Predictive Switching in 2-D Torus Routers

Tsutomu YOSHINAGA†, Shojiro KAMAKURA†, Michihiro KOIBUCHI‡
†Graduate School of Information Systems, University of Electro-Communications, {yosinaga,kamakura}@sowa.is.uec.ac.jp
‡National Institute of Informatics, koibuchi@nii.ac.jp

Abstract

This paper proposes predictive switching in 2-D torus routers to reduce the number of pipeline stages for low-latency communication. By utilizing the communication regularity in parallel applications, a dynamic predicting mechanism presets packet traversal paths inside the router before packet arrivals. Hence, we can bypass the pipeline stages of routing computation, virtual channel allocation and switch allocation when the prediction hits. We considered the predictor architecture and accuracy for several traffic patterns in NAS parallel benchmarks. Our experiments show that a sampled pattern matching (SPM) predictor achieves 77% to 96% of the prediction hit rates when we use the dimension-order routing algorithm. We also discuss a method to improve the prediction accuracy of SPM by examining the frequency of occurrence for the prediction values in the communication history.

Keywords: predictive switching, routing speculation, router pipeline, dynamic prediction, communication regularity

1. Introduction

High-performance routers need to provide high-bandwidth and low-latency communication capabilities. Recent high-end parallel computers consist of tens of thousands of nodes; hence, the average message distance tends to be long. In such large interconnection networks, communication latency is critical because it incurs cumulative hop latency per router. Because fast clock frequency and the increase of routing complexity add to the depth of the router pipeline, it is advantageous to reduce the required number of pipeline stages for the forward packet header. Routing speculation is a technique to reduce the hop latency by performing several tasks simultaneously in a pipeline stage of the router[8]. Route lookahead, which performs the routing computation one hop ahead, can be applied to the speculative pipeline for further shortening[2]. Another technique to reduce the hop latency is mad postman switching, which forwards a message straight on the current dimension before its routing is completed[5]. This strategy works well with a dimension-order routing algorithm in 2-D mesh networks when the average transmission distance of the messages is long. It utilizes characteristics that a packet will make at most one turn from one dimension to another. However, these techniques do not utilize dynamic communication prediction.

The communication patterns in parallel algorithms exhibit some degree of regularity. Ding et al. proposed predictive multiplexed switching, which utilizes communication regularity [3]. It establishes an end-to-end connection in an indirect network before the connection is needed. However, massively parallel computers usually employ connectionless direct networks. To the best of our knowledge, the dynamic communication prediction per router in direct networks has not yet been studied in depth.

In this paper, we propose a technique of per-router basis predictive switching for 2-D tori. Our prediction technique is a dynamic one which utilizes the communication history in each port. The predictor suggests an output port for the next incoming message packet. Then, the associated virtual channel and switch allocation stages are executed before the actual routing computation. Hence, we can reduce the number of pipeline stages to hop a packet when the prediction is hit.

The remainder of this paper is organized as follows. In Section 2, the router architecture which supports dynamic prediction is explained. Section 3 presents the prediction methods. Section 4 discusses the communication regularity in NAS parallel benchmarks [1]. The prediction hit rates for these communication patterns are shown in Section 5. We conclude this paper in Section 6.

2. Router Architecture

Figure 1 shows a block diagram of a router which supports dynamic routing prediction. It is organized with a set of input and output ports which are connected by a crossbar switch, a predictor, and routing logic. For a 2-D torus network, the number of input and output ports is five, including a consumption port and an injection port to and from a pro-
cessing element (PE). Four input and output ports are connected with adjacent routers which are located north, east, west and south in the network. The crossbar switch is controlled by the routing logic and supplies the packet traversal paths from the input ports to the output ports.

Each port consists of multiple virtual channels (VCs) and a memory which stores the communication history. In figure 1, these memories are illustrated as I-Log and O-Log, respectively. In order to save the communication history, each input and output port is numbered from zero to four. Once a packet is transferred from an input port to an output port, the associated number of the output port is added at the end of the I-Log in the input port. At the same time, the input port number is stored at the tail of the O-Log in the output port.

Packet forwarding on this type of router is typically performed by four pipeline stages: routing computation (RC), VC allocation (VA), switch allocation (SA) and switch traversal (ST) [2]. Figure 2(a) shows a typical router pipeline. We assume that a send signal is propagated with flits (flow control units) to latch the data at the adjacent router side. This signal is enabled at the end of each ST stage. The four stages are listed below.

