Max-Min-Fair Best Effort Flow Control in Network-on-Chip Architectures

Fahimeh Jafari^{1,2}, Mohammad H. Yaghmaee¹, Mohammad S. Talebi²,
and Ahmad Khonsari^{3,2}

¹ Ferdowsi University of Mashhad, Mashhad, Iran

² IPM, School of Computer, Tehran, Iran

³ ECE Department, University of Tehran, Tehran, Iran

{jafari,ak}@ipm.ir, hyaghmaee@ferdowsi.um.ac.ir,
mstalebi@gmail.com

Abstract. *Network-on-Chip* (NoC) has been proposed as an attractive alternative to traditional dedicated busses in order to achieve modularity and high performance in the future *System-on-Chip* (SoC) designs. Recently, end to end flow control has gained popularity in the design process of network-on-chip based SoCs. Where flow control is employed, fairness issues need to be considered as well. In fact, one of most difficult aspects of flow control is that of treating all sources fairly when it is necessary to turn traffic away from the network. In this paper, we proposed a flow control scheme which admits Max-Min fairness criterion for all sources. In fact, we formulated Max-Min fairness criterion for the NoC architecture and presented implementation to be used as flow control mechanism.

Keywords: Network-on-Chip, flow control, Max-Min fairness.

1 Introduction

Network-on-Chip (NoC) is a new paradigm structure for designing future *System-on-Chips* (SoC) [1]. A typical NoC architecture provides a scalable communication infrastructure for interconnecting cores. Since the communication infrastructure as well as the cores from one design can be easily reused for a new product, NoC provides maximum possibility for reusability.

NoCs with their flexible and scalable interconnect provide high computational power to support computationally extensive multimedia applications, i.e. those that combine audio, video and data. In contrast to simple data applications, which can work without guarantees of timing of data delivery, multimedia applications require a guaranteed degree of service in terms of required bandwidth and timelines. According to the networking terminology, we refer to the traffic of simple data as elastic or *Best Effort* (BE) traffic and to multimedia traffic as inelastic or *Guaranteed Service* (GS) traffic.

Due to the rapid growth of the number of *processing elements* (PEs) in NoCs [2], employing efficient policy for flow control is inevitable in the design of NoCs to

provide the required *Quality of Service* (QoS). A NoC should support network level flow control in order to avoid congestion in the bottleneck links, i.e. link through which several sources pass [3]. The design and control of NoCs raises several issues well suited to study using techniques of operational research such as optimization and stochastic modeling. Recently, some novel researches have been embarked in studying congestion control in NoCs [4-5]. Congestion control schemes in NoCs mainly focus on utilizing NoC's resources, with the aim of minimizing network cost or maximizing network utility while maintaining the required QoS for Guaranteed Service traffics.

Many strategies for flow control have been proposed for off-chip networks, e.g. data networks, etc. [6-9]. On-chip networks pose different challenges. For instance, in off-chip environments, to overcome congestion in links, packet dropping is allowed. On the contrary, reliability of on-chip wires makes NoCs a loss-less environment.

So far, several works have addressed this problem for NoC systems. In [4], a prediction-based flow-control strategy for on-off traffic in on-chip networks is proposed where the prediction is used in router to be aware of buffer fillings. In [5] a flow-control scheme for Best Effort traffic based on Model Predictive Control is presented, in which link utilization is used as congestion measure. Dyad [10] controls the congestion by switching from deterministic to adaptive routing when system is going to be congested. [11] proposes a flow control scheme as the solution to rate-sum maximization problem for choosing the BE source rates. The solution to the rate-sum optimization problem is presented as a flow control algorithm.

Where flow control is employed, fairness issues need to be considered as well [3]. In fact, one of most difficult aspects of flow control is to choose a policy to accommodate a fair rate allocation. All of the abovementioned studies only regarded the flow control by taking into account the constraints of the system and to the best of our knowledge no policy to maintain fairness among sources was chosen.

