Moadeli, M. and Vanderbauwhede, W. (2009) *A performance model of multicast communication in wormhole-routed networks on-chip.* In: IEEE International Symposium on Parallel & Distributed Processing, 2009. IPDPS 2009. , 23-29 May 2009, Rome, Italy.

A Performance Model of Multicast Communication in Wormhole-Routed Networks on-Chip

Mahmoud Moadeli and Wim Vanderbauwhede
Department of Computing Science
University of Glasgow
Glasgow, UK
Email: {mahmoudm, wim}@dcs.gla.ac.uk

Abstract

Collective communication operations form a part of overall traffic in most applications running on platforms employing direct interconnection networks. This paper presents a novel analytical model to compute communication latency of multicast as a widely used collective communication operation. The novelty of the model lies in its ability to predict the latency of the multicast communication in wormhole-routed architectures employing asynchronous multi-port routers scheme. The model is applied to the Quarc [17] NoC and its validity is verified by comparing the model predictions against the results obtained from a discrete-event simulator developed using OMNET++.

1 Introduction

Traditionally, interconnect architectures for integrated circuits have been bus-based. Driven by the advances in semiconductor technologies reaching sub-$0.1\mu m$ gate lengths, systems-on-chip (SoC) consisting of billions of gates and hundreds of processing units operating at different clock frequencies are becoming reality. As a bus is inherently non-scalable and at the same time the size and complexity of the future SoC does not allow starting the whole design from the scratch, employing a modular type of architecture seems inevitable. Communication centric architectures or "networks on chip" (NoC) have recently been proposed as a solution for the interconnect problem in large SoC designs [21].

Collective communications operations have been traditionally adopted to simplify the programming of applications for parallel computers, facilitate the implementation of efficient communication schemes on various machines, and promote the potability of applications across different architectures [10]. These communication operations are particularly useful in applications which often require global data movement and global control in order to exchange data and synchronize the execution among nodes. The most widely used collective communication operations are *broadcast*, *multicast*, *scatter*, *gather* and *barrier synchronization*.

The support for collective communication may be implemented in *software* or/and *hardware*. The software-based approaches [9] rely on unicast-based message passing mechanisms to provide collective communication. They mostly aim to reduce the height of multicast tree and minimize the contention among multiple unicast messages.

Software-based approaches typically have limitations in delivering the required performance. Implementing the required functionality partially or fully in hardware has proved to improve the performance of collective operations. Depending on required performance, the hardware support for collective communication may be achieved by customizing the switching [5], routing, number of ports [8] or even allocating a dedicated network for collective communication operations.

Hardware-based multicast schemes can be broadly classified into *path-based* and *tree-based.* In a path-based approach, the primary problem for multicasting is finding the shortest path that covers all node in the network [10]. After path selection, the intermediate destinations perform absorb-and-forward operations along the path. Hamilton path-based algorithm [6] and the Base Routing Conformed Path (BRCP) approach [1] are examples of path-based algorithms utilizing absorb-and-forward property at hardware layer.

In the tree-based scheme, the multicast problem is finding a Steiner tree with a minimal total length to cover all network nodes [3]. The tree operation introduces additional network resource dependencies which could lead to deadlock which is difficult to avoid if global information is not available. Hence, in wormhole-routed direct networks, the tree based multicast is usually undesirable, unless the mes-

978-1-4244-3750-4/09/$25.00 ©2009 IEEE

sages are very short.

Broadcast and multicast traffic in Networks on Chip is an important research field that has not received much attention. A multicasting scheme for a circuit-switched network on chip is proposed in [11]. Since the scheme relies on the global network state using global traffic information it is not easily scalable. Multicast operation is provided by Æthereal NoC [13]. However, Æthereal relies on a logical notion of global synchronicity which is not trivial to implement as the system scales. In [4] a multicast scheme in wormhole-switched NoCs is proposed. By this scheme, a multicast procedure consists of establishment, communication and release phase. A multicast group can request to reserve virtual channels during establishment and has priority on arbitration of link bandwidth. In [17] the novel Quarc NoC architecture is introduced to offer highly efficient collective communication operations by employing a *BRCP* broadcast/multicast routing algorithm and a multi-port router architecture.

