Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS)
November 3-6, 1999, Boston (MA), USA

Traffic Control in Wormhole Routing Meshes under Non-Uniform Traffic Patterns

DIRK O. KECK
Institute of Communication Networks and Computer Engineering, University of Stuttgart
Pfaffenwaldring 47, 1st Floor
D-70569 Stuttgart, Germany
Tel.: (+49)-711-685 7964, Fax: (+49)-711-685 7983
keck@ind.uni-stuttgart.de

MICHAEL JURCZYK
Department of Computer Engineering and Computer Science, University of Missouri - Columbia,
121 Engineering Building West
Columbia, Missouri 65211, USA
Tel.: (+1) 573-884-8869, Fax: (+1) 573-882-8318
mjurczyk@cecs.missouri.edu

Abstract: *Nonuniform traffic patterns can severely degrade the performance of wormhole-routing mesh networks in multiprocessor systems. For example, under a temporary hot-spot traffic, a saturation tree might build up temporarily within the network resulting in a temporary network overload that will delay messages substantially. To the knowledge of the authors, no mechanisms were proposed in the open literature so far that are able to control (rather than just route) the traffic flow under those traffic scenarios. This paper introduces and studies several channel assignment strategies for wormhole routers using virtual channels. One proposed assignment strategy is able to effectively control the degrading effects of saturation trees on the uniform background traffic under nonuniform traffic patterns that are known a priori.*

Keywords: mesh networks, network overload, temporary hot-spot, traffic control, wormhole-routing

1. Introduction

In many parallel processing systems, direct interconnection networks are used to interconnect the processing elements (PEs), or to connect the processors with memory modules, e.g., Intel Paragon, Maspar MP-1, Cray T3E, or the Connection Machine CM-2 [1, 2]. In this paper, wormhole-routing mesh network organizations are assumed that interconnect the processors in a MIMD parallel computer.

Hot-spot traffic, in which many processors send data (hot messages) to the same destination, can cause congestion within the network and degrade the overall network performance substantially [3, 4]. If the traffic rate to the hot memory exceeds a certain threshold, a saturation tree of full router buffers builds up from the hot destination – the network is overloaded and even messages not destined to the hot-spot destination are delayed substantially [3, 5]. A variety of mechanisms such as synchronization, Gaussian Elimination algorithms, cache coherence protocols, and the access of a single shared variable by multiple processors can cause hot-spot traffic patterns in multiprocessor systems [3, 5, 6]. The negative influence of those hot-spots on the overall network traffic has to be alleviated to obtain high performance in multiprocessor systems.

Several concepts to alleviate saturation tree effects on network performance in direct and indirect networks have been proposed. These concepts can be divided into four classes: (1) combining techniques [7], (2) flow control techniques [8], (3) enhanced switch box and router designs [4, 9, 10], and (4) adaptive routing protocols [11]. Hardware combining methods do not work for most routing protocols in mesh networks. Flow control techniques, like feedback or discarding networks [8], often result in decreased performance under uniform traffic. Most enhanced router designs for wormhole-routing direct networks (e.g., [9]) are unable to *control* the traffic flow under nonuniform traffic patterns. In direct networks, adaptive routing protocols are able to route messages around congested network areas [11] – however, under hot-spot traffic patterns, the overall network might become congested, so that adaptive routing is unable to find lesser congested routes.

To the knowledge of the author, no concepts for the traffic *control* (rather than just traffic *routing*) in wormhole-routing mesh networks are proposed in the open literature so far. This paper introduces new router mechanisms that can effectively control the traffic flow under nonuniform traffic patterns without a performance degradation under uniform traffic scenarios. Their performance will be compared to the performance of meshes comprising conventional wormhole routers to illustrate the superiority of the proposed mechanisms. On the one extreme, saturation tree formation can be fully suppressed so that an optimal uniform message delay without any network overloads is achieved. On the other extreme, a minimized hot-spot phase length can be achieved. Performance characteristics in-between those two extreme cases can be achieved as well.

