Deadlock Free Routing Algorithm for Minimizing Data Packet Transmission in Network on Chip

K. Somasundaram, Amrita Vishwa Vidyapeetham, India, and University of Turku, Finland
Juha Plosila, University of Turku and Academy of Finland, Finland

ABSTRACT

Network on chip (NoC) has been proposed as a solution for addressing the design challenges of future high performance nanoscale architectures. In NoCs, the traditional routing schemes are routing packets through a single path or multiple paths from one source node to a destination node, minimizing the congestion in the routing architecture. Although these routing algorithms are moderately efficient, they are time dependent. To reduce overall data packet transmission time in the network, the authors consider a network with multiple sources and multiple destinations. Multi-dimensional routing problems appear naturally in several resource allocation problems, communication networks and wireless sensor networks. In this paper, the authors have constructed a deadlock-free multi-dimensional path routing algorithm for minimizing the congestion in NoC.

Keywords: Data Packet Transmission, Deadlock-Free, Multi-Dimensional Routing, Nanoscale Architectures, Network on Chip (NoC)

1. INTRODUCTION

Systems on Chip (SoC) grow in complexity with the advancement of semiconductor technology enabling integration of dozens of cores on a chip. The continuously increasing number of cores calls for a new communication architecture as traditional architectures are inherently non-scalable, making communication a bottleneck. System architectures are shifting towards a more communication centric methodology.

Network on Chip (NoC) has emerged as the design paradigm for design of scalable on-chip communication architectures, providing better structure and modularity than its predecessors (De Michel & Benini, 2006; Rantala, Lehtonen, & Plosila, 2006).

Figure 1 illustrates the implementation of the system d695 from the ITC’02 SoC Test benchmarks suit in SOCIN topology (Marinissen, Iyengar, & Chakrabarty, 2002). This is also a topological aspect of a 4×4 mesh NoC, which shows a model of global level on-chip communication. Instead of busses and dedicated
point-to-point links, a more general scheme is adopted by employing a grid of routing nodes spread out across the chip and connected by communication links. Such regular and parallel structures are very attractive because they can offer well-controlled electrical parameters which enable high-performance circuits by reducing latency and increasing bandwidth. From this perspective, the SoC design will resemble more the creation of large scale communication networks rather than traditional IC design practice.

The simple $4 \times 4$ mesh NoC in Figure 1 consists of Network Interfaces (NIs), Cores, Routers and Links. One Network Interface (NI) is needed to connect each IP core to the NoC. Network Interfaces are converting the transaction request/response into packets and vice versa. For example, in $\times$pipes (Bertozzi & Benini, 2004; Dall’Osso, Biccarì, Giovannini, Bertozzi, & Benini, 2003), two separate NIs are defined for system master and system slaves. The NIs implement the interface by which the cores are connected to the NoC, decoupling computation (the cores) from communication (the network). Links are connected between the nodes with either static or dynamic bandwidth. They may consist of one or more logical or physical channels. The routing methods can be applied either after NoC topology mapping and design or during the mapping process itself. Routing nodes are required to route the data according to chosen protocols, implementing the desired routing strategies.

The general routing schemes in the NoC are either static or dynamic. In static routing, bandwidth and the traffic flows are fixed. On the other hand, in the case of dynamic routing, the paths are selected based on the current bandwidth and traffic flow characteristics of the network. In most of the NoC designs the bandwidth and data width are fixed. Creating a NoC based system requires selection of cores for various purposes and efficient mapping of the cores on the platform. Design choices include binding between core ports and network ports, communication between cores, and allocation of network channel capacity. In most of the NoC designs the bandwidth and data width are fixed, because frequent dynamic changes in traffic flows and routing paths would mean dynamic hop counts and a complex on-chip system implementation. Therefore, in this paper, we concentrate on static routing only.