The literature has witnessed numerous analytical performance model of the unicast traffic [12, 16] and analysis of the unicast traffic in presence of the broadcast traffic [7] in parallel computers and NoC domains. In [18] Shahrabi et al. introduced a model for computing broadcast communication latency in Hypercube. However, in their system under model only the unicast was wormhole-routed and the broadcast communication was not wormhole-routed. Also, their model was developed for architectures adopting *one-port* routers scheme. To the best of our knowledge this work presents the first analytical model to compute the average multicast communication latency in a system adopting wormhole-switching for both unicast and multicast/broadcast communications. The novelty of the method lies in its ability to predict the average message latency in interconnection networks employing multi-port routers.

The rest of the paper proceeds as follows. The next section introduces a method for analyzing the average message latency of multicast communication in all-port wormhole-routed interconnection networks. Section 3 presents a brief description of the Quarc NoC and the broadcast/multicast routing algorithm in the architecture. Section 4 compares our analytical evaluation with simulation results and finally, in Section 5 the conclusion and future works are presented.

2 The Analysis Method

This section introduces a model to evaluate the average message latency of multicast communication in wormhole-routed interconnection networks generating both unicast and multicast/broadcast traffic. We assume that the network employs multi-port routers.

In direct interconnection networks, a router is connected to other neighboring routers through a number of external links. The router is also connected to the local node via one or more internal links. The architectures that adopt only one internal link are referred to as *one-port* architectures. Increasing the number of internal links significantly improves the performance of the collective communication operations [8]. Architectures having an internal link corresponding to each external link are referred to as *all-port* architectures. The schematic of router in a one-port and multi-port router architectures have been depicted in Fig. 1

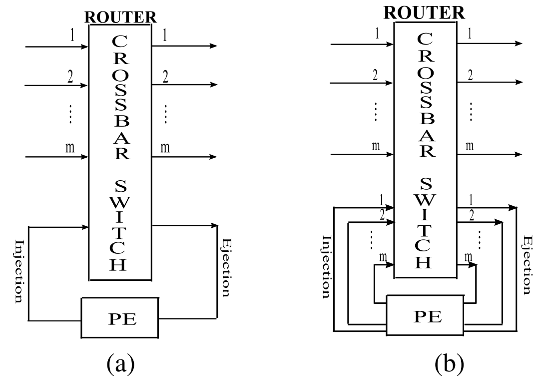

Figure 1. One-port(a) versus multi-port(b) router architecture

In an interconnection network employing multi-port routers scheme, the multicast latency may be defined as the time from the generation of the message at source node until the time when the last flit of the multicast message is absorbed by the last destination of the multicast message among messages leaving m injection ports. The novelty of the approach introduced in this paper lies surely in its ability to predict the average latency in networks adopting all-port or multi-port router architecture in which there is no synchronization between messages emerging from each port. Otherwise, the model was a variation of the available analytical models for modeling unicast communication.

The model uses some widely used assumptions in the literature [12, 16] to compute the average message latency of the unicast messages plus the specific assumptions regarding the multicast messages.

- Nodes are generating the unicast and multicast messages independently and according to a *Poisson* process.
- Unicast destination addresses are selected randomly.
- The arrival at each channel is approximated to be a *Poisson* process.
- Messages are all the same size and larger than the network diameter.

- The routing algorithm is deterministic.
- The network employs wormhole switching.
- The network employs multi-port routers.

In a multi-port router architecture employing deterministic routing, depending on the position of the destination node in the network, the appropriate injection ports should be taken to transmit the message through. We define $S_{j,c}$ as the subset of the network nodes receiving the multicast message initiated by node x_j through injection port i_c.

$$S_{j,c} = \{x_i : i_c \in \{multicast\, path\, from\, x_j\, to\, x_i\}\} \quad (1)$$

where

$$\bigcap_{c=1}^{m} S_{j,c} = \varnothing. \quad (2)$$

To multicast a message it should be sent to disjoint sub-networks through multi-ports of the router. It can be argued that the message latency experienced by the largest network subset can be regarded as the multicast communication latency. Although this may seem reasonable in most situations, the dynamic behavior of the traffic in different sub-networks may easily lead to situations in which the smaller sub-networks deliver messages later than larger networks do. Therefore, it is desirable to find a more reliable solution based on latencies experienced at each sub-network connected to different ports.

To compute the multicast message latency we divide the problem into two separate problems. In the first part the unicast message latency is computed which is, of course, the message latency experienced at each port of the router. In the second part, a method is proposed to compute the multicast latency using the results obtained from the first part.

2.1 Latency of Unicast Communication

The objective of this section is to present the model developed to evaluate the average message latency of unicast in the interconnection networks employing wormhole switching. We define the latency as the time from the generation of the message at source node until the last flit of the message is absorbed by the destination node.