2. Interconnection Network and Traffic Models

2.1 Network Topology and Routing

The network model assumed in this paper is a wormhole-routing 2-dimensional mesh. Each *node* of the parallel system under investigation consists of a processing element (processor/memory pair) and a router, as shown in Figure 1. The router handles all message transmissions (including message segmentation and routing decisions) among the processing nodes. It is assumed that if a message has to traverse intermediate nodes to reach its destination, the message will not be delivered to the intermediate node processors but will stay in the intermediate routers.

Wormhole-routing is a switching technique in which a message is divided into several *flow-control digits* (*flits*) [12], which are the smallest unit of information transmitted between two nodes at once. The head of each message consists of one or more flits which contain the destination (routing) information, while the last flit of each message contains an “End of Message” indication. To avoid deadlocks, a deterministic deadlock-free dimension-ordered routing algorithm (XY-routing) for meshes [12] is assumed in this study. The impact of other routing protocols on the performance of the proposed router mechanisms is part of future work.

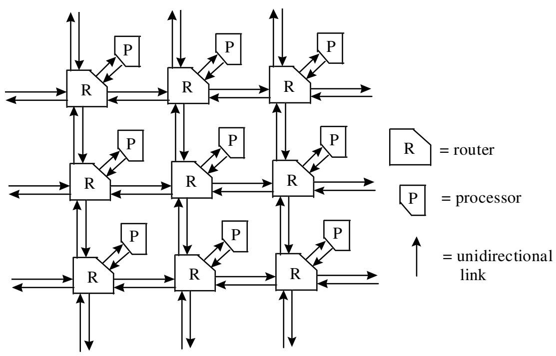


Figure 1: Part of a two-dimensional mesh network

The router architecture assumed in this paper is depicted in Figure 2. VC_{max} *virtual channels* (implemented through multiple parallel flit buffers at each router input port (see Figure 2)) are present at each physical router port to increase the performance of wormhole-routing networks [13]. Multiplexing of the virtual channels is carried out in a round robin manner.

When a message is generated by a node it is stored in a message queue in the processor/router interface and segmented into individual flits (M→F in Figure 2) before being sent into the network. To increase the performance of a network under nonuniform traffic patterns substantially, hot and cold messages are buffered in separate parallel buffers in the source node and flits are time-multiplexed from both queues into the router (see Figure 2) [4, 10]. Also, it is assumed that each router provides four physical consumption channels (one for each of the router inputs), and each physical consumption channel supports VC_{max} virtual consumption channels (one for each virtual channel of the corresponding router input) (see Figure 2). This setup results in the best network performance even under nonuniform traffic patterns but it also results in the highest hardware overhead [9]. It was chosen to represent a best-case scenario (even under this scenario, nonuniform traffic patterns will still degrade network performance substantially).

Throughout the paper, message delay is measured in network cycles. A **network cycle** is defined as the minimal time to transfer a flit out of a router flit buffer through the router, over a router link, and store the flit in the receiver flit buffer.

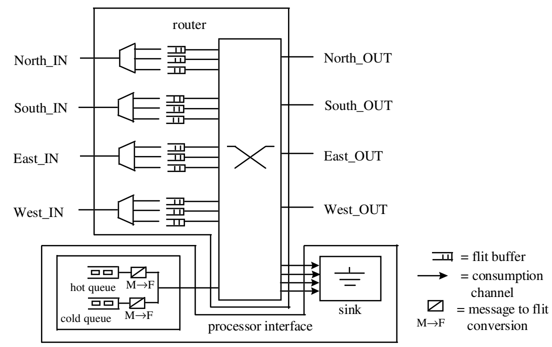


Figure 2: Router architecture with three virtual channels per port

2.2 Traffic Models

For **uniform traffic**, each processor generates uniform messages with a fixed length of FU flits. The destinations of these messages are uniformly distributed. The traffic is characterized by the **uniform traffic load** λ ($0 \leq \lambda \leq 1$), defined as a fraction of the network’s capacity [12].