This paper is organized as follows: A brief review of routing algorithms is given in the next section. Section 3 gives some basic definitions and notations and shows a mathematical model for our problem. In Section 4, we focus on the path selection procedure and its existence. A multi-dimensional flow algorithm for minimizing the data packet transmission is discussed in Section 5. Our experimental results are given in

---

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
Section 6. Finally, we draw some conclusions in Section 7.

2. PREVIOUS WORK

General single path routing algorithms are well studied by many authors (Anido & Seeto, 1988; Banner & Orda, 2007; Faloutsos, 1999; Wang, Jin, Kim, & Malik, 2002). Current routing schemes typically focus on discovering a single or multiple optimal paths for routing from one source to one destination (Banner & Orda, 2007; Bjerregaard & Mahadevan, 2006).

There are several schemes for routing the network traffic with minimum congestion. Multipath algorithms aim to find multiple paths between a source node and a destination node. Many multipath algorithms have been devised in the context of flow control (Banner & Orda, 2007; Cidon, Rom, & Shavitt, 1999; Hansson, Goossens, & Radulescu, 2007; Jia, Nikolaidis, & Gburzynki, 2001; Dally, Carvey, & Dennison, 2003). Several works in the multiprocessor field have focused on the design of efficient routing strategies. Packets of a single flow follow a single path, while different flows can use different paths. Multi-Commodity flow problem is an optimization problem where one wants to route all the traffic demands without exceeding the arc capacities and with an objective in the routing cost or efficiency (Gourdin & Klopfenstein, 2006). There are several multipath schemes for utility minimization in a communication network. Some of such utilities are traffic congestion and the hop count (Lin & Shoff, 2006; Carvey & Dennison, 2003). Recursive Partitioning Multicast (RPM) is a routing technique, where the routing path is computed based on all the destination positions in a network and the network is recursively partitioned according to the position of the current router (Wang, Jin, & Kim, 2009).

The path uniqueness of an interconnection has a disadvantage. Since the number of paths may share common links within the network, simultaneous connections of more than one inlet and outlet pair may result in conflicts in the use of these internal links. In general, these conflicts can be resolved by providing a small amount of buffering per switching element. Such single buffered networks have been analyzed by many authors (Anido & Seeto, 1988; Hansson, Goossens, & Radulescu, 2007; Jia & Nikolaidis, 2001; Rantala, Lehtonen, & Plosila, 2006). A good survey of research and practice of NoC can be found in (Bjerregaard & Mahadevan, 2006).

Multimedia applications on a System-on-Chip (SoC) are extensively being studied for bandwidth requirements over heterogeneous components of the network. However, we have focused here on the quality-of-service (QoS) environment in homogenous networks such as chip multiprocessors. An on-chip network must support guarantee for the delivery of multimedia data (real-time traffic) as well as the normal message-oriented communication (best-effort traffic). For example, in Nguyen, Ngo, and Choi (2005), a Video Object Plane Decoder (VOPD-16 cores), used in high-end video applications, has been designed and evaluated based on a $4 \times 4$ mesh topology similar to the one presented Figure 1.

Multi-dimensional routing is a routing scheme focusing on more than one source-destination pair at a time. These multi-dimensional routing algorithms aim at finding an optimal overall flow by reducing congestion in hot spots of the network. The idea is to consider simultaneously the flows between all source and destination nodes. This is fundamentally more efficient than the current single/multi-path schemes, enabling significant reduction of congestion by optimizing the routing in the network as a whole. In this paper, we will show that our multi-dimensional routing algorithm will minimize the congestion in the network avoiding cyclic deadlocks.

3. PROBLEM FORMULATION

Let $G=(V, E)$ be a digraph which represents a network, where $V$ is set of nodes and $E$ is set of links in the network. We denote by $n$ and $m$ the number of nodes and links in $G$ respectively. We
denote by $C_e$ the capacity of a link $e$, $e \in E$. A path $p$ is a succession of nodes $(v_1, \ldots, v_l)$, $v \in V$ such that $(v_k, v_{k+1}) \in E$ for all $k \in \{1, \ldots, l-1\}$. A path $p$ is said to be a simple path if each one of its nodes appears only once in $p$. In this paper, a path is always a simple path without any cycles, guaranteeing freedom from cyclic deadlocks.

