An Analytical Performance Model for the Spidergon NoC

Mahmoud Moadeli[1], Ali Shahrabi[2], Wim Vanderbauwhede[1], Mohamed Ould-Khaoua[1]
1: Department of Computing Science
University of Glasgow
Glasgow, UK
Email: {mahmoudm, wim, mohamed}@dcs.gla.ac.uk
2 : School of Computing and Mathematical Sciences
Glasgow Caledonian University
Glasgow, UK
Email: a.shahrabi@gcal.ac.uk

Abstract

Networks on chip (NoC) emerged as a promising alternative to bus-based interconnect networks to handle the increasing communication requirements of the large systems on chip. Employing an appropriate topology for a NoC is of high importance mainly because it typically trade-offs between cross-cutting concerns such as performance and cost. The spidergon topology is a novel architecture which is proposed recently for NoC domain. The objective of the spidergon NoC has been addressing the need for a fixed and optimized topology to realize cost effective multi-processor SoC (MPSoC) development [7]. In this paper we analyze the traffic behavior in the spidergon scheme and present an analytical evaluation of the average message latency in the architecture. We prove the validity of the analysis by comparing the model against the results produced by a discrete-event simulator.

1. Introduction

Traditionally, interconnect architectures for integrated circuits have been bus-based. Driven by the advances in semiconductor technologies reaching sub-0.1um gate lengths, realization of the 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 [16] . The main driving factor behind employing NoC have been decoupling the communication fabric from the processing elements. NoCs help resolve the electrical problems in new deep sub-micron technologies, as they allow to structure and manage global wires. At the same time the wires are used more efficiently, requiring fewer wires. A NoC architecture also leads to lower power consumption and enhanced reliability [4]. As a result, NoCs have come to be regarded as the favored on-chip communication paradigm [10].

NoCs in principle are similar to the interconnection networks for parallel computers with multiple processors. NoCs, however have some peculiarities that distinguish them from the parallel computers. The major differences between two fields are energy constraints, design specialization and degree of heterogeneity of the employed components. Since most NoC applications run on the battery operated devices the energy customization and saving is of prime importance in the domain. Also, despite the network for parallel computers that are expected to be deployed for a wide range of unknown applications, the NoCs may be tailored for a particular application. And finally a variety of the components including DSP and FPGA may be distributed over a single NoC as well as the memories and processors.

Typically in the NoC domain the resources are scarce and the applications have some performance requirements. A major challenge in the domain is therefore, fulfilling the applications' performance demands using the limited available resources. Apparently, resource constraints lead to employing the algorithms and techniques that may realize the required functionality in a more cost effective fashion. Researches carried out in the field reveals that deterministic routing and wormhole switching [18] are the dominant rout-

21st International Conference on Advanced Networking and Applications(AINA'07)
0-7695-2846-5/07 $20.00 © 2007 IEEE

ing algorithm and switching technique for NoCs [15, 16].

Deterministic routing algorithms are usually adopted because they require implementing a simpler logic compared to their adaptive counterparts. Implementing the simpler logic in turn leads to realization of smaller routers using less resources.

Employing the wormhole switching originates from the limitation in the buffer size at intermediate routers [1, 16]. Despite the *packet switching* and *virtual cut-through* switching techniques that require enough buffer for at least a whole packet at each intermediate router, wormhole switching implements the functionality with much lower buffer requirement. In wormhole switching a packet is divided into elementary units called *flits,* each composed of a few bytes for transmission and flow control. The header flit governs the route and the remaining data flits follow it in a pipelined fashion. If the header flit blocks, the remaining flits are blocked in situ. Since the packet are not required to reside in each intermediate router in whole and also routers may be implemented by providing buffer for as low as one flit per channel, the wormhole switching can realize low message latency at a low cost.

Another factor that has a significant impact on the size of the routers is the degree of the nodes in the network. The degree of a node determines the number of neighboring router that a particular router is directly connected to. Increasing the node degree can significantly improve the performance of the network. Obviously, realization of routers with a higher node degree typically leads to the difficulties of dealing with larger routers and also more complicated wiring [10] .