If a set of processors accesses a single shared variable simultaneously, the processors belonging to the set send a message (hot message) to the memory module the variable resides in, which results in a **temporary hot-spot traffic pattern**. Because the processors in a MIMD system are independent, they send their hot messages at different times. One way to model such hot message generation is a normal distribution with a mean μ and a standard deviation σ as proposed in [6]. It is assumed that each hot message has a fixed length of FH flits (with $FH < FU$ for typical synchronization scenarios). Furthermore, it is assumed that while the hot-spot access is in progress, the processors perform a fast context switch (e.g., as in the HEP supercomputer [2]) and continue to work on a different program in the meantime. To model this traffic behavior, it is thus assumed that each processor generates uniform traffic with load λ before and after it emits a hot message. This kind of hot-spot scenario was chosen as one type of worst-case dynamic traffic pattern in interconnection networks by many researchers [5, 6].

In many applications, *a priori* knowledge about occurrences of unsymmetrical traffic patterns is available, so that each processor of a parallel computer can distinguish different message classes [6], e.g., through the use of an intelligent compiler. Under hot-spot traffic scenarios, **hot messages** and **uniform messages** can be distinguished and marked prior to entering the network.

The effect of temporary hot-spots in networks with conventional wormhole routers is studied in the following. To simplify the following discussions, 32×32 two-dimensional mesh networks using conventional wormhole routers (Figure 2) are considered with four virtual channels per physical port, each with a flit buffer length of four flits. A temporary hot-spot traffic is assumed with FU = 20, FH = 5, λ = 0.6, μ = 5000, and σ = 50. Also, it is assumed that all 1024 processing nodes are involved in the synchronization process. Throughout the paper, the hot destination node was chosen to be located in the network middle rather than at the network edge. Because of the missing edge connections in a mesh network, links connecting the edge nodes carry more traffic than links connecting nodes in the middle of the network under uniform traffic. This choice therefore results in less network congestion under the hot-spot traffic pattern. A parallel network simulator [14] running on an Intel Paragon MIMD computer [1] was used. A simulation result shown at cycle t in Figure 3 is the average of the results in a moving time window of 100 cycles (from cycle t-99 to cycle t) of 10 independent simulation runs. The resulting delay of the hot messages over time is depicted in Figure 3(a), while Figure 3(b) shows the delay of the uniform background messages over time.

When no saturation tree is present, the uniform message delay is approximately 100 cycles under that particular traffic load. The temporarily filled flit buffers during the hot-spot phase result in a delay increase of the uniform messages of up to 930 network cycles under that scenario. After most of the hot messages reached their destination (all hot messages have traversed the network at cycle 11,000 as depicted in Figure 3a), the flit buffers in the saturation tree start to empty so that the average uniform message delay decreases again until the tree has completely vanished (see Figure 3b).

Considering Figure 3, two overlapping phases can be defined during temporary hot-spot traffic scenarios [10]: 1) the **hot-spot phase**, i.e., the time interval of length T_h from the injection of the first hot message into the network until all hot messages left the network (see Figure 3a), and 2) the **overload phase**, i.e., the time interval of length T_o from the first detection of a network overload until the end of the overload (see Figure 3b).

Depending on the parallel application, a short hot-spot phase might be preferable over a long overload phase, or vice versa. Thus, to increase network performance under nonuniform traffic scenarios that result in a temporary network overload, a mechanism is needed that can control network behavior depending on the applications' needs. One router mechanism proposed in this paper is able to implement this traffic control.

3. Virtual Channel Assignment Strategies

If an *a priori* knowledge about occurrences of nonuniform traffic patterns is available, messages can be marked as hot or cold and the network routers can treat them differently. The traffic control mechanisms proposed in this paper are based on the restricted use of virtual channels of a physical link for hot messages. If a router provides VC_{max} virtual channels for a physical link, then only $\boldsymbol{VC}$ ($0 < VC \leq VC_{max}$) of these VC_{max} virtual channels can be used simultaneously by hot messages during any network cycle.