The fairness of TCP-based flow control algorithms was first analyzed in [12]. The analysis in [12] was based on a single bottleneck link. Different flow control approaches can be classified with respect to the fairness criteria, in favor of which rate allocation is done. One of the famous forms of fairness criterion is Max-Min fairness, which has been discussed in earlier literature and described clearly in [13]. Our main contribution in this paper is to present a flow control scheme for Best Effort traffic in NoC which satisfies Max-Min fairness criterion. Our framework is mainly adopted from the seminal work [13] which presents a basic Max-Min fairness optimization problem. In this paper, we reformulate such a problem for the NoC architecture.

The organization of the paper is as follows. In Section 2 we present the system model, the concept of Max-Min fairness and formulation of the flow control as an optimization problem. In section 3 we present an iterative algorithm as the solution to the flow control optimization problem. Section 4 presents the simulation results and discussion about them. Finally, the section 5 concludes the paper and states some future work directions.

2 System Model

We consider a NoC with two dimensional mesh topology, a set S of sources and a set L of bidirectional links. Let c_l be the finite capacity of link $l \in L$. The NoC assumed to use wormhole routing. In wormhole-routed networks, each packet is divided into a sequence of *flits* which are transmitted over physical links one by one in a pipeline fashion. The NoC architecture is also assumed to be lossless, and packets traverse the network on a shortest path using a deadlock free XY routing. A source consists of Processing Elements (PEs), routers and Input/Output ports. Each link is a set of wires, busses and channels that are responsible for connecting different parts of the NoC. We denote the set of sources that share link l by $S(l)$. Similarly, the set of links that source s passes through is denoted by $L(s)$. By definition, $l \in S(l)$ if and only if $s \in L(s)$.

We assume that there are two types of traffic in the NoC: GS and BE traffic. For notational convenience, we divide S into two parts, each one representing sources with the same kind of traffic. In this respect, we denote the set of sources with BE and GS traffic by S_{BE} and S_{GS} , respectively. Each link l is shared between the two aforementioned traffics. GS sources will obtain the required amount of the capacity of links and the remainder should be allocated to BE sources.

2.1 Max-Min Fairness Concept

Any discussion of the performance of a rate allocation scheme must address the issue of fairness, since there exist situations where a given scheme might maximize network throughput, for example, while denying access for some users or sources. Max-Min fairness is one the significant fairness criteria. Crudely speaking, a set of rates is max-min fair if no rate can be increased without simultaneously decreasing another rate which is already smaller. In a network with a single bottleneck link, max-min fairness simply means that flows passing through the bottleneck link would have equal rates.

The following definition states the formal definition of Max-Min fairness.

Definition 1. A feasible rate allocation $x = (x_s, s \in S)$ is said to be “max-min fair” if and only if an increase of any rate within the domain of feasible allocations must be at the cost of a decrease of some already smaller rate. Formally, for any other feasible allocation y , if $y_s > x_s$ then there must exist some s' such that $x_{s'} \leq x_s$ and $y_{s'} < x_{s'}$ [13].

Depending on the network topology, a max-min fair allocation may or may not exist. However, if it exists, it is unique (see [14] for proof).

In what follows the condition under which the Max-Min rate allocation exists will be stated. Before we proceed to this condition, we define the concept of bottleneck link.

Definition 2. With our system model above, we say that link l is a bottleneck for source s if and only if

1. link l is saturated: $\sum_{s \in S_{BE}(l)} x_s + \sum_{s \in S_{GS}(l)} x_s = c_l$
2. source s on link l has the maximum rate among all sources using link l .

Intuitively, a bottleneck link for source s is a link which limits x_s .

Theorem 1. A max-min fair rate allocation exists if and only if every source has a bottleneck link (see [14] for proof).

2.2 Flow Control Model

Our focus will be on two objectives. First, choosing source rates (IP loads) of BE traffics so that to accomplish flow control in response to demands at a reasonable level. Second, maintaining Max-Min fairness for all sources. We model the flow control problem in NoC as the solution to an optimization problem. For more convenience, we turn the aforementioned NoC architecture into a mathematical model as in [5]. In this respect, the Max-Min fairness flow control problem can be formulated as:

$$\max_x \min_{s \in S} x_s \quad (1)$$

subject to:

$$\sum_{s \in S_{BE}(l)} x_s + \sum_{s \in S_{GS}(l)} x_s \leq c_l \quad \forall l \in L \quad (2)$$

$$x_s > 0 \quad \forall s \in S_{BE} \quad (3)$$

where source rates, i.e. x_s , $s \in S$, are optimization variables.