The network is viewed as a network of queues, where each channel is modeled as an $M/G/1$ queue. For an $M/G/1$ queue the average waiting time is [14]

$$W_{M/G/1} = \frac{\lambda\rho}{2(1-\lambda x)}(1+\frac{\sigma^2}{x^2}) \quad (3)$$

$$\rho = \lambda x \quad (4)$$

where λ is the mean arrival rate, x is the mean service time and σ^2 is the variance of the service time distribution. The model defines σ as

$$\sigma = (x - msg) \quad (5)$$

where msg is the message length.

The service time at ejection channel equals to message length, msg, and service time at each intermediate channel from source to destination may be calculated as

$$\overline{x}_i^{in} = \sum_j ((1-\frac{\lambda_i^{in}}{\lambda_j}P_{i\to j})W_j + \overline{x}_j + 1)P_{i\to j} \quad (6)$$

where W_j is approximated using Eq. 3, x_j is the mean service time at channel j, λ_i^{in} is traffic rate from channel i to channel j, λ_j is traffic rate at channel j and finally $P_{i\to j}$ is the probability of taking channel j after channel i. The detailed explanation of the analytical model is presented in Moadeli et al. [16].

The Eq. 6 can be adopted to compute the service time at all channel from the ejection channel at destination back to the injection channel at source node. Averaging message latency over all nodes in the network yield the average message latency in the network.

2.2 Latency of Multicast Communication

By adopting the analytical model explained in previous section, this section presents an approach to compute the mean multicast message latency in a wormhole-routed interconnection networks employing asynchronous all-port routers. It is important to note that, there is no form of synchronization between flit streams leaving different ports of a router. This means that, each injection port of the router transmits the multicast messages independently of other injection ports.

The communication latency experienced by a message is a factor of three components namely: message length, number of hops and the total waiting times at intermediate channels. Among those parameters, message length and number of hops are fixed, while the waiting time varies. The total waiting times at all intermediate channels from source to destination may be any non-negative real time number. Nevertheless, its average is the total of the waiting times which may be computed using the method explained in previous section. Using the above definition, the average latency of a unicast traffic at injection port i_c at node x_j may be expressed as

$$\overline{L}_{j,c} = \sum_l w_l + \overline{msg} + \overline{D}_{j,c} \quad (7)$$

where

- w_l is the waiting time experienced by the header flit at link l ($l \in source\,to\,destination\,path$).
- $\overline{L}_{j,c}$ is the average communication latency for a traffic leaving injection channel port i_c at node x_j.
- $\overline{D}_{j,c}$ denotes the (maximum) number of hops traversed by a message in sub-network $S_{j,c}$ originating from node x_j.

Therefore, for each individual injection port, i_c ($1 \leq c \leq m$), of an all-port router at node x_j, we are able to define an exponential distribution, $E_{1,\mu_{j,c}}$, which its expected time is the total waiting times (from source to destination) experienced by the header flit of multicast stream leaving injection channel i_c of node x_j. Using the above definition, $\mu_{j,c}$ is expressed as:

$$\mu_{j,c} \quad = \quad \frac{1}{\sum_{l} w_l} \;\; 1 \leq c \leq m,\, 1 \leq j \leq N. \qquad (8)$$

By associating the waiting times at each port of the routers to independent exponentially distributed random variables and recalling the definition of the message latency, the multicast waiting time will be defined as the expected time for the occurrence of the last event among m independent exponentially distributed random variables. Which is of course, the expected total waiting time experienced by the last message delivered to its destination among m messages transmitted at injection ports of the router.

To compute the expected time of the last event we use two properties of the exponential distributions

- The exponential distributions are memory-less.
- The minimum of independent exponential distributions is exponentially distributed. i.e.

$$P\left[min\left\{E_{1,\mu_1}, E_{1,\mu_2}\right\}\right] = e^{-(\mu_1+\mu_2)} \qquad (9)$$

using the above two properties we first compute the expected time for the last event in case of only two independent exponential random variable, $E[max\left\{E_{1,\mu_1}, E_{1,\mu_2}\right\}]$, and then generalize the method for $m \geq 2$ independent exponential random variables.

