# A Novel Flow Control Scheme for Best Effort Traffic in NoC Based on Source Rate Utility Maximization

Mohammad S. Talebi[1], Fahimeh Jafari[1,2], Ahmad Khonsari[1,3]
[1]School of Computer Science, IPM
[2]Ferdowsi University of Mashhad
[3]ECE Department, University of Tehran
mstalebi@ipm.ir, jafari@ipm.ir, ak@ipm.ir

***Abstract*—Advances in semiconductor technology, has enabled designers to put complex, massively parallel multiprocessor systems on a single chip. Network on Chip (NoC) that supports high degree of reusability and scalablity, is a new paradigm for designing core based System-on-Chip. NoCs provide efficient communication services to IPs: communication services with guarantees on throughput and latency (GS) and communication services with no guarantees on them (BE). However, the run-time management of communication in NoC, especially congestion control mechanism is a challenging task. This paper considers a congestion control scenario which models flow control as a utility-based optimization problem. Since BE traffic is prone to congestion, we assume that GS traffic requirements are being preserved at the desired level and regulate BE source rates with the solution of the optimization problem. We propose an iterative algorithm to solve the optimization problem based on Newton's method. The proposed algorithm can be implemented by a centralized controller with low computation and communication overhead.**

***Keywords-congestion control; Network-on-Chip; utility-based optimization; iterative algorithm***

## I. INTRODUCTION

*Network on Chip* (NoC) is a new paradigm structure for designing future *System on Chips* (SoCs) [1,2], where various IP resource nodes are connected to the router based square network of switches using Resource Network Interfaces, and network is used for packet switched on-chip communication among cores [3]. A typical NoC architecture will provide 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. NoC-based system will be much easily used for design, test and production. NoCs are efficient communication architectures. However the run-time management of their communication, especially congestion avoidance is a challenging task. Congestion control has been already the subject of research in the field of NoC. Furthermore, minimizing the network cost (or maximizing network utility) while maintaining the required *Quality of Service* (QoS) is one of the considerable factors in NoC architecture design.

NoCs provide two types of communication services to IPs: *Guaranteed Service* (GS) and *Best-Effort* (BE) [4]. Guaranteed Service requires reservation of resources so as to insure data integrity, lossless and ordered data delivery, while Best-Effort service does not require any reservation of resources and no assurance are meant to be given. BE services are easy to use, while GS services require careful programming to reserve the required resources in the network

During the past two decades, several strategies for congestion control have been proposed for data networks [5-8]. However, this issue for Network-on-Chip systems is still novel and only a few works exist. [9] has proposed a a flow control strategy for on-chip networks based on prediction of future congestion problems by routers. In [10], a controller has been proposed to determine the appropriate loads for the Sources with Best Effort traffic. Dyad [11] control the congestion by employing adaptive routing during congestion phase.

The aforementioned works in this issue for NoC ([9]-[11]) mainly used prediction-based method to control the flow of sources which are prone to congestion. In contrast, we have applied a different approach. In this paper, we model the flow control as a utility-based maximization problem which is constrained by link capacities. We assume GS services are being preserved at the desired level and rate allocation of BE sources is the main role of the optimization problem. We mainly adopt the framework provided by [8] for data networks.

The rest of the paper is organized as follows: in the next section we present the system model and flow control problem. In section III, we obtain the dual of the optimization problem that motivates our approach. In Section IV, we solve the dual problem using Newton's Method present the resultant congestion control algorithm. The simulation results are given in section V and finally, section VI concludes the paper.

## II. SYSTEM MODEL

Our NoC architecture is based on a two dimensional mesh topology and wormhole routing. In wormhole networks, each packet is divided into a sequence of *flit*s which are transmitted over physical links one by one in pipeline fashion. A hop-to-hop credit mechanism assures that a flit is transmitted only when the receiving port has free space in its input buffer. Our

NoC architecture is lossless, and packets traverse the network on a shortest path using a deadlock free XY routing [3].