In this paper we analyze the traffic behavior in the spidergon NoC and present an analytical model for evaluating the average message latency in the architecture adopting the wormhole switching. To the best of our knowledge this work presents the first analytical model for the spidergon scheme.

The rest of the paper proceeds as follows. In the next section we discuss the NoC topologies with special emphasis on the spidergon NoC. In section 3 we have presented a method for analyzing the latency in networks with wormhole routing. Section 4 presents a detailed analysis of the traffic in the spidergon NoC and also adopts the method introduced in section 3 to evaluate the average latency for the spidergon topology. Section 5 will compare our analytical evaluation with simulation results and finally, in section 6 the conclusion and future works have been presented.

2. The Spidergon NoC

The topology of an on-chip network specifies the structure in which routers connect the IPs together. A NoC may have any $2-$Dimensional topologies that have been proposed for interconnection networks. Some of the architectures introduced or adopted for the NoC domain are fat tree [9], butterfly-fat tree [11], mesh [13], torus [16], folded torus [17] and variations of the ring in octagon [2] and the spidergon scheme [7]. Typically, a particular topology is selected in order to trade-off between a number of cross-cutting measures such as performance and cost. A number of important characteristics that affect adopting a particular topology are network diameter, the highest degree of nodes in the network, regularity, scalability and synthesis cost for an architecture.

In [10] Pande et al. have compared different on-chip network topologies. The paper compares different on-chip network architectures from different points of view including latency, throughput, area and energy consumption. To conduct the experiments, they adopted a simulator employing flit-level event-driven wormhole-switching to study the characteristics of the communication-centric parameters of the interconnect infrastructures. In a more recent research Bononi and Concer [5] compared ring, mesh and spidergon topologies using a discrete-event simulator. Their research revealed that in general the spidergon NoC outperforms the irregular mesh and the ring topologies.

The objective of the spidergon topology has been addressing the demand for a fixed and optimized topology to realize low cost multi-processor SoC implementation. In the spidergon topology an even number of nodes are connected by unidirectional links to the neighboring nodes in clockwise and counter-clockwise directions plus a cross connection for each pair of nodes. In the spidergon scheme each physical link is shared by two virtual channels in order to avoid deadlock. Figure 1 depicts a spidergon topology of size 16 and its layout on a chip.

The key characteristics of the spidergon 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.

In the spidergon topology nodes are connected by unidirectional links. Let the number of nodes be an even $N = 2n$. Every node in the network is indexed by a number between 0 to $N - 1$. 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_j for $j = (i + 1) \, mod \, N$, $j = (i - 1) \, mod \, N$ and $j = (i + \frac{N}{2}) \, mod \, N$.

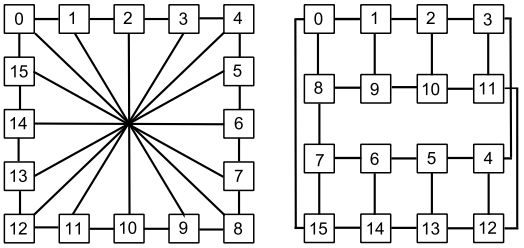

Figure 1. The spidergon topology and the on chip layout.

3. The Analysis Method

The objective of this section is to introduce the model adopted to evaluate the average message latency 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 model we use to analyze our network was first introduced by Draper and Ghosh [3] to evaluate the average latency for k-ary n-cubes. The model was later adopted and generalized for evaluating different interconnection network topologies such as mesh [12]. In this section we explain the model and in section 4 we apply it to analyze the average latency for a network with the spidergon topology.

The model uses some assumptions that are widely used in the literature [14, 8, 12, 3].

- Nodes are generating the messages independently and according to a Poisson process.
- Destination addresses are selected randomly.
- Regardless of the blocking, 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 adopted routing is a deterministic, shortest path routing algorithm. In situations where the number of the intermediate routers are equal if the surrounding channels or cross-network channels are traversed, the surrounding channels are selected.
- The network employs wormhole switching.