According to Eq. 9 the expected time for occurrence of the first event between two independent exponential distributions is

$$E\left[min\left\{E_{1,\mu_1}, E_{1,\mu_2}\right\}\right] = \frac{1}{\mu_1+\mu_2} \qquad (10)$$

Due to memory-less property of exponential distributions, the expected time for the next event after the occurrence of the first event is $\frac{1}{\mu_2}$ or $\frac{1}{\mu_1}$ depending on whether the first event has been fired by E_{1,μ_1}or E_{1,μ_2} respectively. The probability that E_{1,μ_1}or E_{1,μ_2} has been the first event, however, is related to μ_1 and μ_2. In other words, the probability that E_{1,μ_1}has been the first event is $P_{E_{1,\mu_1}} = \frac{\mu_1}{\mu_1+\mu_2}$ and the probability that E_{1,μ_2} has been the first event is $P_{E_{1,\mu_2}} = \frac{\mu_2}{\mu_1+\mu_2}$. Therefore, the expected time for the last event between two independent exponentially distribution is

$$E\left[max\left\{E_{1,\mu_1}, E_{1,\mu_2}\right\}\right] = \frac{1}{\mu_1+\mu_2} + P_{E_{1,\mu_1}} \times E[E_{1,\mu_2}] + P_{E_{1,\mu_2}} \times E[E_{1,\mu_1}]. \qquad (11)$$

Generalizing the solution for $m \geq 2$ yields the expected time for occurrence of the last event between m independent exponentially distributed events as

$$\begin{aligned} E\left[max\left\{E_{1,\mu_1}, E_{1,\mu_2}, ..., E_{1,\mu_m}\right\}\right] = \frac{1}{\mu_1+\mu_2+...+\mu_m} + \\ \frac{\mu_1}{\mu_1+\mu_2+...+\mu_m} E\left[max\left\{E_{1,\mu_2}, E_{1,\mu_3}, ..., E_{1,\mu_m}\right\}\right] + \\ \frac{\mu_2}{\mu_1+\mu_2+...+\mu_m} E\left[max\left\{E_{1,\mu_1}, E_{1,\mu_3}, ..., E_{1,\mu_m}\right\}\right] + ... \\ \frac{\mu_m}{\mu_1+\mu_2+...+\mu_m} E\left[max\left\{E_{1,\mu_1}, E_{1,\mu_2}, ..., E_{1,\mu_{m-1}}\right\}\right]. \end{aligned} \qquad (12)$$

Adopting the above analysis, the expected waiting time experienced by the last message (among m independent streams of a multicast message leaving node x_j) delivered to its destination, W_j, may be computed as

$$W_j = E\left[max\left\{E_{1,\mu_{j,1}}, E_{1,\mu_{j,2}}, ..., E_{1,\mu_{j,m}}\right\}\right]. \qquad (13)$$

Therefore, the average multicast latency at node x_j, $\overline{L}_j$, may be expressed as

$$\overline{L}_j = W_j + \overline{msg} + D_j \qquad (14)$$

where

$$D_j = Max(\overline{D}_{j,c}) \qquad 1 \leq c \leq m \qquad (15)$$

is the maximum hops traversed among m sub-networks connected to node x_j.

Averaging over all nodes in the network yields the average multicast message latency as

$$\overline{L} = \frac{1}{N} \sum_{j}^{N} \overline{L}_j. \qquad (16)$$

3 The Quarc Architecture

The Quarc scheme [17] was introduced as a NoC to provide high performance collective communication at low cost. The Quarc NoC inspired and is quite similar to the Spidergon scheme.

The topology of the Quarc NoC is quite similar to that of the Spidergon NoC. Therefore, the next section presents a brief description of the Spidergon NoC, followed by an introduction to the Quarc NoC.

3.1 The Spidergon NoC

The Spidergon NoC [15] is a network architecture which has been recently proposed by STMicroelectronics [19]. The objective of the Spidergon topology has been to address the demand for a fixed and optimized topology to realize low cost multi-processor SoC implementation.

In the Spidergon topology nodes are connected by unidirectional links. Let the number of nodes be an even $N = 2n$ (where n is a positive integer). Every node in the network is represented by x_i ($0 \leq i < N$). An arbitrary node is assigned label 0 and the label of other nodes is incremented by one as we move clockwise. The channels around the topology are given the same label as the nodes connected to them in clockwise direction. And the channels connecting cross network nodes are given label of the node with the lower index plus N. Each node in the network, x_i ($0 \leq i < N$). is directly connected to node $x_{(i+1)\,mod\,N}$ by a clockwise link, to node $x_{(i-1)\,mod\,N}$ by a counter-clockwise link and to node $x_{(i+\frac{N}{2})\,mod\,N}$ by a cross link. Each physical link is shared by two virtual channels in order to avoid deadlock.