High performance wormhole based interconnect systems often include *virtual channels* (VCs) which increase NoC throughput. Furthermore, virtual channels must be included when links have different capacities to allow the multiplexing of several slow streams over a high bandwidth link. Flits of different VCs that contend for the same link bandwidth are time-multiplexed according to some arbitration policy. Our architecture employs a simple policy in which flits of the active outgoing VCs are transmitted in a round-robin manner over the physical link.

We model the congestion control problem in NoC as the solution to an optimization problem. To have more convenience, we turn the aforementioned NoC architecture into a mathematical model as in [8]. In this respect, we consider NoC as a network with a set of links $L$ and a set of sources $S$. A source consists of Processing Elements (PEs) and Input/Output ports. Each link $l \in L$ is a set of wires, busses and channels that are responsible for connecting different parts of the NoC and has a fixed capacity of $c_l$ packets/sec. We also 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 $L(s)$.

As previously stated, there are two types of traffic in a NoC: Guaranteed service (GS) and Best Effort (BE) traffic. For notational convenience, we divide the set of sources, $S$, into two parts, each one representing sources with the same 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.

Our objective is to choose source rates, $x_s$, of BE traffics so that to maximize the sum of utilities of all BE traffics. Hence the maximization problem can be formulated as [8]:

$$\max_{x_s} \sum_{s \in S_{BE}} U_s(x_s) \qquad (1)$$

subject to:

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

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

where $U_s$ is a positive, concave and strictly increasing function of source rate. Optimization variables are BE source rates, i.e. $(x_S,\ s \in S_{BE})$. $U_s$ is monotonic and we also assume that the curvatures of $U_s$ satisfy the following condition:

$$-U_s''(x_s) \ge \frac{1}{\alpha_s} > 0 \qquad \forall s \in S_{BE} \qquad (4)$$

The constraint (2) states that the sum of BE source rates passing through link $l$ cannot exceed its free capacity, i.e. the portion of $c_l$ which hasn't been allocated to GS traffic.

$U_s$ in the economics literature is referred to as *utility function*, hence problem (1) is called a utility-maximization problem. There are many choices for utility function with specific features and behavior. The simplest form of the utility function is the *Identity Function*, i.e. $U_s(x_s) = x_s$, for which the problem (1) turns into a sum-rate maximization. One of the popular forms of utility functions is logarithmic one, which satisfy *Proportional Fairness* [15]. In this paper, we will consider a general utility function and will not restrict ourselves to a specific form. The investigation of the features of popular utility functions on the rates chosen is one of the directions of our future work.

With the above assumptions, problem (1) is a convex optimization problem with linear constraints. Hence it admits a unique maximizer [12][13], i.e. there exists an optimal source rate vector, $x^* = (x_s^*, s \in S_{BE})$ so that to maximize the sum of utilities in problem (1) while satisfying capacity constraints.

Although problem (1) is separable among sources, its constraints will remain coupled across the network. The coupled natured of such constrained problems, necessitate usage of centralized methods like interior point methods which pose great computational overhead onto the system [12][13].

One way to reduce the computational complexity is to transform the constrained optimization problem into an unconstrained one, for which several methods can be used. According to the Duality Theory [12][13], each convex optimization problem has a dual whose optimal solution can lead to the optimal solution of the main problem. In this respect, the main problem retroactively called *Primal Problem*. As the dual problem can be defined in such a way to be unconstrained, solving the dual is much simpler than the primal. In the sequel, we will obtain the dual of problem (1) and solve it using simple iterative algorithms.

For notational convenience, we define:

$$\hat{c}_l = c_l - \sum_{s \in S_{GS}(l)} x_s \qquad (5)$$

Using the standard optimization methods [12], the Lagrangian of the problem (1) can be written as:

$$L = \sum_{s \in S_{BE}} U_s(x_s) - \sum_{l=1}^{L} \lambda_l \left( \sum_{s \in S_{BE}(l)} x_s - \hat{c}_l \right) \qquad (6)$$

where $\lambda_l > 0$ is the Lagrange Multiplier associated with constraint (2) for link $l$. Usually, $\lambda_l$ is called *shadow price* [15] for the economic interpretation of its role in solving the primal problem through dual.