Given a source node $s \in V$ and destination node $t \in V$, $P(s, t)$ is the set of all directed paths in $G$ from $s$ to $t$. Let $S = \{s_1, \ldots, s_i\}$ and $T = \{t_1, \ldots, t_k\}$, $1 \leq i, k < n$, be respectively, the sets of source nodes and destination nodes and $P(S, T)$ be the set of all directed paths from $S$ to $T$.

A network graph is a graph obtained from an NoC topology. Here the vertices/nodes represent the cores and the edges/links are representing the links between cores. In the case of a network graph, we consider a node as a combination of core, NI and router. An example of a network graph for a $4 \times 4$ mesh NoC (Figure 1) is shown in Figure 2.

In this paper, we consider a link-state routing environment, where each source node has an image of the entire network. Each link $e \in E$ is assigned a length $l_e, l_e \in \mathbb{Z}^+$ and capacity $C_e, C_e \in \mathbb{Z}^+$. Given a path $p$ in $P(S, T)$, the length $L(p)$ of $p$ is defined as the sum of lengths of its links. That is:

$$L(p) = \sum_{e \in p} l_e$$

We define flow $f_e$ of a link $e \in E$ as a non-negative quantity and $f_e \leq C_e$.

In each traffic flow on each link, one can consider either the relative load $f_e / C_e$ or the residual bandwidth $C_e - f_e$. This choice depends on the specific application. For instance, if the main concern is to face unpredicted demand arrivals or demands rerouted after a failure, keeping as much bandwidth available as possible on all the links is preferable; this corresponds to maximizing the residual bandwidth. On the other hand, if the main goal is to palliate a proportional increase of all the demands, the best routing corresponds to minimizing the maximal relative link load. In this paper, we have taken the residual bandwidth as a congestion factor of the link $e$. That is, for each link $e \in E$, the congestion factor $\gamma(e)$ is defined as

$$\gamma(e) = C_e - f_e$$

For a given network, the value of a maximum flow is equal to the capacity of a minimum cut (Ahuja, Manati, \& Orlin, 1993; Murali, Atienza, Benini, \& Micheli, 2007; Gross \& Yellen, 2006). Hence, minimizing the network congestion factor is equivalent to maximizing the flow in the network.

Let $in(v)$ be the set of all links incident into $v$ and let $out(v)$ be the set of all links incident out of $v$. The flow in a network is a function $f$ that assigns a real number to each link in the network graph and also satisfies the following constraints:

$$0 \leq f_e \leq C_e \ \forall e \in E$$

and

$$\sum_{e \in in(v)} f_e - \sum_{e \in out(v)} f_e \geq 0, \ \forall v \in V \setminus (S \cup T)$$

Figure 2. Graph model of $4 \times 4$ NoC grid
We define the excess of a node $v$ as

$$ex_f(v) = \sum_{e \in \text{in}(v)} f_e - \sum_{e \in \text{out}(v)} f_e$$

A maximum flow is a flow of maximum $ex_f(v), v \in T$. We denote by $rev(e)=(v,u)$ the reverse link of the link $e=(u,v)$. The residual capacity of a link $e$ is $\gamma(e)=C_e - f_e$, whereas the residual capacity of a reverse link $rev(e)$ is $\gamma(rev(e))=f_e$. A link $e$ is called augmenting if $\gamma(e)>0$, and $rev(e)$ is augmenting if $\gamma(rev(e))>0$. A path $p$ in $G$ is said to be an augmenting path if its links are augmenting. We denote by $A_e$ the set of all augmenting paths in the network from $S$ to $T$ such that no link intersects among the paths. Furthermore, we denote by $l_e$ the length of the link $e$ and by $f^e$ the total flow along the links that has been routed from a source node $s$ to $t$ through paths with a total length $\lambda$, where $e = (s, t)$.