The constraint (2) says the aggregate BE source rates passing thorough link l cannot exceed its free capacity, i.e. the portion of the link capacity which has not been allocated to GS sources. For notational convenience, we define

$$u = \min_{s \in S} x_s$$

$$\hat{c}_l = c_l - \sum_{s \in S_{GS}(l)} x_s,$$

therefore the above mentioned problem can be rewritten as:

$$u = \min_{s \in S} x_s \quad (4)$$

$$\max u \quad (5)$$

subject to:

$$\sum_{s \in S_{BE}(l)} x_s \leq \hat{c}_l \quad \forall l \in L \quad (6)$$

$$x_s > 0 \quad \forall s \in S_{BE} \quad (7)$$

To solve the above problem, it should be converted so as to be in the form of disciplined optimization problems [15] as follows:

$$\max u \quad (8)$$

subject to:

$$u \leq x_s \quad \forall s \in S \quad (9)$$

$$\sum_{s \in S_{BE}(l)} x_s \leq \hat{c}_l \quad \forall l \in L \quad (10)$$

$$x_s > 0 \quad \forall s \in S_{BE} \quad (11)$$

The above optimization problem can be solved using several methods. In the next section, we introduce a simple and famous algorithm, known as “progressive filling”, to solve (8) iteratively.

In order to compare the results of progressive filling algorithm with the exact values, we solve problem (8) using CVX [16] which is a MATLAB-based software for disciplined convex optimization problems, whose results will be given in section 4.

3 Max-Min Fairness Algorithm

Theorem 1 is particularly useful in deriving a practical method for obtaining a max-min fair allocation, called “progressive filling”. The idea is as follows: rates of all flows are increased at the same pace, until one or more links are saturated. The rates of flows passing through saturated links are then frozen, and the other flows continue to increase rates. All the sources that are frozen have a bottleneck link. This is because they use a saturated link, and all other sources using the saturated link are frozen at the same time, or were frozen before, thus have a smaller or equal rate. The process is repeated until all rates are frozen. Lastly, when the process terminates, all sources have been frozen at some time and thus have a bottleneck link. Using Theorem 1, the allocation is max-min fair.

Theorem 2. For the system model defined above, with fixed routing policy, there exists a unique max-min fair allocation. It can be obtained by the progressive filling algorithm. (see [14] for proof)

In the sequel, we derive the max-min rate allocation as the solution to problem (8) and based on this algorithmic solution, we present a flow control scheme for BE traffic in NoC systems.

Thus, the aforementioned algorithm can be employed to control the flow of BE traffic in the NoC. The iterative algorithm can be addressed in distributed scenario. However, due to well-formed structure of the NoC, we focus on a centralized scheme; we use a controller like [5] to be mounted in the NoC to implement the above algorithm. The necessary requirement of such a controller is the ability to accommodate simple mathematical operations and the allocation of few wires to communicate flow control information to nodes with a light GS load.

Algorithm 1. Max-Min Fair (MMF) Flow Control Algorithm for BE in NoC.

Initialization:

1. Initialize $\hat{c}_l$ of all links.
2. Define:
 - a. T as the set of sources not passing through any saturated link.
 - b. B as the set of saturated links.
 - c. $\bar{B} = L - B$ and $\bar{T} = S_{BE} - T$.
3. Set source rate vector to zero.
4. Initialize $T = S_{BE}$ and $B = \emptyset$.

Loop:

Do until ($T = \emptyset$)

1. $\Delta_s = \min_{l \in \bar{B}} \left[\left(c_l - \sum_{s \in T} R_{ls} x_s(t) \right) / \sum_{s \in T} R_{ls} \right]$
2. $x_s(t+1) = x_s(t) + \Delta_s \quad \forall s \in T$
3. Calculate new bottleneck links and update B and $\bar{B}$.
4. $\forall s \in T$; if s passes through any saturated link then
 $T \Leftarrow T - \{s\}$

Output:

Communicate BE source rates to the corresponding nodes.