Regarding the Lagrangian of problem (1), the dual function is defined as [12]:

$$g(\lambda) = \max_{x_s} L(x,\lambda) \tag{7}$$

where $\lambda$ is the vector of positive Lagrange multipliers. Thus the dual function is given by:

$$\begin{aligned} g(\lambda) &= \max_{x_s} \sum_{s\in S_{BE}} U_s(x_s) - \sum_{l=1}^{L} \lambda_l \left( \sum_{s\in S_{BE}(l)} x_s - \hat{c}_l \right) \\ &= \max_{x_s} \sum_{s\in S_{BE}} \left( U_s(x_s) - x_s \sum_{l\in L(s)} \lambda_l \right) + \sum_{l=1}^{L} \lambda_l \hat{c}_l \end{aligned} \tag{8}$$

By Karush-Kuhn-Tucker (KKT) Theorem [12][13], we can obtain optimal source rates, i.e. $x^* = (x_s^*, s\in S_{BE})$. In doing so, we should find the roots of $\nabla_x L(x,\lambda) = 0$. By taking the derivative of (6) with respect to $x_s$, we have

$$\frac{\partial L}{\partial x_s} = \frac{\partial U_s(x_s)}{\partial x_s} - \sum_{l\in L(s)} \lambda_l \tag{9}$$

Duality theory states that the optimal source rate vector, $x^*$, corresponds to the optimal Lagrange multiplier vector, $\lambda^*$ [12][13]. In other words, if $x$ is a feasible point of the primal problem and $x$ is primal-optimal, the corresponding $\lambda$ will be dual-optimal and vice versa. Therefore, at optimality we have

$$\nabla_x L(x,\lambda)\Big|_{(x^*,\lambda^*)} = \mathbf{0} \tag{10}$$

where $\mathbf{0}$ is a vector with all zero. From (9), we have

$$\frac{\partial L}{\partial x_s}\Big|_{(x_s^*,\lambda^*)} = \frac{\partial U_s(x_s^*)}{\partial x_s^*} - \sum_{l\in L(s)} \lambda_l^*$$

Hence, the optimal source rate is given by

$$x_s^* = f\left( \sum_{l\in L(s)} \lambda_l^* \right) \tag{11}$$

where $f$ is the inverse function of $U_s'$ whose existence is guaranteed by monotonicity of $U_s$ in strict sense.

Substituting $x_s^*$ into (8) yields

$$g(\lambda) = \sum_s \left( U_s(x_s^*) - x_s^* \sum_{l\in L(s)} \lambda_l \right) + \sum_{l=1}^{L} \lambda_l \hat{c}_l \tag{12}$$

where $x_s^*$ is given by (11).

The dual problem is defined as [13]:

$$\min_{\lambda\geq 0} g(\lambda) \tag{13}$$

The dual problem is always convex regardless of convexity or non-convexity of the primal problem. Moreover, the dual problem can be defined to be unconstrained or constrained with simple constraints. Thus, the primal problem has been transformed into an unconstrained convex optimization problem.

Convexity of the primal problem (1) guarantees strong duality. Thereby the duality gap is zero and solving the dual problem leads to optimal point of the primal [12]. Since dual problem is convex, it admits a unique optimum, i.e. a unique minimizer, which can be obtained using optimization algorithms. As the dual problem is unconstrained; solving (13) using search methods is much simpler than the primal.

There exist several methods to search the optimal point of an unconstrained optimization problem iteratively [12]. One famous and simple ones is Gradient Projection Method [12] which uses simple mathematical operations. Another famous one is Newton Method that has better convergence behavior at the expense of higher computational complexity [12]. Due to need for faster convergence, in this paper we use the Newton's Method to solve problem (13).

For notational convenience in solving the problem using the Newton's Method, in the rest of the paper we may use matrix notation. To this end, we define Routing matrix, i.e. $R = [R_{ls}]_{L\times S}$, as following:

$$R_{ls} = \begin{cases} 1 & \text{if } s\in S_{BE}(l) \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

We also define the source rate vector (for BE traffic) and link capacity vector as $x = (x_s, s\in S_{BE})$ and $\hat{c} = (\hat{c}_l, l\in L)$, respectively.

