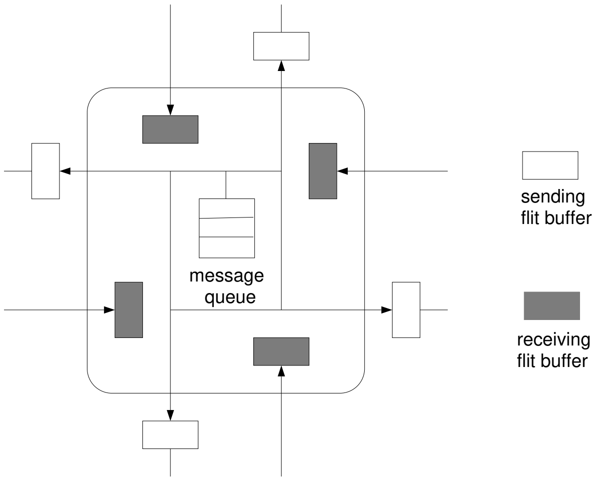


Figure 1: A processing element (PE) in the system model. A flit bypasses the receiving buffer to one of the sending buffers unless the current PE is its destination.

time for an isolated message can be said to be distance-independent. Note that such conflict directly affects communication time in general. This study shows that communication overhead on a wormhole routing multiprocessor system is not "independent" of locations of communicating tasks, and attempts to analyze how it is affected by *communication pattern* which in turn depends on locations of tasks.

The communication characteristics to be considered are the number and distribution of sources, the number and distribution of destinations, the total number of messages, distribution of message size, etc., which are referred to as *communication parameters*. An extensive simulation has been performed to analyze effects of the communication parameters on the quality of a mapping result that can be achieved. Two different objective functions are considered in the optimization procedure of task mapping and are compared in their effects on mapping results.

2. MODELS

2.1 System Model

In this study, a square mesh of size $N \times N$ is employed as a system model onto which a problem graph is mapped. Each PE (except those along the boundary) in a mesh is connected to its four neighboring PE's through four full-duplex channels as shown in Figure 1. That is, over each channel between two adjacent PE's, simultaneous communication in both directions is possible. For other types of channels such as single bidirectional or unidirectional channels, the general performance behaviors would be similar though they can be quantitatively different.

A data transfer from a source node to a destination node is carried out by *wormhole routing*. Each channel between two adjacent PE's has *flit* buffers at both ends (one each). Given a pair of source and destination, the path to be followed is specified by the *deterministic* X-Y routing.

Each PE has a message queue where messages to be sent to other PE's are temporarily stored until they are transmitted to their destinations. The message at the top of the queue is sent first. Once the tail flit of the message being transferred leaves the queue, transmission of next message in the queue may be initiated. Therefore, multiple messages from a PE may be travelling through the network at the same time (while only one message can be transmitted from a PE at any given time).

When a message which has left its source node is blocked by another before its header flit arrives at its destination, it *stays* in the network occupying the flit buffers along the path from the source node to the current node (at which it is blocked) until the blocking flit buffer is released. In the case that multiple messages contend for a flit buffer, the one destined for the farthest node from the current node is given the right to proceed first.

Each PE has multiple, two to four, incoming channels and can receive multiple messages arriving at the same time. It is assumed that virtual channel is not implemented. Therefore, channel contention is equivalent to flit buffer contention.

In this paper, the terms *PE* and *node* are interchangeably used. Whenever necessary, the terms, *system* node and *problem* node are used for clarity.

2.2 Communication Model

A problem (computation) which has been partitioned may be described by a graph to be referred to as a *problem* graph. In the problem graph, a node represents a task and an edge between two nodes represents communication between the corresponding tasks. In this study, it is assumed that tasks are to communicate with each other concurrently.

When communication patterns are *regular*, in most cases, efficient (optimal) algorithms to perform communication between nodes have been developed for various interconnection topologies. Also, the *best* task mapping is known or task mapping is unnecessary in many cases. On the other hand, when the communication pattern is irregular, an elaborate task mapping scheme has been employed in order to reduce communication time.

In order to obtain "general" results, a statistical approach is taken rather than considering a few particular problem graphs. That is, communication pattern (problem-graph) is "random", where certain characteristics, i.e., communication parameters defined below, are controlled.

Population, P, indicates communication intensity in terms of the total number of messages to be exchanged between nodes during a communication period. Each PE that has any messages to send to other PE's places them in the message queue (refer to Figure 1). All PE's with any messages in the queue attempt to transmit messages concurrently. The communication period lasts until delivery of all messages in the queue at all PE's is completed.