The key characteristics of the the topology include good network diameter, low node degree, homogeneous building blocks (the same router to compose the entire network), vertex symmetry and simple routing scheme. Moreover, the Spidergon scheme employs packet-based wormhole routing which can provide low message latency at a low cost. Furthermore, the actual layout on-chip requires only a single crossing of metal layers.

In the Spidergon NoC, two links connecting a node to surrounding neighboring nodes carry messages destined for half of nodes in the network, while the node is connected to the rest of the network via the cross link. Therefore, the cross link can become a bottleneck. Also, since the router at each node of the Spidergon NoC is a typical one-port router, the messages may block on occupied injection channel, even when their required network channels are free. Moreover, performing broadcast communication in a Spidergon NoC of size N using the most efficient routing algorithm requires traversing $N - 1$ hops.

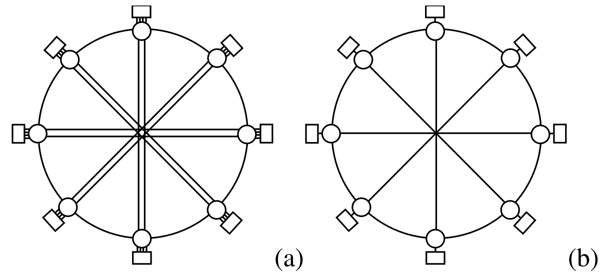

Figure 2. Quarc topology (a) vs Spidergon (b)

3.2 The Quarc NoC

The Quarc NoC improves on the Spidergon scheme by making following changes: *(i)* adding an extra physical link to the cross link to separate right-cross-quarter from left-cross-quarter, *(ii)* enhancing the one-port router architecture to an all-port router architecture and *(iii)* enabling the routers to absorb-and-forward flits simultaneously. The Quarc preserves all features of the Spidergon including the wormhole switching and deterministic shortest path routing algorithm, as well as the efficient on-chip layout.

The resulting topology for an 8-node NoC is represented in Fig. 2.

In the Spidergon NoC, deadlock-free broadcast/multicast can only be achieved by consecutive unicast transmissions. The NoC switches must contain the logic to create the required packets on receipt of a broadcast-by-unicast packet. In contrast, the broadcast/multicast operation in the Quarc architecture is a true broadcast/multicast, leading to much simpler logic in the switch fabric; furthermore, the latency for broadcast/multicast traffic is dramatically reduced.

3.3 Routing algorithm

3.3.1 Unicast routing

For the Quarc, the surprising observation is that there is no routing required by the switch: flits are either destined for the local port or forwarded to a single possible destination. Consequently, the proposed NoC switch requires no routing logic. The route is completely determined by the port in which the packet is injected by the source. Of course, the NoC interface (transceiver) of the source processing element (PE) must make this decision and therefore calculate the quadrant as outlined above. However, in general the PE transceiver must already be NoC-aware as it needs to create the header flit and therefore look up the address of the destination PE. Calculating the quadrant is a very small additional action.

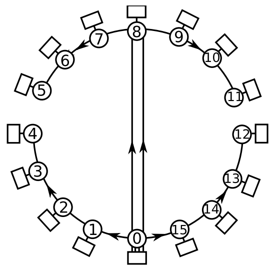

Figure 3. Broadcast in the Quarc NoC.

Bits:	[33:31]	[30:14]	[13:8]	[7:2]	[1:0]
Unicast Header:	0		destination addr.	source addr.	0
Multicast Header:	1	bitstring	destination addr.	source addr.	0
Broadcast Header:	2		destination addr.	source addr.	0
Body:	Payload				1
Tail:	Payload				2

Figure 4. The components of a sample node in the architecture.

3.3.2 Broadcast routing

Broadcast in the Quarc is elegant and efficient: The Quarc NoC adopts a BRCP (Base Routing Conformed Path) [2] approach to perform multicast/broadcast communications. BRCP is a type of path-based routing in which the collective communication operations follow the same route as unicasts do. Since the base routing algorithm in the Quarc NoC is deadlock-free, adopting BRCP technique ensures that the broadcast operation, regardless of the number of concurrent broadcast operations, is also deadlock-free.

To perform a broadcast communication the transceiver of the initiating node has to broadcast the packet on each ports of the multi-port router. The transceiver tags the header flit of each of four packets destined to serve each branch as broadcast to distinguish it from other types of traffic. The transceiver also sets the destination address of each packet as the address of the last node that the flits stream may traverse according to the base routing. The receiving nodes simply check if the destination address at the header flit matches its local address. If so, the packet is received by the local node. Otherwise, if the header flit of the packet is tagged as broadcast, the flits of the packet at the same time are received by the local node and forwarded along the rim. This is simply achieved by setting a flag on the ingress multiplexer which causes it to clone the flits.