**Routing Computation (RC)**: Decodes the packet header and computes the output port candidate(s).

**VC Allocation (VA)**: Performs output port arbitration and allocates the output VC to the winning request.

**Switch Allocation (SA)**: Sets up the crossbar switch based on the VA status.

**Switch Traversal (ST)**: Transfers the packet flit by flit from the input port or injection port to the allocated output port or consumption port via the crossbar switch.

A communication predictor can be straightforwardly added to this kind of traditional pipelined router. The predictor suggests an output or an input port for the next incoming/outgoing packet. This prediction operation is performed between the VA stage of the previous packet and the RC stage after packet arrival. Therefore, unlike the branch prediction of advanced microprocessors, our routing prediction may consume multiple cycles for the duration of packet arrival. The predicted path will be pre-established inside the router between the input and output ports when the output port is available. When a conflict of multiple predictive requests from different I-Logs to the same output port occurs, the prediction result from the O-Log in the selected output port is used to resolve it. Figure 2(b) shows a pipeline for predictive switching. As we can see in the figure, the received flits are transmitted to the pre-allocated output port right after they are buffered in the input VC. We call this action predictive switch traversal (PST). Consequently, the three pipeline stages RC, VA, and SA can be reduced when the prediction hits.

A prediction miss may lead to a wrong path traversal at a PST stage. To accomplish the correct path traversal, the PST and RC stages for the header flit are executed in parallel. A proper output request is created at the RC stage and, if it is mismatched with the prediction, the send signal is not enabled to avoid wrong flit propagation. For cases of routers which take multiple cycles to transfer a header flit, such as on a bit-serial channel, *dead flits* may appear on the mispredicted path. The dead flits must be removed by utilizing a special control packet [5], or by detecting them at routers which do not perform the predictive switching. It is possible to discard the dead flits by properly arranging non-predictive switching routers in the network, such as at positions of the dateline in each dimension.

The misprediction recovery can be implemented with a...
single VC for virtual cut-through (VCT) switching by leaving the flits in the VC until the original ST stages. It is easy to retransmit a packet from the header flit to the correct path by simply restoring a register which points out the header flit. In a wormhole switching router, the recovery mechanism for the misprediction can be implemented using two VCs, one for the predictive switching and another one for the non-predictive switching. An incoming packet is duplicated and stored in both VCs. When the prediction is successful, only flits stored in the predictive VC are transmitted and flits in the non-predictive VC are discarded. In the case of misprediction, the predictive output is disabled and a packet stored in the non-predictive VC is processed by the original pipeline.

3. Communication Prediction

This section discusses the prediction algorithms which we consider in this paper. First, we explain how to apply a sampled pattern matching (SPM) algorithm to our communication predictor. Then, a technique is presented that focuses on the SPM prediction accuracy. A simple latest port matching predictor and a static straight predictor, which are repeated sequence of traffic. For example, the NAS parallel benchmarks (NPB) exhibit very high spatial locality [7]. Therefore, we decided to use the SPM algorithm, which was proposed in [6] as a universal predictor, for a starting point with NPB.

Let us assume that a sequence \(X_1, X_2, ..., X_n\) is a communication history, where \(X_1, X_2, ..., X_n\) denote output port numbers. The SPM algorithm predicts the next value \(X_{n+1}\) (that is, the output port number) by finding the value occurring the most frequently at the sampled positions. First, it searches the longest suffix sequence of \(X_n\), whose copy appears somewhere in \(X_n\). That is, \(X_1, ..., X_n = X_{i-j}, ..., X_{n-j}\) for some \(1 \leq j \leq n\), where the length of the repeated sequence is \(D_n = n - j + 1\). Then, it defines a marker that is a sequence of length \(k = \lceil \alpha D_n \rceil\), where \(0 < \alpha \leq 1\). Such a marker sequence \(X_{n-k+1}^n\) appears \(O(n^{1-\alpha})\) times in \(X_n^k\) with high probability. Finally, the prediction value is given by applying majority rule to all numbers appearing at positions just after the markers. Although the original SPM was considered for \(0 < \alpha < 1\), we use the initial \(\alpha = 1\), since longer matches showed higher hit rates in our preliminary evaluation. When more than two numbers appear at the sampled positions with the same frequency of occurrences, we recompute the algorithm by shortening the length of the marker sequence.

