C-21 Speculative Routing in Multicomputer Direct Networks

Tran Cong So, Shigeru Oyanagi and Katsuhiro Yamazaki
Department of Computer Science, Ritsumeikan University

1. Introduction

This paper proposes a new selection scheme for adaptive routing algorithm that uses local network information to speculate the best output channel. Although previous selection functions make decision of output channel based on only local node status [2], our method uses additional exchanging information between nodes that is more accurate thus leading to improvement in throughput and latency.

2. Speculative Routing

2.1 Overview

Selection function selects the best output link based on the channel's profitability. Apparently, the channel profitability not only relies on status of current node but also depends on adjacent nodes statuses. Figure 1 is a simple example of channel profitability based on the next-node busy conditions. Supposed a message need to be routed from source node A to destination node I. At node A, the routing function shows that both the link AB and AD can be used without causing a deadlock to the network. Because the node B is a busy node, the probability that its output will be busy would be high. As a result, the link AB has lower profitability than the link AD, so that selection function should choose the link AD to forward message on. Similarly, at nodes D and E selection function should avoid busy nodes G and H. Therefore, the message travels on ADEFI path that would provide the least latency. On the contrary, in the oblivious straight selection function, supposed the link BC is busy, the message goes along the ABEHI path, which includes two busy nodes that clearly would cause more delay than the ADEFI of speculative case.

Channel's profitability is composed of several status variables. The most important status is the *{free, busy}* status set. If there is only one free channel, we should forward a message to it immediately to reduce the latency. The *next-node utilizations* (network conditions on that channel's direction) are useful incase of there are multiple free channels available to forward a message. If all channels are busy, the next-node utilizations help the message to decide what channel to wait. The number of messages that currently requests a specific channel also highly affects channel's profitability.

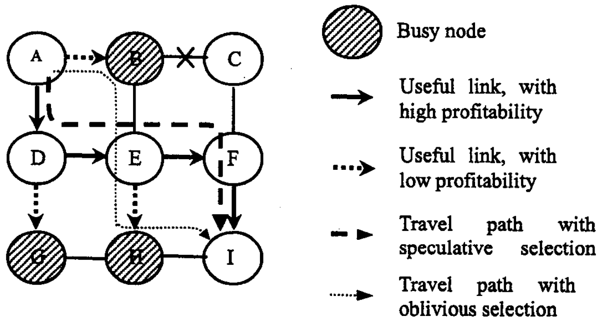

Figure 1. Message traverse using link's profitability.

2.2 Level of Speculation

The level of speculation defines how far a node is looking forward to do selection. In the one-hop configuration (Figure 2), the current node A is only collecting information from its next nodes B, C, D and E. Logically, we can see that two-hop configuration of node A consist of 12 nodes noted from B to M . Because the role of diagonal nodes F, G, H, I on link AB, AC, AD, AE are symmetric, we can eliminate those node's information in selective calculation. This configuration then composes of current node A in receiving information from 8 neighboring nodes B, C, D, E and J, K, L, M. The configuration then becomes easier to implement and much more practical than the three-hop configuration, which would consist of 20 neighbors.

3. Information exchange mechanism

The two-hop configuration was implemented to evaluate speculative selection effectiveness. Figure 3 illustrates how nodes exchange LUC (Local Utilization Counter) to other nodes. The LUC represents how busy

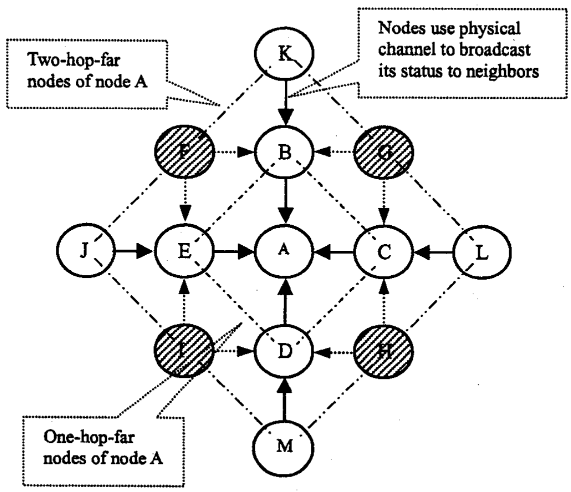

Figure 2. Level of speculation.

is a node by counting the number of messages currently stay in the node. Each node contains network condition buffers (NCB) that are classified into 1-hop and 2-hop types. In the T_n transaction, the LUC's value of node A is transmitted to node B's NCB 1-hop. In the next T_{n+1} transaction, the LUC value of node A at the T_n time slice, is transmitted from NCB 1-hop of node B to NCB 2-hop of node K. Simultaneously, the NCB 1-hop in the node B is updated with the new value of node A's value at the T_{n+1} time slice. The process is periodically iterated to keep the information up-to-date. During the transaction, the routers use physical links and temporary halt the message transmission.