We view our network 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 [6]

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

$$\rho = \frac{\lambda}{x} \quad (2)$$

where λ is the mean arrival rate, x is the mean service time and σ^2 is the variance of the service time distribution. In calculating the arrival rate we assume that the mean departure rate of a channel is equal to the mean Poisson arrival rate at the channel provided that the channel is stable ($\rho <$ 1).

During the journey toward a destination, a message passes through a number of channels. Since in wormhole routing flits follow the header flit, the waiting time needs to be evaluated only for the header flit. The time that the header flit spends in each channel is comprised of two components, the waiting time for the next channel and one time unit to actually cross the channel. After the header flit is granted access to the ejection channel at the destination node the network requires an extra $\overline{msg}$ time units to pipeline the message to destination. Here, $\overline{msg}$ is the average size of the messages in flits. Therefore, the latency may be defined as the total waiting and service times the header flit experiences at each intermediate channel plus $\overline{msg}$ cycles to complete transmission. Thus, for an arbitrary node j in the network latency may be expressed as

$$\overline{L}_j = w_{inj,j} + \overline{x}_{inj,j} + \overline{D} - 1 \quad (3)$$

where $w_{inj,j}$ and $\overline{x}_{inj,j}$ are the average waiting and service time at injection channel and $\overline{D}$ is the average distance in terms of the number of channels traversed.

Averaging on all nodes in the network yields the average latency for the network.

$$L = \frac{1}{N}\sum_j \overline{L}_j \quad (4)$$

$$= \frac{1}{N}\sum_j (w_{inj,j} + \overline{x}_{inj,j}) + \overline{D} - 1$$

As we modeled each channel as an M/G/1 queue, $w_{inj,j}$ will be derived once the mean service time and its variance is known.

Since in wormhole routing a message typically spans several channels at each time, the service time of each channel depends on the waiting and service time of its subsequent channels. Therefore, to analyze the service time of each channel the waiting and service times of all possible successive channels are required. Figure 2 can aid in analyzing the service time for an arbitrary channel i.

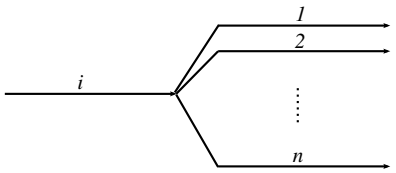

Figure 2. Service time at channel i depends on n successive channels.

Figure 2 illustrates that the traffic leaving channel i, may be injected to any of its n subsequent channels, where n is a non negative integer. We denote by $P_{i\rightarrow j}$ the probability that the traffic enters the channel j after leaving channel i. The average service time experienced at channel i therefore can be expressed as:

$$\overline{x}_i^{in} = \sum_j (w_j + \overline{x}_j).P_{i\rightarrow j} \tag{5}$$

This equation states that the service time at each channel is derived once the waiting time and service time of its successive channels are known. The mean waiting time, w_j , may be approximated using an M/G/1 queuing model with service time x_j.

Draper and Ghosh [3] suggested that waiting time for an M/G/1 queuing system which is used for modeling wormhole routing may be approximated by

$$W_j = \frac{\lambda_j \overline{x}_j^2}{2(1-\lambda_j \overline{x}_j)}(1 + \frac{(\overline{x}_j - \overline{msg})^2}{\overline{x}_j^2}) \tag{6}$$

The above waiting time, W_j, is the mean waiting time for a server in which messages arrive at rate λ and no message may block other messages. In wormhole routing, however a message may block other messages and therefore W_j will not equal to w_j. In fact when a message has occupied a channel there is not any competing traffic from this channel for the subsequent channels. And the traffic on the channel competes only with the incoming traffic from other channels. Therefore, the mean w_j is less than mean W_j. The situation is addressed by introducing a blocking probability which is defined as

$$P_{bl} i \rightarrow j = 1 - \frac{\lambda_i^{in}}{\lambda_j} P_{i\rightarrow j} \tag{7}$$

where λ_i^{in} is the incoming traffic on c_j from c_i, λ_j is the total traffic rate on c_j and $P_{i\rightarrow j}$ is the probability that c_j is traversed after leaving c_i.