If $P \in P^{S,T}$, then $\gamma(P)$ is the congestion factor in the path $P$. Our aim is to minimize the total congestion in the entire network.

Hence our problem is:

Minimize \[
\sum_{P \in P^{S,T}} \gamma(P)^{v}
\]

Subject to

$$ex_f(v) = 0 \quad \forall v \in (S \cup T)$$

$$\sum_{e \in \text{in}(v)} f_e - \sum_{e \in \text{out}(v)} f_e = 0$$

$$\sum_{e \in \text{in}(v)} f_e = d$$

$$f_e \leq C_e, \forall e \in E$$

$$\int_{e} = 0 \quad \forall e \in E, \lambda \in (0, L - l_e)$$

$$\int_{e} \geq 0 \quad \forall e \in E, \lambda \in (0, L)$$

If $l = 1$, then $L(P) \leq H(P), \forall e \in E$ and $\forall P \in P^{S,T}$.

The objective function minimizes the network congestion factor. Constraint (1) states that the total inflow and outflow are equal at any internal node $v$, $v \in V \setminus (S \cup T)$. Equation (2) gives a condition that the amount of traffic occurring at the destination nodes is the same as the amount of traffic initiated at the source nodes. This will ensure that there is no loss of data at the receiving ends. Equation (3) states that the total flow at the source nodes must be equal to a given demand $d$. The inequality (4) shows that for any link in $G$, flow never exceeds the link capacity and (5) gives the link capacity utilization constraint. The constraint (6) is indicating that the flows are nonnegative. We denote by $H(P)$ the hop count for the path $P$, and assuming that the length of each link is one unit, then the constraint (7) gives the condition for the hop count.

As an example, consider a $4 \times 4$ mesh network for some traffic patterns, where the source nodes are indicated by $\bigcirc$ and the destination nodes are indicated by $\blacklozenge$.

Figures 3(a) and 3(b) illustrate two different routings on the $4 \times 4$ mesh network. Figure 3(a) shows a routing with two common links. If two links are shared by different paths, then their cores are also shared by the paths. This leads to buffering not only due to the links but also due to the cores. On the other hand, Figure 3(b) shows that there is no congestion at any link. The routing strategy in Figure 3(b) is less congested, even though its overall hop count is larger than that of the first one (Figure 3(a)).

4. PATH SELECTION

In the present section, we will construct the augmenting paths from $S$ to $T$ in a network graph. In a general situation, the number of paths between a source and destination node is exponential. One can choose an augmenting path as the shortest path from a source node to a destination node using Dijkstra’s shortest
path algorithm (Gross & Yellen, 2006; Orlin, Madduri, Subramani, & Williamson, 2009), but this does not guarantee that the minimum path length constraint is met. We can also solve our mathematical model by using linear programming with polynomial time. This is a standard flow technique applied in many flow problems. However, this type of techniques cannot be used for our purpose here since they do not respect the path length restrictions either. Hence, we are proposing two algorithms for constructing all possible augmenting paths from \( S \) to \( T \). First we find an augmenting path from any one source to any one destination using the Construction of Augmenting Path algorithm (CAP), and using Successive Augmenting Paths algorithm (SAP) we will then generate all possible augmenting paths from \( S \) to \( T \).

Our augmenting path construction is based on a vertex labeling strategy and tree growing method. Here, the idea is to grow a tree of paths with labeling. In each tree, we increase the label of each link starting from the source node \( s \). We will obtain the augmenting path as soon as the destination node \( t \) is labeled.

We start with a source node \( s \) and labeled as zero. Let \( w \) be an adjacent node to \( s \) with positive \( \sum_{e \in \delta(s)} f_e = d > 0 \), where \( e \) is the link between \( s \) and \( w \). Now, the node \( w \) will be added to the set \( V=\{s\} \). Increment \( \Delta \) as \( \Delta + f_{ew} \), the path construction will find new nodes with minimum labels and finally a path will be generated by tracing the nodes from \( t \) in backward direction.