Assume that a router has received the first header flits of a message. In order to transfer the message to a neighboring node, at least one virtual channel belonging to the corresponding router output port (that is connected to that neighbor node) must be available. A virtual channel is **available** if it is not currently used by a message (i.e., the last flit sent over that channel was a tail flit of a message) and if there is space for at least one flit in the flit buffer at the input side of the receiving router. If there are multiple virtual channels available, the router must select a particular one.

- **Random Selection Strategy**: One of the multiple available virtual channels is selected randomly. This strategy is most often used in conventional wormhole routers [12].
- **Empty-First Selection Strategy**: If, among the available virtual channels, there exist channels that have an empty flit buffer at the receiving router side, one of these channels will be selected randomly. If no empty receiver buffers are present, then the message will be assigned to an available virtual channel that has flits buffered in the receiver flit buffer that belong to the same message class as the message to be sent over that link.

Compared to the random selection strategy, the

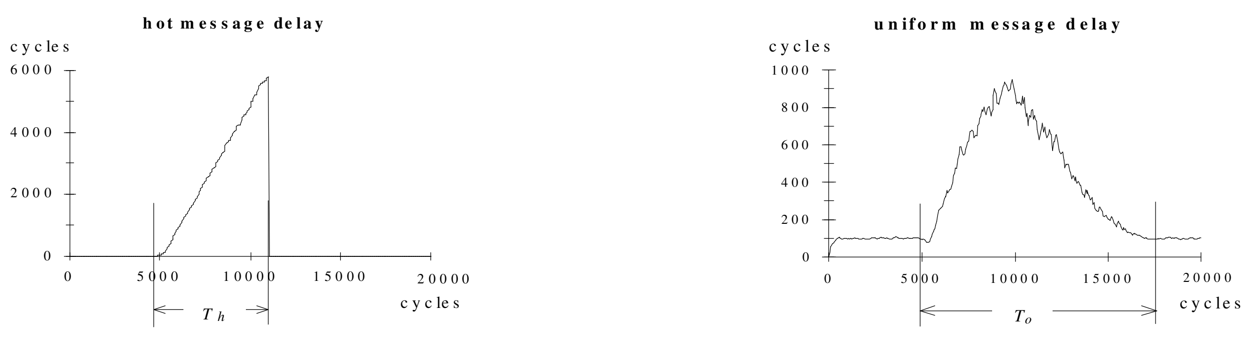


Figure 3: Delay of (a) hot and (b) uniform messages in a 32×32 mesh with 4 virtual channels per router port under hot-spot traffic with FU=20, FH=5, λ=0.6, μ=5000, σ=50

empty-first strategy requires additional hardware to be implemented. There has to be an additional control line for each virtual channel at each router port to signal whether the corresponding virtual channel buffer is empty or not. Each router also needs to keep track of the message type that was last assigned to a virtual channel.

To control the traffic flow under nonuniform traffic scenarios, only VC virtual channels can be used by hot messages simultaneously during any network cycle. The following two additional assignment strategies can be used for hot messages:

• **Fixed Hot Assignment Strategy**: VC virtual channels of any physical link are reserved for hot messages exclusively (cold messages will never be assigned to those channels). Because cold messages can only use $VC_{max} - VC$ virtual channels, the network performance under pure uniform traffic might be less than in conventional meshes, especially for a larger VC. Also, simulations show that for a larger VC, networks are unable to recover from a network overload caused by a temporary hot-spot traffic pattern. Thus, this assignment strategy cannot efficiently control the traffic flow and will therefore not be considered further in this paper.

• **Variable Hot Assignment Strategy:** In this case, channels are not reserved for hot messages exclusively. An available channel is assigned to a hot message as long as less than VC channels are currently used by hot messages. This strategy ensures that, under pure uniform traffic, network performance will not be less than in conventional meshes (in the uniform traffic case, the enhanced router functionality degrades to that of a conventional router).

The first two selection strategies can be combined with the last one, resulting in two overall assignment strategies studied further in this paper:

1. **RV** (**random selection, variable assignment**): Available virtual channels are assigned randomly to messages; channels are not exclusively reserved for hot messages. If $VC = VC_{max}$, this router is equivalent to a conventional wormhole router, even under nonuniform traffic.
2. **EV** (**empty-first selection, variable assignment**): Available virtual channels with empty receiver flit buffers are assigned first; channels are not exclusively reserved for hot messages.