Finally, w_j is obtained as the product of the W_j and $P_{bl} i \rightarrow j$

$$w_j = W_j.P_{bl} i \rightarrow j \tag{8}$$

By combining equations 5, 7 and 8 we obtain the service time for an intermediate channel as

$$\overline{x}_i^{in} = \sum_j ((1 - \frac{\lambda_i^{in}}{\lambda_j} P_{i\rightarrow j}) W_j + \overline{x}_j) P_{i\rightarrow j} \tag{9}$$

where W_j is approximated by equation 6 using the $\overline{x}_j$ as mean service time.

We now apply this method to analyze the average latency for the spidergon topology.

4. Analysis of the Spidergon Architecture

As mentioned earlier the spidergon scheme avoids deadlock by employing virtual channels. However, since one of the virtual channels in the links is only behaving as the escape channel, the traffic passing through it is in negligible compared to the total traffic passing the link. Therefore, in our evaluation we consider the total traffic on each physical links rather than traffic on each individual virtual channel.

To simplify their model Draper and Ghosh [3] introduced the concept of an equivalence class. An equivalence class is defined as a set of channels with similar stochastic properties with respect to the arrival and service rate. By assigning each channel to an equivalence class, the network evaluation will only require manipulation of equivalence classes. Otherwise, all channels had to be taken into account explicitly for analysis.

In the spidergon topology because of the symmetry all channels around the network present the same stochastic properties so they belong to the same equivalence class. Channels inside the network also have the same traffic and service rate and are accounted as a single equivalence class. Although in the steady state, the injection and ejection channels have the same arrival rates, their service rates differs due to blocking property of wormhole routing. Therefore, injection and ejection channels are considered as two different equivalence classes.

In the rest of this section we first discuss the traffic on equivalence classes and interaction between them and later obtain the average latency for our network.

4.1. Traffic Analysis

To evaluate the service and waiting time at each channel the detailed traffic information is required. Traffic on

each channel is comprised of several incoming streams and will be transmitted to a number of successive channels. In this section the traffic on each equivalence class has been provided. As the traffic distribution is slightly different depending on whether the number of nodes is a factor of four ($N = 4x$) or only a factor of two ($N = 4x + 2$) we have presented them separately when required.

We denote by λ the traffic rate at which a node sends to each individual destination. In steady state the mean arrival rate to each channel equals to the mean departure from the channel. Therefore, the ejection channels have the same traffic rates as injection channels.

$$\lambda_{inj} = \lambda_{ej} = (N-1).\lambda \tag{10}$$

The traffic on each surrounding channel is comprised of the traffic from three sources, i) cross network link, ii) previous link and iii) injection link. The incoming traffic rate from the cross link is denoted by $\lambda_{cross \to surr}$ and equals

$$\lambda_{cross \to surr} = \begin{cases} (\lceil \frac{N}{4} \rceil - 1).\lambda & N = 4x \\ (\lfloor \frac{N}{4} \rfloor - 1).\lambda & N = 4x+2 \end{cases} \tag{11}$$

And the message rate from the injection channel, $\lambda_{inj \to surr}$ is

$$\lambda_{inj \to surr} = \left\lceil \frac{N}{4} \right\rceil .\lambda \tag{12}$$

Finally the contribution of the previous surrounding link, $\lambda_{surr \to surr}$ equals

$$\lambda_{surr \to surr} = \begin{cases} (\lceil \frac{N}{4} \rceil - 1)^2.\lambda & N = 4x \\ (\lfloor \frac{N}{4} \rfloor^2 - \lfloor \frac{N}{4} \rfloor + 1).\lambda & N = 4x+2 \end{cases} \tag{13}$$

Adding the above values together yields

$$\lambda_{surr} = \begin{cases} \lceil \frac{N}{4} \rceil^2.\lambda & N = 4x \\ (\lfloor \frac{N}{4} \rfloor^2 + \lfloor \frac{N}{4} \rfloor + 1).\lambda & N = 4x+2 \end{cases} \tag{14}$$