Source spread, S, is defined to be the number of sources, i.e., sending nodes. This parameter quantifies the spatial density of sending nodes. Note that $1 \leq S \leq N^2$ since a $N \times N$ mesh is considered in this study. $P \geq N^2$ doesn't necessarily mean $S = N^2$ because a PE may have more than one message to send. In order to have more control over communication pattern, an auxiliary parameter is used, which is the maximum number of messages, M_s, that can be sent by a PE.

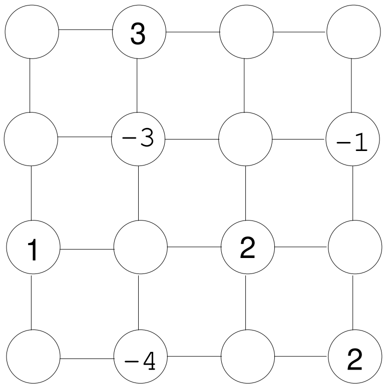


Figure 2: An example for communication parameters where P = 8, S = 4 (M_s=3), and D = 3 (M_d=4). A positive number indicates the number of messages to be sent while a negative number does the number of messages to be received.

Destination spread, D, is defined to be the number of destinations, i.e., receiving nodes where $1 \leq D \leq N^2$. This parameter indicates the spatial density of receiving nodes. As in the source spread, $P \geq N^2$ doesn't always imply $D = N^2$. The maximum number of messages that can be destined to (received by) a PE is denoted by M_r.

Uniformity specifies the distribution of message size, in terms of its mean, U_m, and standard deviation, U_d. For example, $U_d = 0$ means that all messages have the same size.

Obviously, communication overhead depends on the population of messages, e.g., more messages usually results in a higher overhead. How much the overhead can be reduced by mapping (i.e., optimizing locations of communicating tasks) must vary with population. A larger S given P means that messages to be sent are distributed over a larger number of PE's and therefore a larger number of messages exist being routed simultaneously while a smaller S indicates that fewer PE's are to send a larger number of messages. A similar interpretation can be given to D. Therefore, it is expected that, for the same population, source and destination spreads would affect the room for optimization of locations of communicating tasks, i.e., effectiveness of task mapping. Also, the size of message is a factor to be considered especially in the case of wormhole routing, which determines how long the flit buffers at the intermediate PE's (including source and destination PE's) are occupied by a message. Though this set of parameters may not be inclusive, this should be sufficient to characterize a communication pattern for this study.

3. TASK MAPPING

In task mapping, one determines the correspondence between a set of tasks (*problem graph*) and a group of PE's (*system graph*) such that a certain objective is achieved. In defining what to be achieved in task mapping, one may employ a cost or objective function. Depending on the objective function, the final mapping can be significantly different even for the same problem and system graphs. Also, given an objective function, the quality of mapping that can be achieved would vary with optimization method employed.

The objective of this study is not to develop a new mapping scheme. One of the issues to be addressed is whether a mapping, optimization in particular, is needed or not given an application (problem graph) and a multiprocessor system. If it is needed or not would depend on how much we want to improve assignment of tasks (and how much room we have for optimization) over random mapping. Also, another related issue is what (objective function) is to be optimized in mapping.

In the case of wormhole routing, as long as the size of a message is sufficiently large, the time required to send it from a node to another is *almost* distance-independent and is mainly proportional to the message size. However, this is true only when there is no conflict among messages (network contention) such that a message is not delayed by others. When there are multiple messages being transferred in a network, a message may experience *blocking*. Then, the actual delivery time of a message may be significantly longer than the "ideal" one which is mainly dependent on the message size only (independent of distance). The purpose of this study is to analyze the effect of the communication parameters on the *total communication time*, T, that can be achieved by mapping and also to examine what type of objective function is to be employed in optimizing mapping.

A more accurate objective function (quantifying more realistically what is to be optimized) would require more parameters to be considered and a more complex formulation and therefore usually takes a longer time to evaluate. That is, a proper objective function should be selected considering the acceptable accuracy and complexity.

3.1 Two Objective Functions

There are two common components found in most objective functions used for task mapping: distance between communicating nodes (PE's) and edge conflict (congestion). Based on these two, the following two simple objective functions are employed in this study. One, to be referred to as OF_d, is the maximum of weighted distances between all communicating nodes. The weighted distance is the product of the size of a message and the number of hops the message is to travel. The other, to be referred to as OF_e, is the maximum of edge congestions for all system edges (channels). Edge congestion is defined as the sum of sizes of messages to be transferred over a system edge. These objective functions are general enough for wide application and can be easily evaluated for fast mapping.