The broadcast in a Quarc NoC of size 16 is depicted in Fig. 3. Assuming that Node 0 initiates a broadcast, it tags the header flits of each stream as broadcast and sets the destination address of packets as 4, 5, 11 and 12 which are the address of the last node visited on left, cross-left, cross-right and right rims respectively. The intermediate nodes receive and forward the broadcast flit streams, while the destination node absorbs the stream.

3.3.3 Multicast routing

Similar to broadcast, in multicast operation, the last node to be visited must be specified as destination address in the header flit. For broadcast all nodes in the path from source to destination are the receiver nodes. While, in case of multicast the target addresses are specified in the *bitstring* field. Each bit in the *bitstring* represents a node which its hop-distance from the source node corresponds to position of the bit in the *bitstring*. Status of each bit indicates whether the visited node is a target of the multicast or not. Fig. 4 shows the format of the flits for unicast, multicast and broadcast transmissions.

4 Validation

To validate the analytical model we have developed a discrete event simulator of the Quarc NoC operating at flit level using OMNET++ [20]. The schematic of the components in each node is shown in Fig. 5.

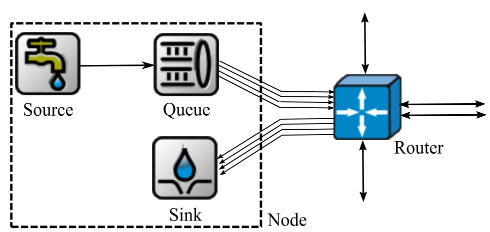

Figure 5. The components of a sample node in the architecture.

The source produces the messages according to a *Poisson* distribution. The passive queue has two queues to store the messages belonging to multicast and unicast traffic. The passive queue sends the messages based on their creation time. The passive queue is connected to the router through four injection channels. The router is connected to three neighboring routers, a sink and a passive queue. It receives the flits of the messages and sends them to the appropriate routers or its corresponding sink. The sink is connected to the router via four ejection channels and absorbs the messages destined for the node it belongs to.

Similar to the assumptions defined for the model, the resources are non-preemptive. While servicing a message, if other messages try to receive service the routers record their information. After the last flit of the current message