The traffic on a channel around the network has two targets, next channel and ejection channel. The rate of the traffic transmitted to the next surrounding channels equal to the traffic transmitted from the previous channel. And the remaining traffic which is absorbed by ejection channel is

$$\lambda_{surr \to ej} = (\frac{N}{2} - 1).\lambda \tag{15}$$

And finally, the traffic on each channel inside the network is

$$\lambda_{cross} = \begin{cases} (2\lceil \frac{N}{4} \rceil - 1).\lambda & N = 4x \\ (2\lfloor \frac{N}{4} \rfloor - 1).\lambda & N = 4x+2 \end{cases} \tag{16}$$

4.2. Average Message Latency

As the spidergon NoC is a symmetric topology, nodes have similar stochastic properties. Thus, finding the average latency is limited to analysis of only one node. Also, having just a small number of equivalence classes further reduces the complexity.

After leaving the injection channel a message traverse left or right links with similar probability equal to

$$P_{inj \to right} = P_{inj \to left} = \frac{\lceil \frac{N}{4} \rceil}{N-1} \tag{17}$$

And the probability that the message enters the cross link will be

$$P_{inj \to cross} = \frac{(2\lfloor \frac{N}{4} \rfloor - 1)}{N-1} \tag{18}$$

Therefore, the service time of the injection channel can be expressed as

$$\begin{aligned} \overline{x}_{inj} \quad = \quad & (w_{right} + \overline{x}_{right}) P_{inj \to right} + \\ & (w_{left} + \overline{x}_{left}) P_{inj \to left} + \\ & (w_{cross} + \overline{x}_{cross}) P_{inj \to cross} \end{aligned} \tag{19}$$

A message entering the left or right links may traverse up to $\lceil \frac{N}{4} \rceil$ nodes. Each intermediate node that receives the message checks its destination address. If the node is the target it absorbs the message, otherwise the message is forwarded to the next channel in the same direction. This scenario is depicted in figure 3.

The service time at each intermediate node i may be expressed as

$$x_i^{surr} = x_{ej}.\frac{1}{i} + (w_{i-1}^{surr} + x_{i-1}^{surr})\frac{i-1}{i} \tag{20}$$

where, w_i^{surr} and x_i^{surr} are the waiting time and service time at channel i, and x_{ej} is the service time at the ejection channel which equals to $\overline{msgl}$.

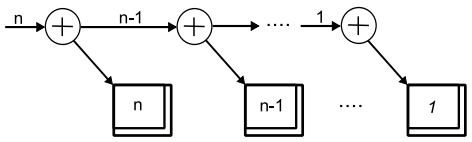

Figure 3. Possible destinations for a message on a surrounding channel.

By adopting the above equation,

$$\overline{x}_{right} = \overline{x}_{left} = x^{surr}_{\lceil \frac{N}{4} \rceil} \tag{21}$$

A message traversing an internal link after leaving the injection channel may be destined for the cross network node or it can enter the left or right channels with the same probability.

$$\begin{aligned} \overline{x}_{cross} = x_{ej}.\frac{1}{2\left\lfloor \frac{N}{4} \right\rfloor - 1} + \\ (w_{crossright} + \overline{x}_{crossright})\frac{2(\left\lfloor \frac{N}{4} \right\rfloor - 1)}{2\left\lfloor \frac{N}{4} \right\rfloor - 1} + \\ (w_{crossleft} + \overline{x}_{crossleft})\frac{2(\left\lfloor \frac{N}{4} \right\rfloor - 1)}{2\left\lfloor \frac{N}{4} \right\rfloor - 1} \end{aligned} \tag{22}$$

Again $\overline{x}_{crossright}$ and $\overline{x}_{crossleft}$ which are the service time experienced at the right or left channels at the opposite side may be evaluated using equation 20. The maximum number of hops a message may take in any direction at cross network is $\left\lfloor \frac{N}{4} \right\rfloor - 1$. Therefore, we have

$$\overline{x}_{crossright} = \overline{x}_{crossleft} = x^{surr}_{\left\lfloor \frac{N}{4} \right\rfloor - 1} \tag{23}$$