Let G_p and G_s be a problem graph and a system graph, respectively. $G_p = \{V_p, E_p\}$ where V_p is the problem node set of which elements are subtasks, v_{pi}, i.e., $V_p = \{v_{pi}\}$ for $1 \leq i \leq N$ when there are N subtasks, and E_p is the problem edge set of which elements correspond to communication between subtasks. Specifically, e_{pij} is the edge between v_{pi} and v_{pj} and $|e_{pij}|$ denotes the communication intensity (i.e., message size) between v_{pi} and v_{pj} that communicate with each other, for $1 \leq i, j \leq N$ ($i \neq j$). Similarly, $G_s = \{V_s, E_s\}$. $V_s = \{v_{sk}\}$ where v_{sk} is a system node corresponding to a PE_k, and $E_s = \{e_{pkl}\}$ where e_{pkl} denotes a channel or link between v_{sk} and v_{sl} that are adjacent to each other.

Suppose that v_{pi} and v_{pj} are mapped onto $v_{sk(i)}$ and $v_{sl(j)}$, respectively, where the notation $x(y)$ means that v_{py} is mapped onto v_{sx}. Let's denote the weighted distance between $v_{sk(i)}$ and $v_{sl(j)}$ by $d_{k(i)l(j)}$. Then, the objective function, OF_d, can be defined as follows.

$$OF_d = \max_{i,j} d_{k(i)l(j)} \quad (1)$$

Now, let $R_{k(i)l(j)}$ be the set of system edges along the routing path from $v_{sk(i)}$ to $v_{sl(j)}$. Define $|e_{smn}|$ to be $\sum_{i,j} |e_{pij}|$ for $e_{smn} \in R_{k(i)l(j)}$ and all i, j. Note that if $|e_{pij}| = 1$ for all i, j, $|e_{smn}|$ denotes the number of messages which share the link e_{smn}. Then, OF_e can be expressed as follows.

$$OF_e = \max_{m,n} |e_{smn}| \quad (2)$$

OF_d represents a class of objective functions by which distances between communicating tasks are minimized. On the other hand, OF_e is a typical formulation of objective function in which messages are to be spread over as many different channels as possible (independent of distances between tasks). Note that OF_d would be a reasonable choice for a system where the communication overhead is mainly distance-dependent while OF_e is for a system where availability of resources such as channels and buffers determines communication time (resource-dependent). However, in practice, unless there is no or very low contention, both types of dependency exist.

3.2 Effects of Task Mapping

Objective Function, OF_d

One effect of task mapping with OF_d is the decreased distance between communicating nodes and therefore a shorter propagation delay. Also, it (a smaller OF_d) increases the probability that a message is not blocked by other messages since it travels a shorter distance. That is, the network latency is usually reduced. In the case of wormhole routing, the total communication time heavily depends on the "degree" of contention (blocking) in addition to the size of message.

Objective Function, OF_e

Minimizing OF_e is qualitatively equivalent to minimizing the number of communicating nodes (source and destination) in each row and column of a mesh. That is, it tends to spread the communicating nodes uniformly over rows and columns. Decreasing the number of communicating nodes in a row would reduce the possibility of contention during routing in the X dimension. Similarly, a smaller number of communicating nodes per column lead to less contention during the Y dimension routing.

3.3 Optimization

An essential procedure in task mapping is optimization. Several different approaches were employed in the past, from a simple heuristic method to a time-consuming global optimization technique. The emphasis of this study is not on mapping scheme itself. Therefore, a simple optimization technique is employed to emphasize practicality for both of the objective functions.

The optimization strategy employed in this study is the pairwise exchange. Starting from the random (initial) assignment, optimization is carried out through iterations. In each iteration, a pair of PE's (system nodes) on which problem nodes are to be swapped temporarily are randomly selected. If the swapping doesn't increase the (value of) objective function, it is granted. Otherwise, no change is made to the current mapping. This iterative optimization process continues until the number of consecutive iterations (n_{con}) which would increase the objective function exceeds a certain limit (N_{limit}) or the total number of iterations (n_{total}) reaches an allowed maximum (N_{max}).

4. SIMULATION RESULTS AND DISCUSSION

4.1 Simulation

An extensive computer simulation has been carried out to analyze effects of communication patterns on task mapping. Random problem graphs are used for this simulation. Generation of problem graphs are controlled by a given set of communication parameters. For each problem graph, the three mappings are evaluated in terms of T on a mesh with wormhole routing. The first is a *random* mapping, i.e., no task mapping effort is made. The other two are obtained by task mapping with the two objective functions described in Equations (1) and (2). Then, improvement (reduction in time) achieved by each of the two task mappings over the random assignment is considered.