4. Speculative Selection Algorithm

In order to calculate the output channel, we define the set of input status:

- The link's (virtual channel) availability F= *{free, busy}*.
- The link's dimension D: D = 1 if the message came from the same link's dimension, otherwise D = 0.
- The link's current request number *R*
- The next-node LUCs belong to the link value N_1 *and* N_2, which are corresponding one-hop-far and two-hop-far values

The selection function evaluates $E = D + R + N_1 + kN_2$. With two-hop configuration, k = 0.5 to decrease the weight of N_2. The best channel is the channel with minimal E value if there are multiple free channels or all channels are busy.

5. Simulation Results

Using the Opt-y adaptive wormhole routing function [1], we implemented simulation with two-hop speculative selection function. The simulation has been done with 16x16-mesh network and fixed message length of 20 flits. Both routing function and speculative selection function can be performed in 1 cycle. The information exchange process occupies physical links 10 cycles (2 cycles for handshaking and 8 cycles for exchanging LUCs) in every 100 cycles. The simulation results (Figures 4 and 5) show that the speculative selection routing (SSR) improves both the throughput and latency of the network compared with the oblivious straight selection function of Opt-y, especially in the saturation region. This is due to messages consistently avoid hotspots and concentrated region that leads to gradually balance of traffic in the network.

6. Conclusion

This paper presented a new approach to construct selection functions with speculation for multi computer direct interconnection networks. The SSR improves the performance in the saturated region and proves that the SSR substantially reduces latency in the router. This means the traffic is more balanced all over the network. More experiments in other routing functions, network topologies, and traffics should be done in the future.

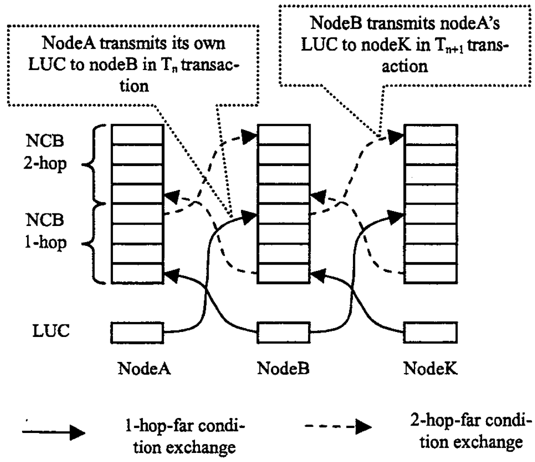

Figure 3. Two-hop information exchanges.

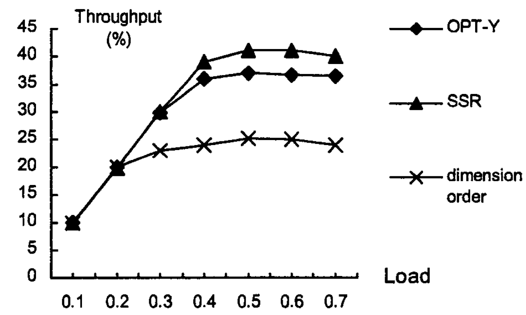

Figure 4. Throughput vs. Load. Hotspot Traffic.

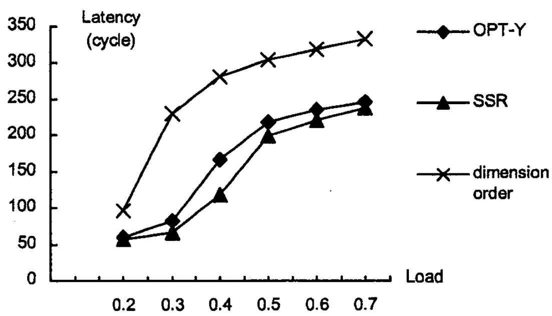

Figure 5. Latency vs. Load. Hotspot Traffic.

References

[1] L. Schwiebert and D. N. Jayasimha, "Optimal Fully Adaptive Minimal Wormhole Routing for Meshes," Journal of Parallel and Distributed Computing, Vol.27, No.1, pp.56-70, 1995.

[2] Loren Schwiebert, "A Performance Evaluation of Fully Adaptive Wormhole Routing including Selection Function Choice," IEEE International Performance, Computing, and Communications Conference, pp. 117-123, 2000.