## III. Flow Control For Best Effort Sources

In this section, we will solve the dual problem using Newton's Method [12] and present a congestion control mechanism to be run for BE traffic by a controller in NoC systems.

The Newton's Method adjusts shadow prices, i.e. Lagrange multiplier vector, in opposite direction to the scaled version of gradient of the dual function as follows [12]:

$$\lambda(t+1) = \left[ \lambda(t) - \gamma(t) [\nabla^2 g(\lambda(t))]^{-1} \nabla g(\lambda(t)) \right]^+ \tag{15}$$

where $\lambda(t+1) = (\lambda_l(t+1),\ l\in L)$, $\gamma(t) > 0$ is a stepsize, $[x]^+ \triangleq \max\{x,0\}$ and $\nabla^2 g(\lambda(t))$ is the Hessian of $g(\lambda)$. Since $U_s$ is strictly concave, $g(\lambda)$ is continuously differentiable [13], hence $\nabla g(\lambda)$ exists. Using (14), the $l$-th element of the gradient vector is given by:

$$\frac{\partial g(\lambda)}{\partial \lambda_l} = \frac{\partial}{\partial \lambda_l}\sum_s \left( U_s(x_s^*) - x_s^* \sum_{l\in L(s)} \lambda_l \right) + \hat{c}_l \qquad (16)$$

Regarding the system model, we have

$$\sum_s x_s^* \sum_{l\in L(s)} \lambda_l = \sum_l \lambda_l \sum_{s\in S_{BE}(l)} x_s^*$$

Therefore,

$$\frac{\partial g(\lambda)}{\partial \lambda_l} = \hat{c}_l - \sum_{s\in S_{BE}(l)} x_s^* \qquad (17)$$

or equivalently in the matrix form

$$\nabla g(\lambda) = \hat{c} - Rx \qquad (18)$$

To obtain the Hessian of $g(\lambda)$, we have

$$\nabla^2 g(\lambda) = -R\nabla_\lambda x \qquad (19)$$

or equivalently,

$$\frac{\partial^2 g(\lambda)}{\partial \lambda_k \partial \lambda_l} = -\frac{\partial}{\partial \lambda_k}\sum_s R_{ks} x_s^* \qquad (20)$$

Substituting (11) into above equation, yields

$$\begin{aligned}\frac{\partial^2 g(\lambda)}{\partial \lambda_k \partial \lambda_l} &= -\frac{\partial}{\partial \lambda_k}\sum_s R_{ks} x_s^* \\ &= -\frac{\partial}{\partial \lambda_k}\sum_s R_{ks} f\left(\sum_{l\in L(s)} \lambda_l^*\right) \\ &= -\frac{\partial}{\partial \lambda_k}\sum_s R_{ks} f\left(\sum_l R_{ls}\lambda_l^*\right) \qquad (21)\end{aligned}$$

Using the rule of derivation for inverse function, we have

$$\frac{\partial^2 g(\lambda)}{\partial \lambda_k \partial \lambda_l} = -\sum_s \sum_l R_{ls} R_{ks} \left(\frac{1}{U_s''(x_s^*)}\right)$$

Defining $F(t)$ as the following

$$F(t) = \mathrm{diag}(-1/U_s''(x_s(t)), s \in S_{BE}) \qquad (22)$$

we have

$$\nabla^2 g(\lambda) = RF(t)R^T \qquad (23)$$

and the update equation is given by:

$$\lambda(t+1) = \left[\lambda(t) - \gamma(t)\left(RF(t)R^T\right)^{-1}(\hat{c} - Rx)\right]^+ \qquad (24)$$

where $x_s(\lambda(t))$ is the approximate of $x_s^*$ in time $t$.
(18)