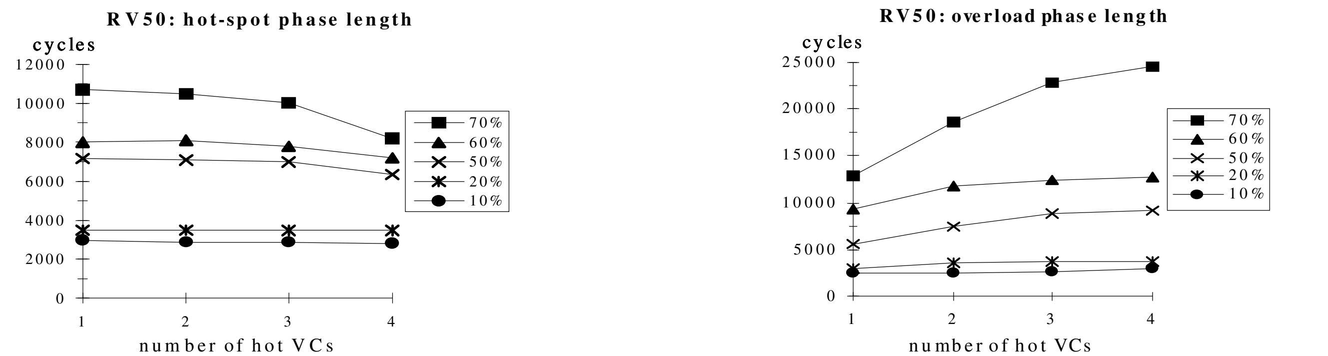
• **Hot virtual channel count:** A router has to count how many virtual channels at each output port are currently used by hot messages. For this reason, a hot counter is needed at each router output port. The update mechanism of this hot counter has a major impact on network performance, as will be discussed in the following.

The only mechanism to update the counter in an RV router is to increment the counter when a hot header flit is assigned to a virtual channel and to decrement the counter when a hot tail flit leaves the router output port (because there is no feedback from the receiver router other than whether the receiver flit buffer is full or not). This might result in an incorrect hot channel count. A hot message that has left an RV router output port might still occupy a virtual channel buffer at the neighboring router's input port. Thus, this virtual channel is still used by that hot message which is not reflected in the hot channel count at the sending RV router output port. To avoid this scenario, a router needs to know if the end of a hot message is still buffered in the flit buffer of the receiving router. Only routers using the empty-first selection strategy have the knowledge whether a flit buffer at the receiver side is empty or not. Therefore, if EV routers are used, an advanced hot counter update strategy should be used. The counter at the sending router's output port will only be decremented if a hot tail flit leaves the flit buffer in the receiving router and leaves this buffer empty. This way, the counter will always represent the actual number of flit buffers at the receiver side that are partially or fully filled with hot flits (of course, this can only work if only messages of the same type are currently buffered in any flit buffer, which is guaranteed by the EV assignment strategy). Therefore, throughout this paper, this advanced counter update mechanism will be used in conjunction with the EV assignment strategy, while RV routers will use the simple update method.

In the next section, the performance of these two assignment strategies (RV, and EV with the advanced counter update) in meshes under temporary hot-spot traffic scenarios is studied and compared. It will be shown that the EV strategy is able to effectively control the traffic flow and the network performance under those traffic scenarios.