In the following example, the longest matching suffix is 0012 and the sampled numbers are 3, 2 and 2. Hence, the

![Figure 3. Two cases of the occurrence of port numbers at sampled positions, (a) concentration of port 2, (b) scattering of ports 1 and 2.](image)

3.2. Prediction Accuracy in SPM

The prediction accuracy depends on how much communication regularity is inherent in the traffic patterns. Imagine the two cases of the occurrence frequency of port numbers at sampled positions, as illustrated in Figure 3. Case (a) shows the explicit characteristic that port number 2 is mainly used at the sampled positions after the markers. On the other hand, in case (b), port numbers 1 and 2 competitively appear at the sampled positions. To clarify the relation between such a scattering of occurrences and prediction accuracy, we executed an experimental simulation for two routing algorithms, X-Y dimension-order routing (DOR) and Duato’s adaptive routing (Duato) [4], on a 32×32 torus network. Three communication patterns—uniform random, bit-reversal and matrix transpose—with the traffic rate 0.1 (flit/node/cycle) were used, and the maximum marker length was set to 1,024.

Table 1 shows the prediction hit rates vs. the number of ports appearing at the sampled positions more than the average number of times. The results clearly show the tendency for a smaller numbered of ports to attain higher hit rates. For environments where the misprediction penalty is relatively high, such that the dead flits consume network bandwidth, predictive switching should be restricted to highly probable cases. Based on this policy, we examine a modified version of the SPM algorithm, called SPM-1, which performs predictive switching only when a single port appears at the sampled positions more than the average number of times.
### 3.3. Static Straight (SS) Prediction

The static straight (SS) strategy predicts all incoming packets from the network will be forwarded straight on the same dimension, in the same manner as mad postman switching. For example, packets in the north input port are always predicted to be output from the south port in 2-D tori. In a dimension-order routing algorithm, it fails at most two times per packet in 2-D tori, where the packets turn from one dimension to another, and at the destination nodes. Therefore, packets which move a long distance increase the prediction hit rates, whereas the communication locality is a negative factor for this strategy. SS does not apply prediction for injection packets since all network ports would be allocated to the opposite network input ports.

However, the static predictor does not require a history memory, hence its implementation cost is low.

### 3.4. Latest Port Matching (LPM) Prediction

The latest port matching (LPM) method predicts that the next packet will use the same output port with the previous packet for each input port. This method requires only a single history record in each port so that the prediction can be performed in a short time.

### 4. Communication Regularity

This section discusses the communication regularity of several examples in the NAS parallel benchmarks. As Kim and Lilja analyzed in [7], many parallel scientific applications, including the NPB, exhibit message destination locality. That is, an individual node has only a small number of favored communication partners, although the overall distribution of the message destinations is uniform. Our concern in this paper is mainly the regularity of the order of output port usage for messages in each router. In order to clarify whether message destination locality is reflected in the packet traversal order, we analyzed the communication patterns of the following five programs: conjugate gradient method (CG), multi-grid solver (MG), lower-upper diagonal solver (LU), scalar pentadiagonal matrix solver (SP), and block tridiagonal matrix solver (BT). We executed each program of class W on a cluster of PCs with 64 nodes. The nodes are numbered from 0 to 63. Our analysis was made for ten thousand messages.

Table 2 shows the communication distances for these programs. The CG and MG programs include short- and long-distance messages, and their average distances are 2.53 and 2.66, respectively. When we look at the order of message destinations in CG, we found that there were two patterns of order. Nodes on the diagonal positions in the 2-D torus send messages to three nodes in such an order of A → B → C → C → B → A → A → B → C. For example, a diagonal node 36 sends messages to nodes 32, 37, and 38 in the order of 32 → 38 → 37 → 37 → 38 → 32 → 32 → 38 → 37. Another pattern for non-diagonal nodes is A → B → C → C → B → A → D → A → B → C. These patterns, which can be utilized for prediction of the injection port, are repeated on each node. Although multi-hop messages may disturb the order of usage in network ports, we can still expect some degree of regularity. We will consider this in the next section.

In the LU program, all the messages are sent to adjacent nodes. For example, node 36 sends messages to four nodes, including node numbers 28, 35, 37, and 44. The order of message destinations also has a pattern such that 37 → 35 → 44 → 28 is repeated two times, then 37 → 44 is repeated 31 times, and finally 35 → 28 is repeated 31 times. We can utilize this pattern for the prediction of the injection port, and the prediction of the other network ports is easy because all the input messages from adjacent nodes are received at the consumption port.

The SP and BT programs include 1- and 2-hop messages. When we focus on node 36 in the case of SP, pattern 37 → 35 → 44 → 28 → 43 → 29 is followed by seven messages to each destination of 37, 35, 44, 28, 43 and 29.

As we confirmed in several examples, many parallel applications involve regularity in the order of message destinations when we focus on individual nodes. We can expect to achieve high prediction hit rates of a message’s output direction for such programs with regular communication patterns.