4 Simulation Results

In this section we examine the proposed flow control algorithm, listed above as Algorithm 1, for a typical NoC architecture. We have simulated a NoC with 4×4 Mesh topology which consists of 16 nodes communicating using 24 shared bidirectional links, each one has a fixed capacity of 1 Gbps. In our scenario, packets traverse the network on a shortest path using a deadlock free XY routing. We also assume that each packet consists of 500 flits and each flit is 16 bit long.

In order to simulate our scheme, some nodes are considered to have a GS data, such as Multimedia, etc., to be sent to a destination while other nodes, which maybe in the set of nodes with GS traffic, have a BE traffic to be sent. As stated in section 2, GS sources will obtain the required amount of the capacity of links and the remainder should be allocated to BE traffics.

We are mainly interested in investigating the fairness properties among source rates. In order to investigate the rate allocation in the optimal sense, we solved problem (8) using CVX [16], which is a MATLAB-based software for disciplined

Fig. 1. Network topology

convex optimization problems. Optimal source rates, obtained by CVX, are shown in Fig. 2.

Source rates obtained from Algorithm 1 is depicted in Fig. 3. The main feature regarding Fig. 1 and Fig. 2 is that both yield equal values for the minimum source rate, i.e. 0.03 Gbps. The main difference is in the aggregate source rate which is greater for the result of Algorithm 1.

In order to compare the results of the proposed Max-Min fair flow control with other fairness criteria, we have accomplished rate allocation based on maximizing the sum of source rates, i.e. the so-called Rate-Sum Maximization, whose results are depicted in Fig. 4. Comparing Fig. 3 with Fig. 4, it's apparent that although Rate-Sum criterion aims at maximizing the sum of source rates, there is no guarantee for the rates of *weak* sources, i.e. sources which achieve very small rate. Indeed, in many scenarios with Rate-Sum criterion, such sources will earn as small as zero.

To compare the results of the three above mentioned schemes in more detail, we have considered five parameters featuring the merit of the different schemes as following:

1. least source rate
2. sum of source rates
3. Variance of source rates with respect to mean value.
4. Jain's fairness Index [17]
5. min-max ratio [17]

These parameters are presented in Table 1. Jain's fairness Index and max-min ratio, are defined by (12) and (13), respectively.

$$\text{Jain's Fairness Index} = \frac{\left(\sum_{s=1}^S x_s\right)^2}{S \sum_{s=1}^S x_s^2} \quad (12)$$

$$\text{Min-Max Ratio} = \frac{\min_{s \in S} x_s}{\max_{s \in S} x_s} \quad (13)$$

From table 1 we realize that rate allocation with Maximum Rate-Sum criteria, yield slightly greater rate-sum from Max-Min Fair criteria, i.e. Algorithm 1. However, as discussed above, Algorithm 1 guarantees that the rate allocation is max-min fair, and hence the minimum source rate wouldn't be greater with any other feasible rate allocation and hence rate allocation is carried out in favor of weak sources. On the contrary, Maximum Rate-Sum has no guarantee on such sources and as a result, the weakest source, has achieved his rate as low as zero. Another point which is worth mentioning is that similarity of the rate allocation to uniform rate allocation is further in the Max-Min scheme. To be more precise, we have calculated the variance of source rates in with respect to mean value of source rates in equilibrium. Table 1 shows that the variance of Max-Min rate allocation, obtained from Algorithm 1, is evidently less than that of Maximum Rate-Sum scheme, which in turn implies the inherent fairness in the Max-Min rate allocation.