By putting all pieces together, the average latency experienced in the spidergon architecture will be

$$\overline{L} = w_{inj} + \overline{x}_{inj} + \overline{D} - 1 \tag{24}$$

where $\overline{D}$ is

$$\overline{D} = \begin{cases} (2\left\lceil \frac{N}{4} \right\rceil^2 + 4\left\lceil \frac{N}{4} \right\rceil + 1)/N & N = 4x \\ (2\left\lfloor \frac{N}{4} \right\rfloor^2 + 2\left\lfloor \frac{N}{4} \right\rfloor - 1)/N & N = 4x + 2 \end{cases} \tag{25}$$

5. Validation

To validate the analytical model we have developed a discrete event simulator operating at flit level. Each simulation experiment is run until the network reaches its steady state, i.e. until a further increase in simulated network cycles doe not change the collected statistics appreciably. Statistics gathering was inhibited for the first 20000 messages to avoid distortions due to start-up transient. The simulator operates on the same assumption as the analysis. Some of the assumptions are mentioned here. A network cycle is defined as the time required that a flit traverse between two adjacent router or between a router and an IP. The time consumed in the routers is also ignored in simulation. Messages are generated at each node according to a Poisson process. Also all messages are assumed to be of equal size.

Destinations at each node are selected randomly and the traffic is uniform. The latency for a message is considered as the time a message is created in the source to the time when the last flit of the message is absorbed by the destination IP.

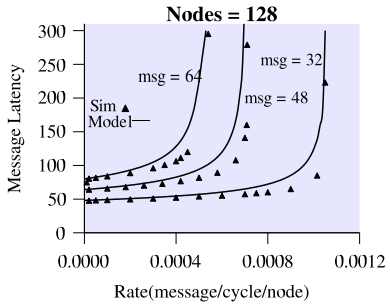

Figure 5. The analytical evaluation against the simulation for the spidergon NoCs of size 128 **for message lengths** $32, 48$ **and** 64 **flits.**

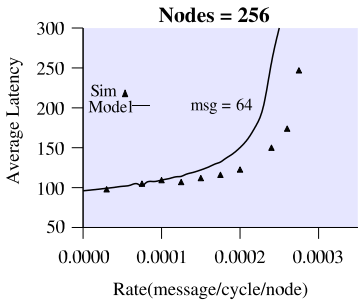

Figure 6. The analytical evaluation against the simulation for the spidergon NoCs of size 256 **for message length of** 64 **flits.**

The model is compared against the simulation results for

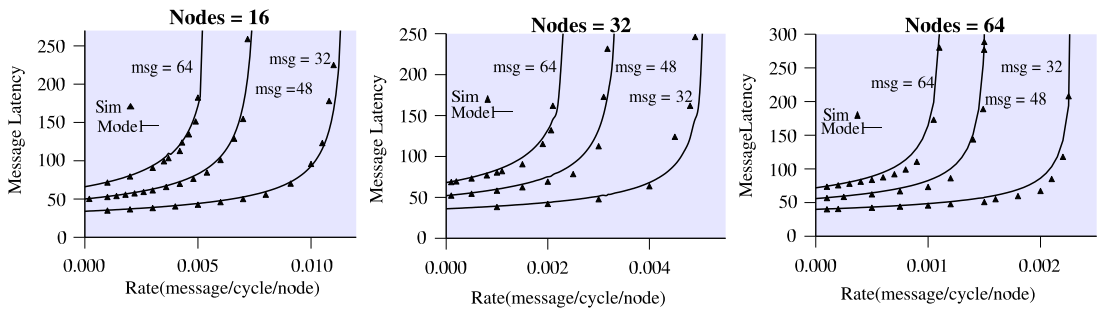

Figure 4. The analytical evaluation against the simulation for the spidergon NoCs of size $16, 32$ **and** 64 **for message lengths** $32, 48$ **and** 64 **flits.**

numerous configurations by changing the message length and also the network size. Figure 4 and 5 compare the simulation results against the analysis for the networks of size $16, 32, 64$ and 128 when the length of the message is set to $32, 48$ and 64. While figure 6 shows the results obtained by simulation against the analytical values in the network of size 256 when the network exchanges the messages of length 64.