### 5. Evaluation

This section considers the prediction hit rates of communications with the traffic patterns in the NAS parallel bench-
5.1. Experimental Conditions

We evaluated prediction accuracy by comparing the prediction values with the output port numbers of messages in a network simulation without performing the prediction. Communication data were obtained by executing NPB programs (CG, MG, LU, SP and BT) of class W on a real PC cluster with 64 nodes, although the packet size was fixed to 32 flits in our network simulation. We used 12,000 packets at an injection rate of 0.2 (flits/node/cycle) based on each communication pattern. An 8 × 8 2-D torus was used with two routing algorithms: X-Y dimension-order routing (DOR) and Duato’s adaptive routing (Duato). The number of VCs per input port were two for DOR, and three for Duato. Virtual cut-through flow control, which transfers flits in pipeline but stores a whole packet in a router when it is blocked, was used.

5.2. Prediction Hit Rates

First, we evaluate the prediction accuracy for the injection ports with three dynamic predictors, SPM, SPM-1 and LPM. Figure 4 shows the average hit rates of all 64 source routers for (a) DOR and (b) Duato. The SPM and SPM-1 predictors achieve high hit rates of over 90% in DOR for all programs since the communication locality and regularity are directly affected by the repeat sequences of message destinations. On the other hand, the LPM predictor shows poor hit rates except for SP and BT, in which messages are sent repeatedly to the same destinations. In the other three programs, message destinations often varied, hence LPM misses the prediction. For Duato’s adaptive routing, the hit rates for each predictor are slightly degraded compared to those of DOR, since the adaptive routing algorithm varies the communication regularity by finding an alternative available output port rather than using the same one in the previous history sequence.

Second, we discuss the average prediction hit rates of all input ports, including four network ports and one injection port. This time, we also evaluate the SS predictor with the three dynamic predictors, SPM, SPM-1 and LPM. Figure 5 shows the average hit rates of all 64 routers. We notice that the SPM and SPM-1 predictors attain very high hit rates of over 95% for LU in both cases of DOR and Duato. This is because its particular communication pattern is repeated without changing the order. It is easily understood that a communication pattern between two adjacent nodes is hardly varied for each input port, since there is no inter-
Table 3. Rates of predictable cases by SPM-1.

<table>
<thead>
<tr>
<th></th>
<th>NPB</th>
<th>DOR (%)</th>
<th>Duato (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CG</td>
<td>88</td>
<td>88</td>
<td></td>
</tr>
<tr>
<td>MG</td>
<td>87</td>
<td>86</td>
<td></td>
</tr>
<tr>
<td>LU</td>
<td>96</td>
<td>96</td>
<td></td>
</tr>
<tr>
<td>SP</td>
<td>93</td>
<td>86</td>
<td></td>
</tr>
<tr>
<td>BT</td>
<td>94</td>
<td>87</td>
<td></td>
</tr>
</tbody>
</table>

5.3. Analysis

This section discusses the analysis of the SPM and SPM-1 predictors. As we could see in the previous section, SPM works well for parallel applications which have communication regularity. The SPM-1 predictor improves its hit rates by focusing on cases with only a single output port that is highly probable to be selected after particular marker sequences in the communication history. The other scattering cases, in which two or more ports are selected after the markers for more than the average number of times, are not used for predictive switching. Therefore, we have a trade-off between the prediction accuracy and the number of predictable cases. Table 3 shows the rates of such predictable cases by SPM-1. We found that more than 86% of the messages can be the targets of predictive switching for our test programs, and the SPM-1 predictor effectively reduces the prediction misses of SPM. The rates of DOR are higher than those of Duato, and programs with higher prediction hit rates include cases more predictable by SPM-1.

Prediction hit rates in the SPM algorithm are also affected by the value of $\alpha$, which decides the length of markers against the longest suffix sequence in the communication history. Generally, the number of occurrences of longer markers in the history is decreased, especially for programs which have a variety of message distances. Table 4 shows the prediction hit rates of SPM for three values, $\alpha = 1$, 0.75, and 0.5. We notice that the best value of $\alpha$ varies depending on the program and routing algorithm, although the difference is not large. In the CG and MG benchmarks, smaller values of $\alpha$ give higher hit rates. This means that the number of samples should be increased to produce an accurate prediction in these two programs. On the other hand, the other three programs attain the highest hit rates when $\alpha = 1$. It could be considered that the number of occurrences of markers, which are the copies of the longest suffix, is enough for the prediction. We can confirm the relation between the prediction hit rates and length of used markers in Figure 6 through 8.