The abovementioned update equation necessitates matrix inversion in each iteration which imposes very large computational complexity to the system. One remedy to this problem is to consider the main diagonal elements of the Hessian and to ignore cross terms. Regarding this simplification, we only need to calculate the main diagonal elements of $RF(t)R^T$. By defining

$$E(t) = [E_{ij}(t)]_{L\times L}$$

$$E_{ij}(t) = \begin{cases} [RF(t)R^T]_{ii} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \qquad (25)$$

The update equation using the simplified method can be rewritten as:

$$\lambda(t+1) = \left[\lambda(t) - \gamma(t)E^{-1}(t)(\hat{c} - Rx)\right]^+ \qquad (26)$$

where $E(t)$ is a diagonal matrix and its inverse calculation poses very light computational load onto the system. It is worthnoting that (26) admits a very simple scalar form as:

$$\lambda_l(t+1) = \left[\lambda_l(t) - \frac{\gamma(t)}{[RF(t)R^T]_{ll}}\left(\hat{c}_l - \sum_{s\in S_{BE}(l)} x_s\right)\right]^+ \qquad (27)$$

which in turns implies that the necessary mathematical operations using the simplified method only involve simple operations and admit very low computational complexity overhead.

(11) and (26) together form an iterative algorithm as the solution to the problem (13) and thereby problem (1). In this respect, optimal source rates for BE sources can be found while satisfying capacity constraints and preserving GS traffic requirements. Thus, the aforementioned algorithm can be used to control the flow of the BE sources in the NoC. The aforementioned iterative solution can be addressed in distributed scenarios. However, due to well-formed structure of the NoC, we focus on a centralized scheme; we consider a controller to be mounted in the NoC to implement this algorithm. The necessary requirement of such a controller is the ability to accommodate mathematical operations especially performing matrix inversion as in (11) and (26) and the allocation of few dedicated links to communicate flow control information to nodes with a light GS load. We summarize the proposed algorithm for Best Effort traffic as follows.

**Algorithm 1: Congestion Control for BE Traffics in NoC**

***Initialization:***

1. **Initialize** $\hat{c}_l$ **of all links.**
2. **Set link price vector to zero.**

***Loop:***

**Do until** $(\max|x_s(t+1)-x_s(t)|<Error)$

1. $\forall l \in L$ **: Compute new link prices:**

$$\lambda(t+1)=\left[\lambda(t)-\gamma(t)E^{-1}(t)(\hat{c}-Rx)\right]^{+}$$

**where** $\gamma(t)$ **can be selected as** $\gamma(t)=a/(b+t)$ .

2. **Compute new BE source rates as follows**

$$x_s(t+1)=f(\sum_{l\in L(s)}\lambda_l(t+1))$$

**where** $f^{-1}=U_s^{'}(x_s)$

***Output:***

**Communicate BE source rates to the corresponding nodes.**

Stepsize has an important role on the convergence behavior of the update equation. There are several choices for stepsize, each one belonging to a predefined category and having certain advantages and drawbacks (see [14] and references herein).

In the family of iterative algorithms for distributed scenarios, stepsize is usually chosen to be a small enough constant so that to guarantee the convergence of the algorithm. Constant stepsize benefits from robustness against propagation delay and errors in estimation especially in asynchronous schemes[1]. However, it mainly suffers from slow convergence rate. On the contrary, time-varying stepsizes can be adapted to vary to achieve faster convergence rate. Due to well-formed structure of the NoC and its unified administration, in this paper we use a time-varying stepsize. Several categories for time-varying stepsize exists [14]. In this paper, we focus on a specific category known as *square-summable but not-summable* which satisfy the following conditions [13][14]:

$$\gamma(t)\geq 0 \qquad \forall t \tag{28}$$

$$\sum_{k=1}^{\infty}\gamma^2(t)<\infty \tag{29}$$

$$\sum_{k=1}^{\infty}\gamma(t)=\infty \tag{30}$$

One typical example is of the form $\gamma(t)=a/(b+t)$, where $a>0$ and $b\geq 0$, which we will use in our simulations.

[1] Note that (24) presents a synchronous scheme, and may diverge in asynchronous cases, e.g. real world conditions with large delays, etc.