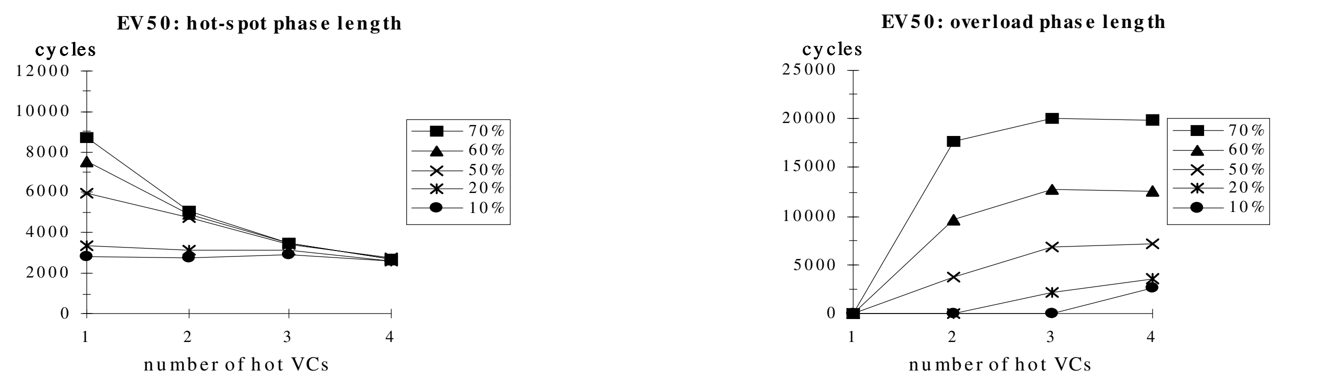
4. Performance of the Virtual Channel Assignments

The performance of the two assignment strategies introduced in the last section is exemplified by a 32×32 mesh with 4 virtual channels per router port under a temporary hot-spot traffic pattern with FU = 20, FH = 5, μ = 5000, and σ = 50, and uniform traffic loads of 10%, 20%, 50%, 60%, and 70%. The hot-spot phase and overload phase lengths achievable with the RV strategy are depicted in Figure 4, while the phase lengths for the EV strategy are shown in Figure 5. The RV network is unable to fully suppress any saturation trees. For VC = 1, EV is able to fully suppress any network overloads (T_o = 0) while the hot-spot phase length is significantly smaller than in the RV case. With increasing VC, EV is able to decrease the hot-spot phase length which is traded-off by a longer overload phase. However, even for $VC = VC_{max}$ = 4, the overload phase length is still smaller than in the RV case (which is the operation of a conventional wormhole router in the RV case), while a minimal hot-spot phase length is achieved. Thus, the RV strategy can (slightly) control the traffic flow; however, it is not possible to fully suppress any saturation trees and therefore any network overloads or to achieve a minimal hot-spot phase length, which can be achieved with the EV strategy. To judge the performance gain using EV routers, the length of the hot-spot phase and the overload phases achievable with EV routers and with conventional wormhole routers are depicted in Figure 6. Only for high traffic loads and $VC = 1$, the conventional router achieves a slightly shorter hot-spot phase (see Figure 6a). The EV router is able to achieve significantly shorter hot-spot and overload phases for all other cases. Also, for $VC = 1$, network overloads can be fully eliminated for the traffic load range depicted. For example, for $VC = 1$, network overloads of up to 24,500 network cycles in length (when using conventional wormhole routers) can be fully eliminated (when using wormhole routers with the EV assignment strategy), which is a substantial performance gain. Thus, the EV strategy with the advanced counter update is able to effectively control the traffic flow under nonuniform traffic patterns. Also, the EV router outperforms conventional wormhole routers in almost all cases. The disadvantage of the EV strategy is the hardware overhead as compared to a conventional wormhole router. An additional control line is needed for each virtual channel at each router port to signal whether the corresponding flit buffer is empty or not. This will