Communication Parameters

The size of mesh employed is 8 × 8, non-wrapped-around. The population, P, varies from 10 up to 120. For each P, both of $\frac{S}{P}$ and $\frac{D}{P}$ are between $\frac{1}{8}$ and 1. The size of a message, U_m, is changed from 5 up to 30 flits. Note that the maximum distance between two PE's in a 8 × 8 mesh is 15 and therefore the maximum message size of 30 flits should be long enough to examine cases where messages occupy whole paths from sources to destinations during transmission. Also, the standard deviation of the message size is varied from 0 to 5 for $U_m = 20$.

Simulation Results

Reduction factor, which is defined below, is used for performance analysis.

$$R_d = \frac{(T_r - T_d) \times 100}{T_r}$$

where R_d and T_d are the reduction factor and the communication time achieved by mapping with OF_d, respectively, and T_r is the communication time achieved by the random mapping.

The reduction factor indicates how much reduction in communication time one can expect by task mapping on a mesh with wormhole routing. Note that R_d, expressed in percentage, approaches to 100 as T_d to 0. The reduction factor for mapping with OF_e is defined similarly. That is, $R_e = \frac{(T_r - T_e) \times 100}{T_r}$ where T_e is the communication time achieved by mapping with OF_e.

In Figures 3 and 4, the reduction factor is observed as the population, P, varies for different source (S) and destination (D) spreads. In Figures 5, 6 and 7, the effects of S and

D on the reduction factor achieved are mainly examined. In Figures 8 and 9, the message size is varied for different populations to analyze the message size dependency of reduction factor. In Figure 10, variation of message size is considered. In each case of the simulation, multiple problem graphs with the same communication parameters are used and the *average* reduction factor is reported.

4.2 Discussions

As can be seen in Figures 3, 4, 5, 6 and 7, it is clear that task mapping can have a significant effect on the communication time for concurrently communicating tasks on meshes with wormhole routing and the degree of effect varies with the communication parameters.

Population

For a relatively light traffic (a small P), one can expect a significant reduction in communication time by a simple task mapping scheme as shown in Figures 3 and 4. As P increases, however, the reduction factor tends to decrease.

When only a few messages are in a system, there would not be much interaction among them. Therefore, it is easy to improve both of OF_d and OF_e by a large margin. Also, time required for a message would be more dependent on distance than on edge (buffer) conflict. This is why the reduction factor is larger for OF_d than for OF_e when P is small as can be seen in Figures 3 and 4.

As P increases, messages interact with each other more. Therefore, improving (reducing the value of) either objective function for a part of problem graph is highly likely to have a negative effect on mapping of other parts. Hence, a smaller reduction in the communication time is expected. Since more buffer contention which increases network latency in a wormhole routing system occurs, mapping with OF_e has a better chance to reduce the communication time more (refer to Figures 3 and 4). Note that minimizing OF_e reduces buffer conflict more explicitly.

Source/Destination Spread

For the same population, the reduction factor depends on distribution of messages over nodes. In Figures 5, 6, and 7, it can be seen that a smaller source spread (S) results in a smaller reduction in communication time. This observation may be explained as follows. A smaller spread means that messages are concentrated on fewer nodes (PE's), i.e, the number of messages per node is larger. Consider the following two cases. In one case, a node (say node A) needs to send a message to multiple (other) nodes. Therefore, the maximum of distances (for example) between node A and other nodes is to be minimized. In the other case, there are independent multiple pairs of nodes, and nodes in each pair need to communicate each other. Then, the distance between two nodes in each individual pair should be minimized. In general, minimization in the former case is harder. Hence, when messages are not spread over many nodes, communication time is reduced less by mapping.

It is also noticed that the reduction factor, R, is less sensitive to D than to S. Remember that it is assumed that each PE can transmit only one message at any given time while it may receive multiple messages arriving at different buffers simultaneously. That is, receiving capacity is higher than sending capacity. Therefore, variation in the destination spread is less influential to the reduction factor than that in the source spread.

OF_d and OF_e

It can be observed that mapping with OF_e performs better than with OF_d as both of S and D decrease with a fixed P. When both of the source and destination spreads decrease, (the same number of) messages are exchanged between a smaller number of nodes. This leads to the increased number of messages from or to a node. Therefore, in order to minimize network latency, it would be more effective to reduce channel (buffer) conflict rather than distance between source and destination. When the number of messages per node is large, the increase in network latency due to a path conflict would be larger. Therefore, in such cases, OF_e is the one to be employed in mapping.