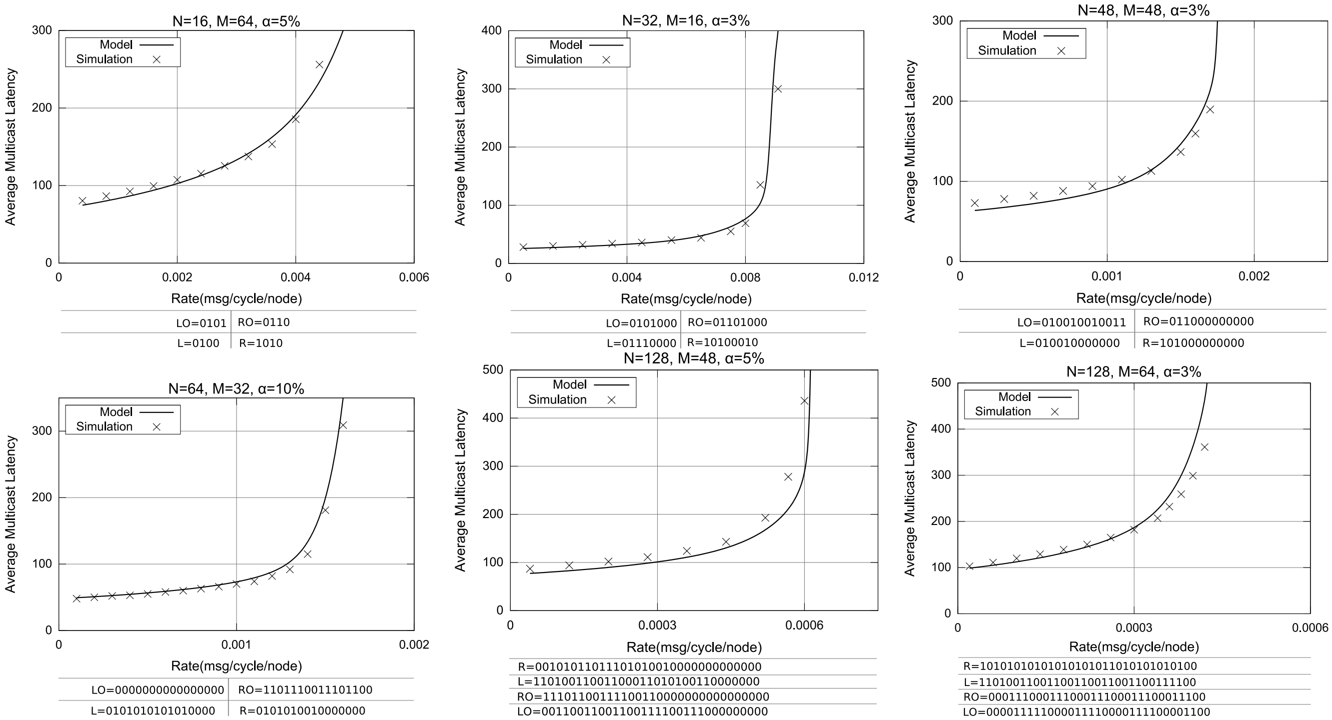

Figure 6. Comparison of the analytical model against the simulation results for random multicast destinations.

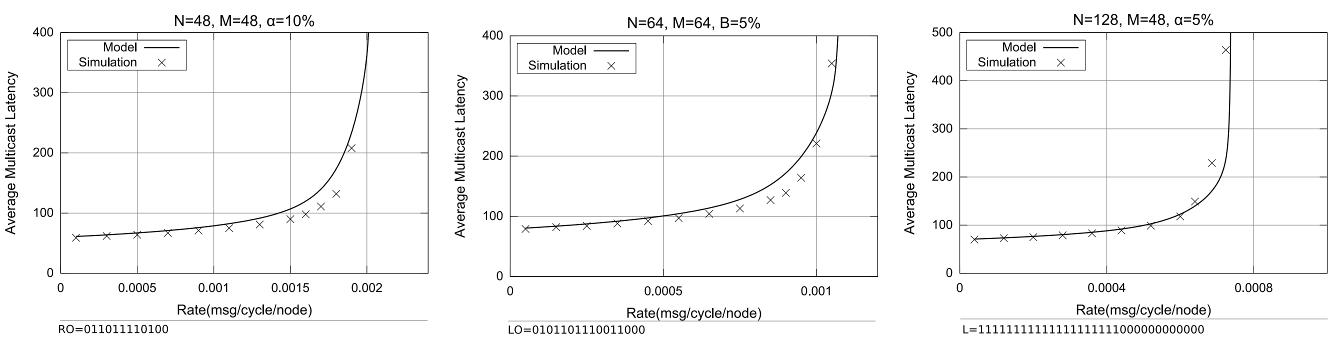

Figure 7. Comparison of the analytical model against the simulation results for localized destinations.

leaves the router, the router investigates the messages waiting on the recently released resource, serving them based on a FIFO policy.

Destinations of unicast messages at each node are selected randomly. The latency of a unicast message is regarded as the time from generation of unicast message at the source node until the time when the last flit of the message is absorbed by the sink at destination. The multicast destinations are selected randomly (by the authors) at the beginning of the simulation. Multicast message latency is the time from generation of the multicast message at the source node until the time when the last flit of the message is absorbed by the sink at the last destination.

The model predictions are compared against the simulation results for numerous configurations by changing the Quarc network size, message length and the rate of multicast traffic. Figures 6 and 7 compare the simulation results against the analysis for the networks ranging from 16 to 128 nodes. The message length may be16, 32, 48 or 64 flits size. And the multicast traffic may comprise 3%, 5% or 10% of the overall traffic. The graphs in Fig. 6 show the configurations in which the multicast destinations set are selected randomly. While, the graphs in Fig. 7 represent the configurations where the destination nodes are on the same rim. In other words the destination sets in graphs of Fig. 7 are localized.

In graphs, N, M and α represent the number of nodes in the Quarc NoC, message length and rate of multicast traffic respectively. While L, R, LO and RO denote the *bitstrings* corresponding to multicast destinations at left, right, cross-left and cross-right of the node respectively. The horizontal axis in the figures shows the message rate while the vertical axis describes the latency. As can be seen from the figures the analytical model presents an excellent approximation of the network latency in a wide range of configurations.