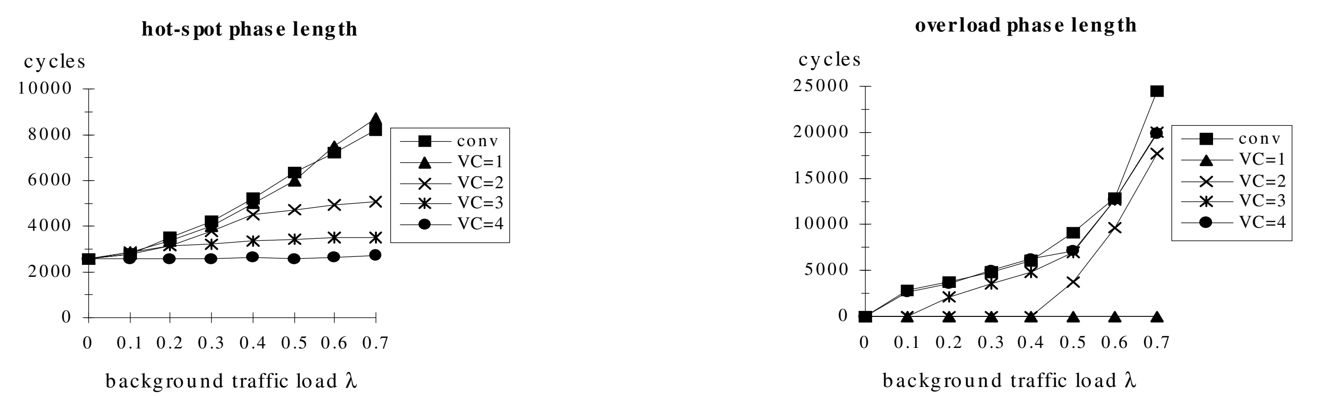
(a) (b)

*Figure 4: (a) Hot-spot and (b) overload phase length in a 32×32 mesh with 4 virtual channels and the **RV** strategy under hot-spot traffic (FU=20, FH=5, μ=5000, σ=50)*



(a) (b)

*Figure 5: (a) Hot-spot and (b) overload phase length in a 32×32 mesh with 4 virtual channels and the **EV** strategy under hot-spot traffic (FU=20, FH=5, μ=5000, σ=50)*



(a) (b)

Figure 6: (a) Hot-spot and (b) overload phase length in a 32×32 mesh with conventional and EV routers with 4 virtual channels under a hot-spot (FU=20, FH=5, μ=5000, σ=50)

restrict the implementation of the EV strategy to routers with a moderate number of virtual channels per link.

Extensive simulations show that a variation of the network size, number of virtual channels per physical link, and buffer and message lengths has an influence on the absolute values of the phase lengths, while the overall network behavior under nonuniform traffic scenarios is not affected by it. Results and conclusions drawn throughout this paper are therefore valid for a wide range of different network and router configurations, and message lengths. Also, simulations with other traffic patterns in which saturation trees can build up also demonstrate the advantages of the EV assignment strategy. Examples of such traffic patterns are transient partial hot-spot traffics, and a broad variety of permutation patterns that result in nonuniform traffic spots (NUTS) (e.g., the bit-reverse permutation traffic or the transpose traffic [15]). The network performance degradation due to those traffic patterns can be effectively controlled with the EV assignment strategy as well.

5. Adaptive EV Assignment Scheme

Depending on the parallel application, a short hot-spot phase might be preferable over a long overload phase, or vice versa. For example, a short hot-spot phase ensures fast synchronization. However, if a second synchronization follows shortly after the first one, it might encounter an overloaded network (resulting from the first synchronization) and might be delayed substantially. In this case, a longer hot-spot phase but a shorter overload phase for the first synchronization might improve overall application performance. To account for these different application needs, an **adaptive EV assignment scheme** can be employed. Assuming that for all routers within a network of a parallel machine the parameter VC can be altered at run-time by the machine, a user and/or an advanced compiler can determine whether the hot-spot phase length or the overload phase length is of more importance for the optimal performance of a hot-spot producing program section so that the network can be controlled accordingly during program execution. To suppress any saturation tree effects on the uniform background traffic, VC should be set to 1, while VC should be set to a high value (e.g., $VC = VC_{max}$) to obtain a minimized hot-spot phase length.