Message Size

In Figures 8 and 9, message size dependency is analyzed. As the message size increases, the reduction factor one can expect to achieve decreases. A longer message occupies flit buffers along its path for a longer time. Therefore, the probability of reducing delay experienced by a message by a certain amount would be lower for longer messages. This is true especially when there is a larger number of such messages in a system as can be seen by comparing Figures 8 and 9. It is also confirmed in these figures that the larger the source spread is the larger the reduction factor becomes.

Variation of Message Size

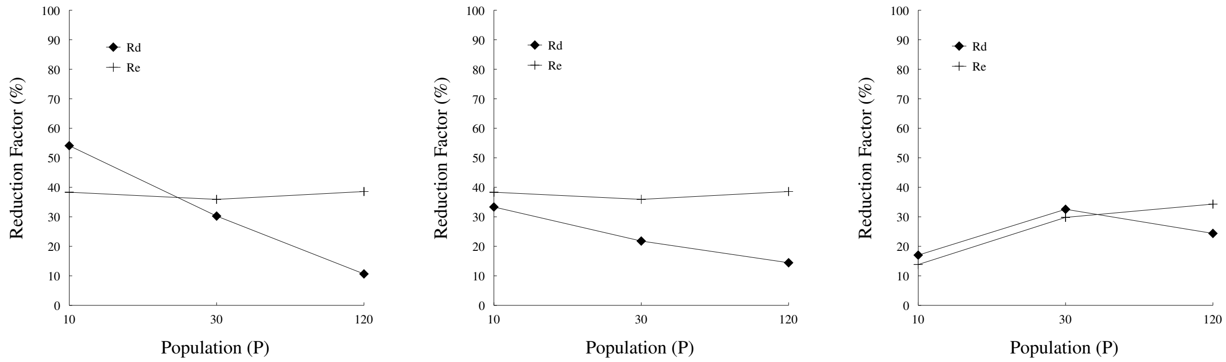
As shown in Figure 10, variation (standard deviation, U_d) in the message size has a very marginal effect on reduction factor in most cases. The reason for this observation is that the overall behavior is determined by the average message size rather than its variance which may change *instantaneous* behavior. However, a slight decrease in reduction factor can be observed as U_d increases.

5. SUMMARY

The simulation results may be summarized as follows:

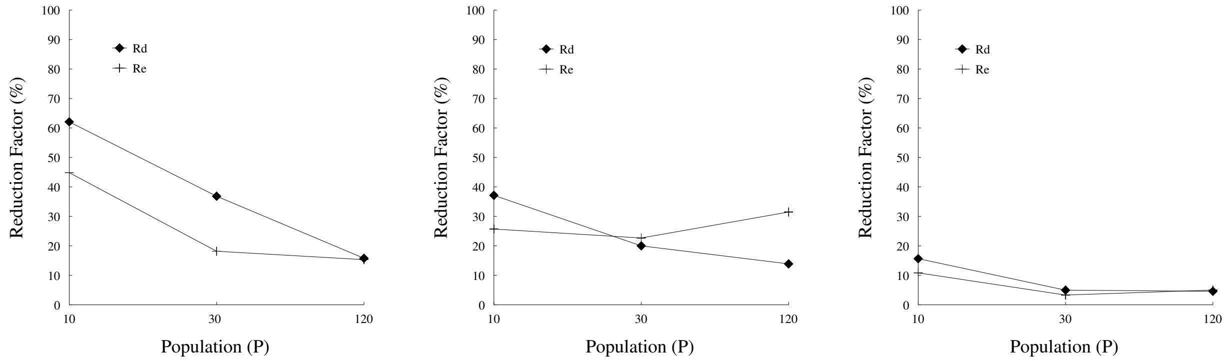
- In a wormhole routing multiprocessor system, mapping (spatial assignment of communicating tasks) has a significant effect on communication overhead.
- Unless population is too high, a significant reduction in communication time is achievable by a simple task mapping.
- A higher percentage of reduction is obtained for a smaller population.
- A smaller spread tends to result in a less reduction.
- Minimizing channel (buffer) conflict is more effective than minimizing distance when population is relatively high or when the spread is smaller.

It is observed that the simulation results reported in this paper do not always match with the above summary. This observation is believed to be largely due to the local search algorithm employed in optimization step of mapping. While the algorithm is simple to implement and fast to execute, it



(a) (b) (c)

Figure 3: Reduction factor as a function of population when $N = 8$, $D = 16$, $U_m = 10$ and $U_d = 0$. (a) $S = 16$ (b) $S = 8$ (c) $S = 2$.