Fig. 2. Rate allocation using CVX results

Fig. 3. Rate allocation using Algorithm 1

Fig. 4. Rate allocation using Rate-Sum Maximization

Table 1. Quantitative comparison between different rate allocation schemes

	Least Rate ($\times 10^8$ bps)	Sum of Source Rates ($\times 10^8$ bps)	Variance	Fairness Index	Min-max Ratio
Max-Min Fair (Mathematical Model)	0.310	10.079	0.1558	0.7181	0.1856
Max-Min Fair (Algorithm 1)	0.310	13.545	0.5004	0.5888	0.1148
Maximum Rate- Sum	0	15.349	1.1974	0.4346	0

5 Conclusion

In this paper we addressed the flow control problem for BE traffic in NoC systems. We considered two objectives. First, choosing source rates (IP loads) of BE traffics so that to accomplish flow control in response to demands at a reasonable level. Second, maintaining Max-Min fairness for all sources. Flow control was modeled as the solution to a simple algorithmic solution to an optimization problem. The algorithm can be implemented by a controller which admits a light communication and communication overhead. Finally, we compared the results of the proposed Max-Min fair flow control with Rate-Sum Maximization scheme based on several criteria such as Jain’s fairness index, max-min ratio, and etc. comparison shows using the proposed flow control scheme, rate allocation has larger fairness index, which denotes that the aim of the proposed flow control scheme is to allocate NoC resources in a fair manner.

References

1. Benini, L., DeMicheli, G.: Networks on Chips: A New SoC Paradigm. Computer Magazine of the IEEE Computer Society 35(1), 70–78 (2002)

2. Dally, W.J., Towles, B.: Route Packets, Not Wires: On-Chip Interconnection Networks. In: Design Automation Conference, pp. 684–689 (2001)

3. Cidon, I., Keidar, I.: Zooming in on Network on Chip Architectures. Technion Department of Electrical Engineering (2005)
4. Ogras, U.Y., Marculescu, R.: Prediction-based flow control for network-on-chip traffic. In: Proceedings of the Design Automation Conference (2006)
5. van den Brand, J.W., Ciordas, C., Goossens, K., Basten, T.: Congestion- Controlled Best-Effort Communication for Networks-on-Chip. In: Design, Automation and Test in Europe Conference and Exhibition, pp. 948–953 (2007)
6. Kelly, F.P., Maulloo, A., Tan, D.: Rate control for communication networks: Shadow prices, proportional fairness, and stability. *J. Oper. Res. Soc.* 49(3), 237–252 (1998)
7. Mascolo, S.: Classical control theory for congestion avoidance in high-speed internet. In: Decision and Control IEEE Conference, vol. 3, pp. 2709–2714 (1999)
8. Gu, Y., Wang, H.O., Hong, Y., Bushnell, L.G.: A predictive congestion control algorithm for high speed communication networks. In: American Control Conference, vol. 5, pp. 3779–3780 (2001)
9. Yang, C., Reddy, A.V.S.: A taxonomy for congestion control algorithms in packet switching networks. *J. IEEE Network* 9(4), 34–45 (1995)
10. Hu, J., Marculescu, R.: DyAD - smart routing for networks-on-chip. In: Design Automation Conference, pp. 260–263 (2004)
11. Talebi, M.S., Jafari, F., Khonsari, A., Yaghmae, M.H.: A Novel Congestion Control Scheme for Elastic Flows in Network-on-Chip Based on Sum-Rate Optimization. In: International Conference on Computational Science and its Applications, pp. 398–409 (2007)
12. Chiu, D.M., Jain, R.: Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *J. Computer Networks and ISDN Systems* 17(1), 1–14 (1989)
13. Bertsekas, D.P., Gallager, R.: Data Networks. Prentice-Hall, Englewood Cliffs (1992)
14. Le Boudec, J.Y.: Rate adaptation, Congestion Control and Fairness: A Tutorial. Ecole Polytechnique Fédérale de Lausanne (EPFL) (2001)
15. Bertsekas, D.P.: Nonlinear Programming. Athena Scientific (1999)
16. Grant, M., Boyd, S., Ye, Y.: CVX (Ver. 1.0RC3): Matlab Software for Disciplined Convex Programming, [url{http://www.stanford.edu/boyd/cvx}](http://www.stanford.edu/boyd/cvx)
17. Jain, R., Chiu, D., Hawe, W.: A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems. DEC Research Report TR-301 (1984)