## IV. Simulation Results

In this section we examine the proposed congestion control algorithm, listed above as Algorithm 1, for a typical NoC architecture. We have simulated a NoC with $4\times 4$ Mesh topology which consists of 16 nodes communicating using 24 shared bidirectional links; each one has a fixed capacity of 1 Gbps. We assume packets traverse the network on a shortest path using a deadlock free XY routing. 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 Guaranteed Service 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 Best Effort traffic to be sent. As stated in section II, GS sources will obtain the required amount of the capacity of links and the remainder should be allocated to BE traffics.

In our simulation we have chosen logarithmic utility functions. In this respect, for source $s$ , we choose $U_s(x_s)=\log x_s$ . Such a utility function satisfies fair conditions among sources and is said to be *Weighted Proportionally-Fair* which is an important property in economics [15]. Due to this property, such utility functions exhibit fair behavior across all nodes.

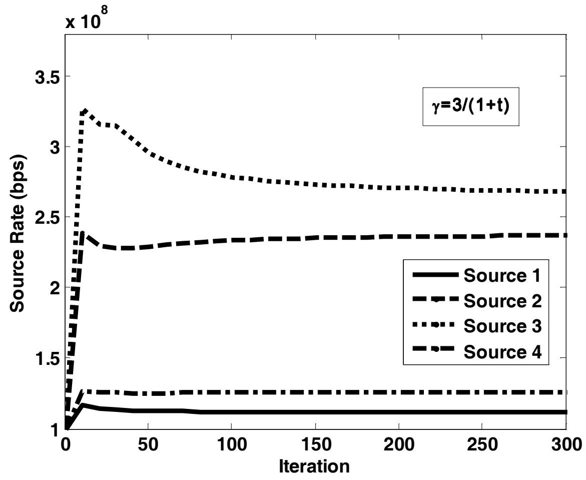
On of the most significant issues of our interest, is the convergence behavior of the source rates. We used two different scenarios for step-size; both of them are chosen to be *square-summable but not summable.* In this regard, step sizes are chosen as $\gamma=3/(1+t)$ and $\gamma=1/(1+t)$ which satisfy (28)-(30).

Variation of source rates for some nodes using aforementioned step sizes are shown in Fig. 1(a)-(b). Regarding Fig. 1(a), it's apparent that after about 80 iterations, all source rates will be in the vicinity of the steady state point of the algorithm. However, for the second case, Fig. 1(b) reveals that at least 100 iterations needed to have source rates in the vicinity of the optimal point. Comparing Fig. 1(a) and 1(b), we realize that the initial value of the step size, directly influences the rate of convergence.

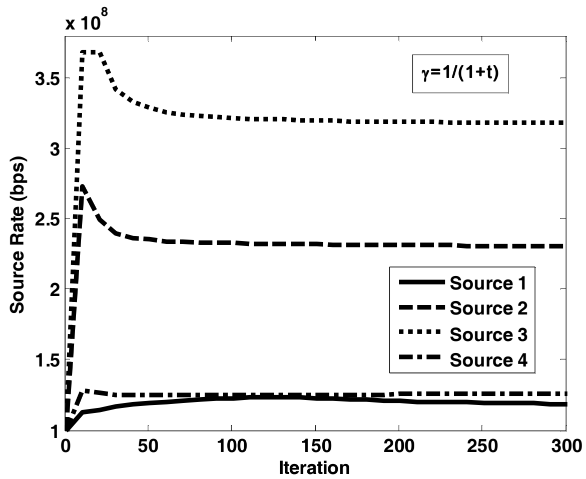
In order to have a better insight about the algorithm behavior, the relative error with respect to optimal rates which averaged over all sources, is also shown in Fig. 2. It is worthnoting that optimal source rates are obtained using CVX [16] which is a MATLAB-based software for solving disciplined convex optimization problems.