(a) (b) (c)

Figure 4: Reduction factor as a function of population when $N = 8$, $D = 2$, $U_m = 10$ and $U_d = 0$. (a) $S = 16$ (b) $S = 8$ (c) $S = 2$.

must have found local (rather than global) optimum solutions (mapping) in some cases. Also, the unavoidable imperfectness of simulation is a minor factor contributing to this deviation.

6. REFERENCES

[1] H. S. Stone. "Multiprocessor scheduling with the aid of network flow algorithms", *IEEE Transactions on Software Engineering*, SE-3(1):85–93, January 1977.

[2] S. H. Bokhari, "On the Mapping Problem", *IEEE Transactions on Computers*, C-30(3), pp207-214, March 1981.

[3] S. W. Bollinger and S. F. Midkiff. "Heuristic technique for processor and link assignment in multicomputers", *IEEE Transactions on Computers*, 40(3):325–333, March 1991.

[4] G. Cybenko. "Dynamic load balancing for distributed memory multiprocessors", *Journal of Parallel and Distributed Computing*, 7:279–301, 1989.

[5] S.-Y.Lee and J.K.Aggarwal, "A Mapping Strategy for Parallel Processing", *IEEE Transactions on Computers,* C-36, no. 4, pp433-442, April 1987.

[6] J.C. Jacob and S.-Y. Lee, "Task Spreading and Shrinking on a Network of Workstations with Various Edge Classes", Proceedings of IEEE International Conference on Parallel Processing, vol III, pp174-181, August 1996.

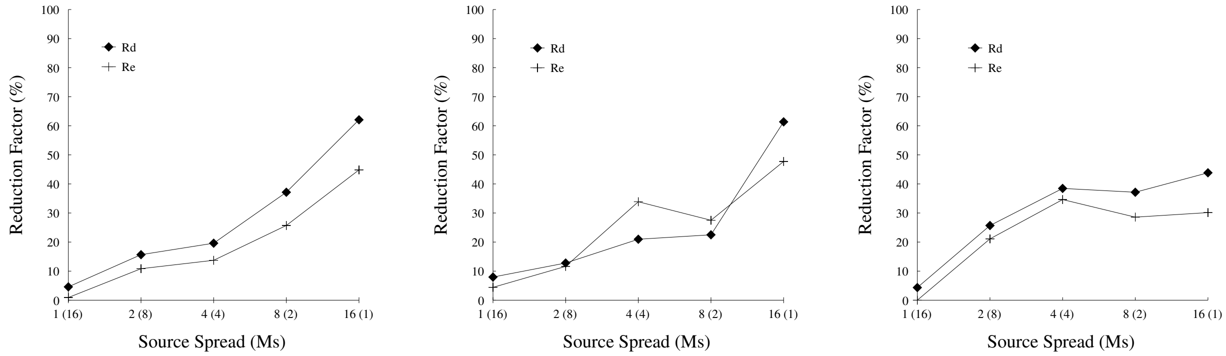
[7] S. E. Hambrusch, F. Hameed, and A. A. Khokhar. "Communication Operations on Coarse-Grained Mesh Architectures", *Parallel Computing*, vol. 21, pages 731-751, 1995.

[8] S. Ranka, R. Shankar and K. Alsabti. "Many-to-Many Communication with Bounded Traffic", *Proc. of Symp. on the Frontiers of Massively Parallel Computation*, 1995.

[9] S. E. Hambrusch, A. A. Khokhar and Y. Liu. "Scalable S-to-P Broadcasting on Message-Passing MPPs", *Proc. of International Conference on Parallel Processing*, vol. I, pages 69-76, 1996.

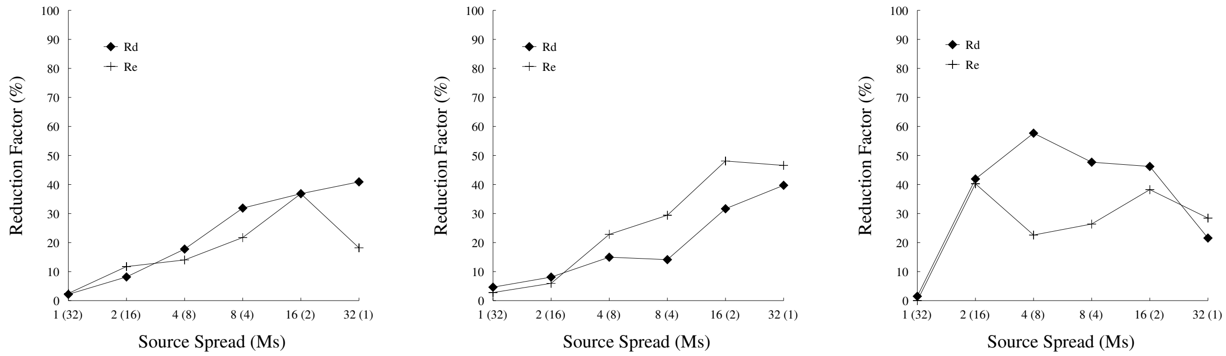
[10] W. Dally and C. Seitz, "The Torus Routing Chip", *J. Distributed Computing,* vol.1, No.3, pp187-196, 1986.