The horizontal axis in the figures shows the message rate while the vertical axis describes the latency. As can be seen from the the figures the analytical model presents a good approximation of the network latency in the presence of the light and heavy traffics. In particular the figures reveal that the model well predicts the network saturation points.

6. Conclusion

The spidergon topology emerged to realize cost effective MPSoC development using a fixed and optimized NoC architecture. In this paper we have analyzed the traffic in the architecture and presented a model to compute the mean message latency in the spidergon architecture employing wormhole switching. Extensive simulation experiments have shown the analytical model predicts the message latency with a good degree of accuracy in a wide range of traffic rates. In particular the model well predicts the saturation points in the networks with different configurations.

Our next objective is to investigate the impact of employing more virtual channels and adopting adaptive routing on the latency in the spidergon scheme. Developing a cost model to study the effect of the channel length on evaluations as the size of the network grows is another goal we pursue. Moreover, we are going to analytically compare the spidergon NoC with other topologies in the domain including mesh and torus.

References

[1] E. Bolotin, et. al. QoS architecture and design process for Networks-on- Chip. *Journal of Systems Arch*, 2004.

[2] F. Karim et al. An Interconnection Architecture for Networking Systems on Chip. *IEEE Microprocessors*, 22(5):36–45, Sept. 2002.

[3] 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.

[4] L. Benini and G. D. Michelli. Powering Networks on Chips Energy-efficient and reliable interconnect design for SoCs. *International Symposium on Systems Synthesis*, pages 33–38, 2001.

[5] L. Bononi and N. Concer. Simulation and Analysis of Network on Chip Architectures: Ring, Spidergon and 2D Mesh. *DATE*, pages 154–159, 2006.

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

[7] 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.

[8] M Ould-Khaoua. Message latency in the 2-dimensional mesh with wormhole routing. *Microprocessors and Microsystems*, 1999.

[9] A. G. P. Guerriert. A generic architecture for on-chip packet-switched interconnections. *Proceedings of Design Automation Conference (DAC)*, pages 683–689, 2001.

[10] Partha Pratim Pande, Cristian Grecu, Michael Jones, Andre Ivanov, and Resve Saleh. Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures. *IEEE Transactions on Computers*, 54(8):1025–1040, Aug. 2005.

[11] P.P. Pande, C. Grecu, A. Ivanov, and R. Saleh. Design of Switch for Network on Chip Applications. *Proceedings of Int'l Symposium on Circuit and Systems (ISCAS)*, 5:217–220, May 2003.

[12] R Greenberg and L. Guan. Modelling and Comparison of Wormhole Routed Mesh and Torus. *Ninth IASTED intel*, 1997.

[13] S. Kumar et al. A network on chip architecture and design methodology. *Proceedings of Int't Symp. VLSI (ISVLSI)*, pages 117–124, 2002.

[14] S. Loucif, M. Ould-Khaoua, and L. M. Mackenzie. Analysis of fully-adaptive routing in wormhole-routed tori. *Parallel Computing*, 1999.

[15] Umit Y. Ogras, Jingcao Hu, and Radu Marculescu. Key Research Problems in NoC Design: A Holistic Perspective. *International Conference on Hardware - Software Codesign and System Synthesis*, 1987.

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

[17] W. J. Dally and C.L. Seitz. The Torus Routing Chip. Technical report, Technical Report 5208:TR: 86, Computer Science Dept. California Inst. of Technology, 1-19, 1986.

[18] W. J.Dally and C. L. Seitz. Deadlock-free Message Routing in Multiprocesor Interconnection Networks. *IEEE Transactions on Computers*, 1987.