6. Conclusion

Nonuniform traffic patterns can severely degrade the performance of networks in multiprocessor systems. To the knowledge of the authors, no mechanisms were proposed in the open literature so far that are able to *control* (rather than just to route) the traffic flow in mesh networks under those traffic scenarios. This paper introduces and studies two channel assignment strategies for wormhole routers with virtual channels to be used in mesh networks. It is shown that the choice of virtual channels assigned to hot messages and the counting method of hot virtual channels have a major impact on network performance. The EV (empty-first, variable assignment) strategy with an advanced counter update mechanism is able to effectively control the degrading effects of saturation trees on the uniform background traffic under nonuniform traffic patterns that are known *a priori*. On the one extreme, saturation trees and network overloads can be fully suppressed, while on the other extreme, a minimized hot-spot phase (which would result in a minimal synchronization time if the nonuniform data traffic stems from a synchronization) can be obtained. Performance characteristics in-between those two extreme cases can be achieved as well. Also, the EV router outperforms conventional wormhole routers in almost all cases. To accompany the needs of different applications, an adaptive assignment scheme can be employed.

REFERENCES

[1] M. Jurczyk, H. J. Siegel, and C. Stunkel, “Interconnection Networks for Parallel Computers” in *Encyclopedia of Electrical and Electronics Engineering Volume 10* , J. G. Webster, ed., John Wiley and Sons, New York, NY, 1999, pp. 555-564.

[2] T. Schwederski and M. Jurczyk, *Interconnection Networks: Structures and Properties (in German)*, Teubner Verlag, Stuttgart, Germany, 1996.

[3] S. P. Dandamudi, “Reducing hot-spot contention in shared memory multiprocessor systems,” *IEEE Concurrency*, to appear, 1999.

[4] M. Jurczyk and T. Schwederski, “Phenomenon of higher order head-of-line blocking in multistage interconnection networks under nonuniform traffic patterns,” *IEICE Trans. on Information and Systems*, Vol. E79-D, No. 8, August 1996, pp. 1124-1129.

[5] M. Charney, “The role of network bandwidth in barrier synchronization,” *Journal of Parallel and Distributed Computing*, Vol. 28, No. 2, August 1995, pp. 79-107.

[6] S. Abraham and K. Padmanabhan, “Performance of the direct binary n-cube network for multiprocessors,” *IEEE Trans. on Comp.*, Vol. C-38, No. 7, July 1989, pp. 1000-1011.

[7] P.-C. Yew, N.-F. Tzeng, and D. H. Lawrie, “Distributing hot-spot addressing in large-scale multiprocessors,” *IEEE Trans. on Comp.*, Vol. C-36, No. 4, 1987, pp. 388-395.

[8] W.S. Ho and D.L. Eager, “A novel strategy for controlling hot spot congestion,” *1989 International Conference on Parallel Processing*, August 1989, pp. 14-18.

[9] D. Basak and D. K. Panda, “Alleviating consumption channel bottleneck in wormhole-routed k-ary n-Cube systems,” *IEEE Transactions on Parallel and Distributed Systems*, Vol. 9, No. 5, May 1998, pp. 481-496.

[10] M. Jurczyk and T. Schwederski, “Switch box architecture for saturation tree effect minimization in multistage interconnection networks,” *1995 International Conference on Parallel Processing*, August 1995, pp. I/41-I/45.

[11] P. T. Gaughan and S. Yalamanchili, “Adaptive routing protocols for hypercube interconnection networks,” *IEEE Computer*, Vol. 26, No. 5, May 1993, pp. 12-23.

[12] L. M. Ni and P. K. McKinley, “A survey of wormhole routing techniques in direct networks,” *IEEE Computer*, Vol. 26, No. 2, February 1993, pp. 62-76.

[13] W. J. Dally, “Virtual-channel flow control,” *IEEE Trans. on Parallel and Distributed Systems*, Vol. 3, No. 2, March 1992, pp. 194-205.

[14] D. O. Keck and M. Jurczyk, “Parallel discrete event simulation of wormhole routing interconnection networks,” *IASTED International Conference on Parallel and Distributed Computing Systems*, October 1998, pp. 391-394.

[15] T. Lang and L. Kurisaki, “Nonuniform traffic spots (NUTS) in multistage interconnection networks,” *1988 ICPP*, August 1988, pp. 191-195.