[11] L. Ni and P. McKinley, "A Survey of Wormhole Routing Techniques in Direct Networks", *IEEE Computer*, pp62-76, vol.26, No.2, February 1993.



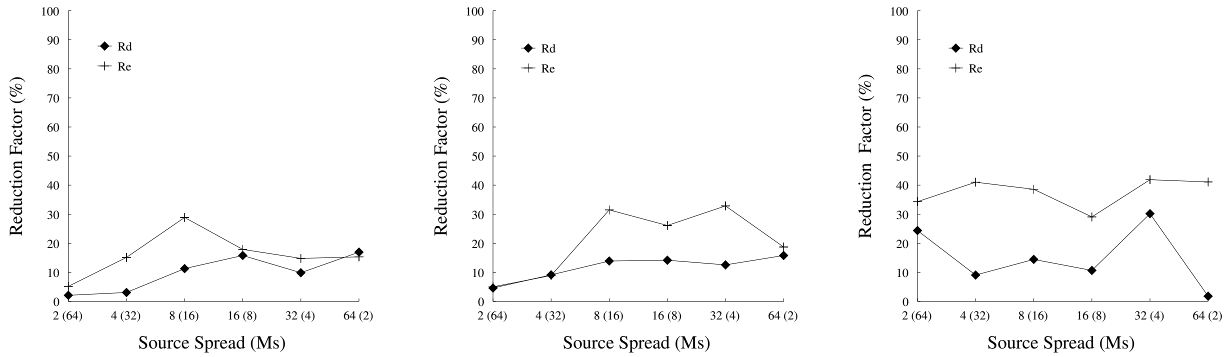
(a) (b) (c)

Figure 5: Reduction factor as a function of source spread when $N = 8$, $P = 10$, $U_m = 10$ and $U_d = 0$. The number in parentheses is M_s. (a) $D = 16$, $M_r = 1$ (b) $D = 4$, $M_r = 4$ (c) $D = 1$, $M_r = 16$.



(a) (b) (c)

Figure 6: Reduction factor as a function of source spread when $N = 8$, $P = 30$, $U_m = 10$ and $U_d = 0$. The number in parentheses is M_s. (a) $D = 32$, $M_r = 1$ (b) $D = 8$, $M_r = 4$ (c) $D = 1$, $M_r = 32$.



(a) (b) (c)

Figure 7: Reduction factor as a function of source spread when $N = 8$, $P = 120$, $U_m = 10$ and $U_d = 0$. The number in parentheses is M_s. (a) $D = 64$, $M_r = 2$ (b) $D = 16$, $M_r = 8$ (c) $D = 2$, $M_r = 64$.