5 Conclusion

Analytical models of communication latency have been extensively reported for wormhole-routed interconnection networks. All these models, however, have assumed a uniform traffic pattern taking into account only unicast messages. This paper has introduced a novel analytical model to predict the average message latency of wormhole-routed multicast communication in direct interconnection networks adopting asynchronous multi-port routers. The analytical multicast model has been applied to the Quarc NoC, a highly efficient NoC for performing collective communications. and its validity has been verified by comparing the analytical results against the results obtained from a discrete-event simulator developed using OMNET++.

Our next objective is to investigate the validity of the model in other relevant interconnection networks such as multi-port mesh and torus.

References

[1] Dhabaleswar K. Panda , Sanjay Singal , and Ram Kesavan . Multidestination Message Passing in Wormhole k-ary n-cube Networks with Base Routing Conformed Paths. *IEEE Transactions on Parallel and Distributed Systems*, 1995.

[2] Dhabaleswar K. Panda , Sanjay Singal, and Pradeep Prabhakaran. Multidestination Message Passing Mechanism Conforming to Base Wormhole Routing Scheme. *Proceedings of the First International Workshop on Parallel Computer Routing and Communication*, 1994.

[3] Ju-Young Park. Construction of Optimal Multicast Trees Based on the Parameterized Communication Model. *Proceedings of International Conference on Parallel Processing*, 1996.

[4] Lu Zhonghai , Yin Bei , and A. Jantsch, . Connection-oriented multicasting in wormhole-switched networks on chip. *IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures*, 2006.

[5] William J. Dally and Charles L. Seitz . The torus routing chip. *Journal of Distributed Computing*, 1986.

[6] X. Lin , A.-H. Esfahanian , and A Burago. Adaptive Wormhole Routing in Hypercube Multicomputers. *Journal of Parallel and Distributed Computing*, pages 274–277, 1998.

[7] A. Shahrabi, M. Ould-Khoua, and L. Mackenzie. An analytical model of wormhole-routed hypercubes under broadcast traffic. *Journal of Performance Evaluation*, 2003.

[8] David F. Robinson, Dan Judd, Philip. K. McKinley, and Betty. H. C. Cheng. Efficient multicast in all-port wormhole-routed hypercubes. *Journal of Parallel and Distributed Computing*, 1995.

[9] Hong Xu, Ya-Dong Gui, and Lionel M. Ni . Optimal software multicast in wormhole-routed multistage networks. *IEEE Transactions on Parallel and Distributed Systems*, 1997.

[10] J. Duato, S. Yalamanchili, and L. Ni. *Interconnection networks: An Engineering Approach*. Morgan Kaufmann, 2003.

[11] J. Liu, L.-R. Zheng, and H. Tenhunen . Interconnect intellectual property for network-on-chip. *Journal of System Architectures*, 2003.

[12] Jeffrey T. Draper and Joydeep Ghosh. A comprehensive analytical model for wormhole routing in multicomputer systems. *Journal of Parallel and Distributed Computing*, 23(2):202–214, Nov. 1994.

[13] K. Goossens, J. Dielissen, and A. Radulescu. Aethereal network on chip: concepts, architectures, and implementations. *IEEE, Design and Test of Computers*, pages 414–421, 2005.

[14] L. Kleinrock. *Queueing Systems Volume I: Theory*. John Wiley and Sons, 1975.

[15] M. Coppola, R. Locatelli, G. Maruccia, L. Pieralisi, and A. Scandurra. Spidergon: a novel on-chip communication network. *Proceedings of International Symposium on System-on-Chip*, 2004.

[16] M. Moadeli, A. Shahrabi, W. Vanderbauwhede, and M. Ould-Khaoua. An analytical performance model for the Spidergon NoC. *21st IEEE International Conference on Advanced Information Networking and Applications*, pages 1014–1021, 2007.

[17] M. Moadeli, W. Vanderbauwhede, and A. Shahrabi. Quarc: A Novel Network 0n-Chip Architecture. *Parallel and Distributed Systems, International Conference on*, 2008.

[18] A. Shahrabi, L. M. Mackenzie, and M. Ould-Khaoua. A performance model of broadcast communication in wormhole-routed hypercubes. In *MASCOTS '00: Proceedings of the 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, page 98, Washington, DC, USA, 2000. IEEE Computer Society.

[19] STMicroelectronics. www.st.com.

[20] A. Varga. Omnet++. *IEEE Network Interactive, in the column Software Tools for Networking, www.omnetpp.org*, 16(4):683–689, 2002.

[21] W. J. Dally and B. Towles. Route packets, not wires: On-chip interconnection networks. *Proceedings of Design Automation Conference (DAC)*, pages 683–689, 2001.