## V. Conclusion and Future Work

In this paper we addressed the problem of congestion control for BE traffic in NoC systems. Congestion control was considered as the solution to the source rate utility maximization problem which was solved indirectly through its dual using Newton's method. This was led to an iterative algorithm which can be used to determine optimal BE source rates and thereby as a means to control the congestion of the NoC. The algorithm can be implemented by a controller which admits a light communication and communication overhead. Further investigation about convergence behavior of the algorithm and the effect of different utility functions on the




(a)




(b)

Figure 1. Source rates for (a) $\gamma = \dfrac{3}{1+t}$ and (b) $\gamma = \dfrac{1}{1+t}$.

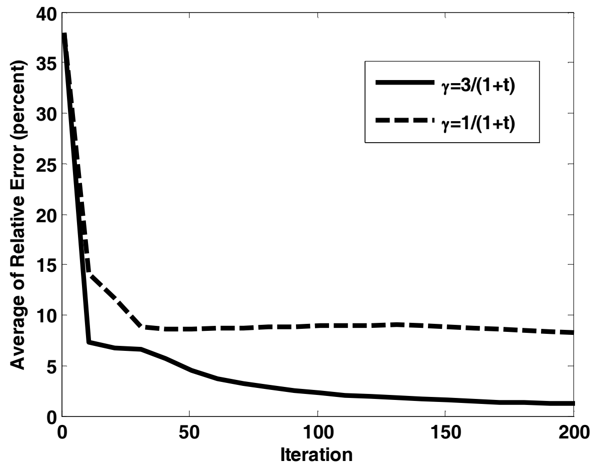



Figure 2. Average of relative error with respect to optimal solution for the three cases.

BE rates and fairness provision is the main directions of our future studies.

## REFERENCES


[1] L. Benini, and G. De Micheli, “Network on Chips: A New SoC Paradigm”, IEEE Computer, pp. 70-78, January 2002.

[2] A. Jantsch, and H. Tenhunen, "Networks on Chip", Kluwer Academic Publishers, Boston, USA, January 2003.

[3] W. J. Dally, and B. Towles, “Route Packets, Not Wires: On-Chip Interconnection Networks”, Proceedings of the 38th Design Automation Conference, ACM/IEEE, Las Vegas, Nevada, USA, pp. 684-689, June 2001.

[4] B. Gebremichael, F.W. Vaandrager and M. Zhang, “Formal Models of Guaranteed and Best-Effort Services for Networks on Chip”, Technical Report ICIS-R05016, ICIS, Radboud University Nijmegen, March 2005.

[5] G. Almes et al. RFC2581: “TCP congestion control”. Technical report, Network Working Group, 1999.

[6] Y. Gu et al. “A predictive congestion control algorithm for high speed communication networks”. Proceedings of American Control Conference, 2001.

[7] C. Yang et al. “A taxonomy for congestion control algorithms in packet switching networks”. IEEE Network, 9, 1995.

[8] S. H. Low. “Optimization Flow Control, I: Basic Algorithm and Convergence”, IEEE/ACM Transactions on Networking, 7(6): 861-874, 1999.

[9] U. Ogras et al. “Prediction-based flow control for network-onchip traffics”. Proceedings of Design Automation Conference (DAC) , 2006.

[10] J. W. van den Brand, C. Ciordas, K. Goossens1 and T. Basten. “Congestion-Controlled Best-Effort Communication for Networks-on-Chip”. Proceedings of Design, Automation and Test in Europe Conference and Exhibition (DATE), April 2007.

[11] J. Hu et al. “DyAD - smart routing for networks-on-chip”. Proceedings of Design Automation Conference (DAC), 2004.

[12] S. Boyd and L. Vandenberghe, “Convex Optimization”. Cambridge, U.K, Cambridge Univ. Press, 2004.

[13] Dimitri P. Bertsekas, ”Nonlinear Programming”, Athena Scientific. 1999.

[14] S. Boyd, “Convex Optimization II Lecture Notes”, Stanford University, 2006.

[15] F. P. Kelly. “Charging and rate control for elastic transaction”. European Transactions on Telecommunications, 8:33{37, 1997.

[16] Michael Grant, Stephen Boyd and Yinyu Ye., “CVX (Ver. 1.0RC3): Matlab Software for Disciplined Convex Programming”, Download available at: http://www.stanford.edu/~boyd/cvx.