These figures plot the prediction hit rates for each port with the average length of the markers which are used in each port. Figure 6 shows the results for CG. We found that low prediction hit rates are concentrated at marker lengths of less than 15, and that 25 is enough to achieve reasonable hit rates. The plots in Duato are more scattered than those of DOR due to the short length of the markers, resulting in lower prediction hit rates. Although we don’t show the graphs for the MG benchmark, it is quite similar to Figure 6.

For the LU benchmark, which is shown in Figure 7, DOR and Duato show the same relation, since there is no routing adaptivity. This graph shows a concentration for marker lengths between 25 and 65. And the prediction hit rates are relatively stable at more than 95%. We also confirmed that a marker length of 25 is enough to achieve high prediction...
Table 4. Prediction hit rates (%) for the length of markers, $\alpha \times$ (length of the longest suffix).

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>CG DOR</th>
<th>Duato DOR</th>
<th>MG DOR</th>
<th>Duato DOR</th>
<th>LU DOR</th>
<th>Duato DOR</th>
<th>SP DOR</th>
<th>Duato DOR</th>
<th>BT DOR</th>
<th>Duato DOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>79.8</td>
<td>74.6</td>
<td>73.6</td>
<td>69.7</td>
<td>96.5</td>
<td>96.5</td>
<td>91.0</td>
<td>71.1</td>
<td>92.4</td>
<td>73.7</td>
</tr>
<tr>
<td>0.75</td>
<td>80.2</td>
<td>76.0</td>
<td>75.6</td>
<td>71.4</td>
<td>96.0</td>
<td>96.0</td>
<td>89.6</td>
<td>70.8</td>
<td>91.0</td>
<td>73.1</td>
</tr>
<tr>
<td>0.5</td>
<td>79.7</td>
<td>76.1</td>
<td>77.0</td>
<td>72.5</td>
<td>95.8</td>
<td>95.8</td>
<td>89.8</td>
<td>69.7</td>
<td>91.2</td>
<td>71.3</td>
</tr>
</tbody>
</table>

Figure 7. Prediction hit rates vs. average length of markers for LU. This graph is typical for DOR and Duato.

As illustrated in Figure 8, the results for BT with DOR show a tendency similar to LU, although the BT results are scattered between 5 and 30 for Duato. We notice again that the adaptive routing algorithm worked to reduce the length of the markers, and it incurred poor prediction hit rates for the short markers. The SP benchmark shows almost similar results to BT.

5.4. Large Networks

In mad postman switching, the SS predictor works well on a large-sized 2-D torus network with DOR. Our experiments on a $32 \times 32$ torus showed prediction hit rates of more than 80% regardless of uniform and non-uniform traffic. This result occurs because more than 80% of the total messages are transferred in a straight direction on the current dimension. This characteristic can also be utilized by the SPM-1 predictor. SPM-1 achieves prediction hit rates of approximately 90% for non-uniform communication, such as bit-reversal and matrix transpose traffic. However, SPM-1 showed prediction hit rates of 80% for uniform random traffic.

In non-uniform traffic, we found that the port utilization frequency for messages varies by the position of the ports, and the output direction of messages for particular ports is quite unique, such as the always straight direction. This kind of regularity is a major reason of the high hit rates of SPM-1 for non-uniform traffic. On the other hand, every port is evenly utilized in uniform random traffic with smaller communication regularity. Consequently, the prediction hit rates of SPM-1 for uniform random traffic are degraded approximately 10% compared to that of the non-uniform traffic, mainly in cases with markers of short length.

6. Conclusion

This paper studied the predictive switching technique in 2-D torus networks. Since many parallel applications exhibit communication regularity, it is highly possible to predict the output port direction for the next incoming message by utilizing a dynamic predictor. This technique reduces the required number of pipeline stages of routers to forward message headers; hence, a dynamic predictor can contribute to low-latency communication.
Our experiments utilizing several test programs from the NAS parallel benchmarks showed that the SPM-1 predictor achieves hit rates of more than 77% and 71% for dimension-order and Duato’s adaptive routing, respectively. SPM-1 is useful both for long and short distance communication, especially for dimension-order routing. We also clarified the relation between prediction hit rates and the lengths of markers in communication history. Namely, short markers miss the prediction accuracy. We believe that such dynamic information helps to improve the prediction accuracy. For our future work, we need to consider a practical design of the dynamic predictor.

Acknowledgments

We thank Dr. Kenji Kise of TIT and Dr. Kenichi Iwata of Yamagata University. They gave us valuable comments for the predictor design.

This research is supported in part by the Grants-in-Aid for Scientific Research of Japan Society for Promotion of Science (JSPS) No.17360178, and NII Joint research.

References