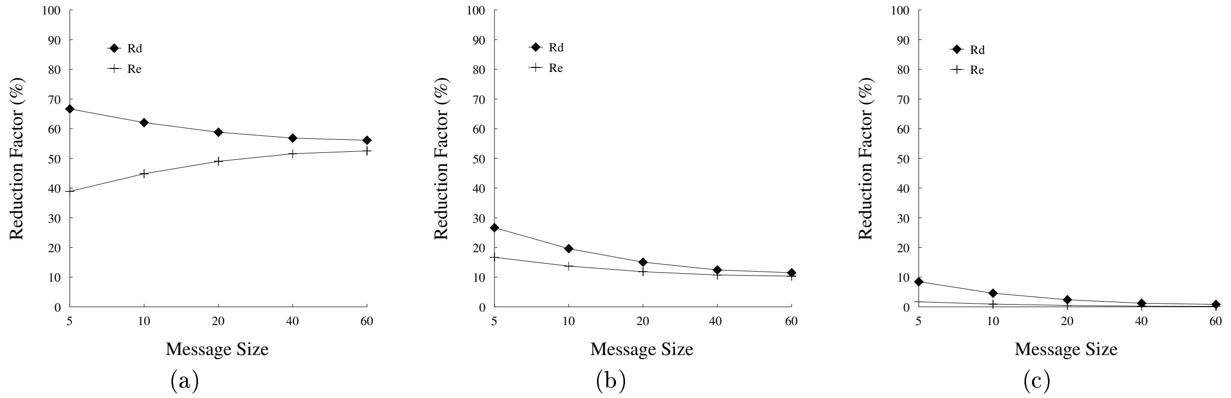


Figure 8: Reduction factor as a function of message size when N = 8, P = 10, U_d = 0, D = 16 and M_r = 1. (a) S = 16, M_s = 1 (b) S = 4, M_s = 4 (c) S = 1, M_s = 16.

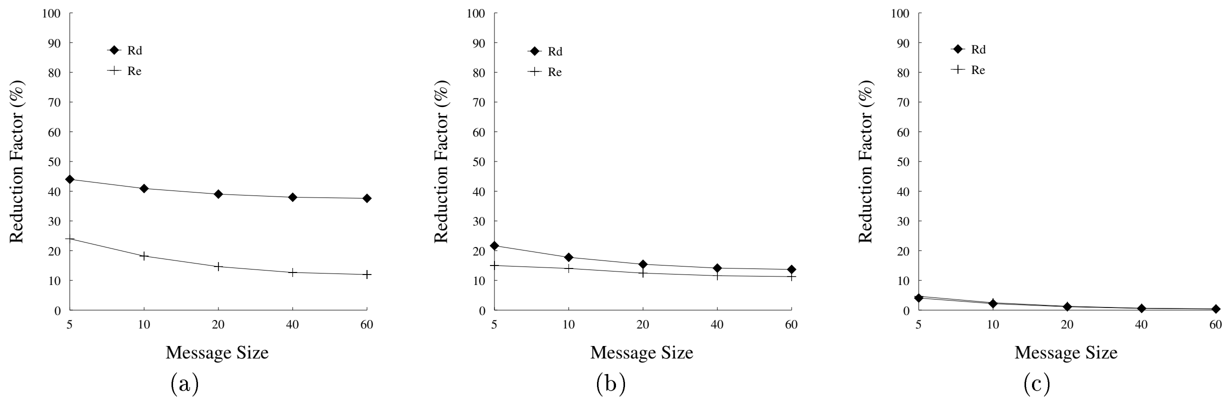


Figure 9: Reduction factor as a function of message size when N = 8, P = 30, U_d = 0, D = 32 and M_r = 1. (a) S = 32, M_s = 1 (b) S = 4, M_s = 8 (c) S = 1, M_s = 32.

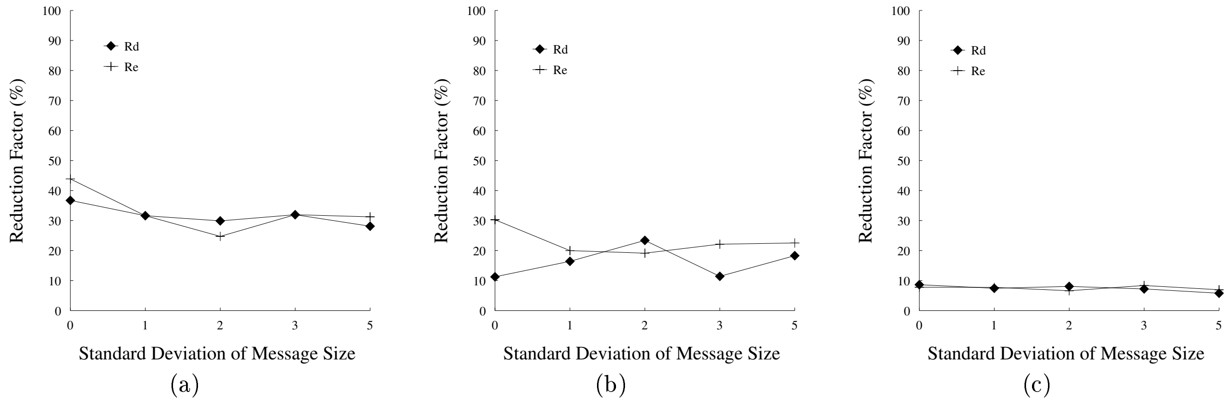


Figure 10: Reduction factor as a function of variation of message size when N = 8, P = 10, U_m = 20, D = 16 and M_r = 2. (a) S = 32, M_s = 1 (b) S = 4, M_s = 8 (c) S = 2, M_s = 16.