Using the above CAP algorithm, we can construct one augmenting path. Based on this procedure, we can generate a set of augmenting paths such that the augmenting paths are link disjoints. The Successive Augmenting Path (SAP) algorithm generates all the possible augmenting paths in a network.

**Successive Augmenting Paths (SAP)**

\[ G = \text{network} \]
\[ \text{If augmenting path } p \text{ exist in } G \]
\[ \text{Find an augmenting path } p \text{ in } G \text{ using CAP} \]
\[ G = G - \{ e \in p \ / \ e \text{ is augmenting path} \} \]
\[ \text{Terminate with set } A_P \text{ of augmenting paths.} \]

For example, consider a network graph with 10 nodes and 17 links as in Figure 4. Let \( S = \{ n_1, n_2, n_3 \} \) be a set of source nodes and \( T = \{ n_9, n_{10} \} \) be a set of destination nodes. The path set \( P_1 = \{ n_1 \rightarrow n_4 \rightarrow n_7 \rightarrow n_9, n_2 \rightarrow n_5 \rightarrow n_8 \rightarrow n_9, n_3 \rightarrow n_6 \rightarrow n_7 \rightarrow n_{10} \} \) can be an augmenting path even though the two paths are sharing the node \( n_7 \). But the set paths \( P_2 = \{ n_1 \rightarrow n_4 \rightarrow n_5 \rightarrow n_9, n_2 \rightarrow n_4 \rightarrow n_7 \rightarrow n_{10}, n_1 \rightarrow n_6 \rightarrow n_7 \rightarrow n_9 \} \) cannot be an augmenting path since the link \( (n_7, n_9) \) is shared by two paths.

Our SAP will generate all possible augmenting paths in the network graph from \( S \) to \( T \) without any link sharing. This will avoid the buffering at the link levels. The hop count re-

---

Figure 3. (a) Routing with common links; (b) routing with distinct links

(a) ![Routing with common links](image1.png)

(b) ![Routing with distinct links](image2.png)
Restriction is one of the important measures in NoC routing. The following theorem gives the existence of a path in CAP with the hop count restriction. The proof outline of the theorem is similar to that of Lemma 1 in Banner and Orda (2007).

**Theorem 1.** Let $G$ be a network graph. Then there exists a $k$, $k \leq L$, such that the path sequence $(n_0, n_1, \ldots, n_k)$ is a path from source to destination in $G$.

**Proof:** As $f_{v_i}^\ell > 0$, there exists a link $e_i \in E$ such that $e_i = (n_j, n_k)$ and $f_{v_i}^\ell > 0$. By repeated application of the constraints (1) and (2), we get a path of links $e_0 = (n_0, n_1) - e_1 - \ldots - e_i = (n_i, n_{i+1})$ such that $\Delta_i = \sum_{k=0}^{i-1} l_{v_k}$, where $\Delta_i = \sum_{k=0}^{i-1} l_{v_k} > L$ and $n_{i+1} \in T$. It is enough to prove that $i \leq L$.

Suppose $i > L$. It is easy to see that, for every $i$, $i > L$ implies that $f_{v_i}^\ell > 0$.

**Box 1. Construction of Augmenting Path (CAP)**

1. Initialize node set $V = \{s\}$
2. Write label 0 on node $s$ and $\Delta_0$
3. Initialize label counter $i=1$
4. While node set $V$ does not contain the destination node $t$
   - If there are links with positive residuals
     - Let $e$ be link whose labeled endpoint $v$ has the smallest possible label with positive $f^\ell_{v_i}$.
     - Let $w$ be the unlabeled endpoint of link $e$
     - Set back point $(w) = v$
     - Write label $i$ on vertex $w$ and $\Delta_{i+1} = \Delta_i + l_{v_i}$ and $i = i+1$
   - Else
     - Return to an augmenting path by following back points.

**Figure 4. Example for path selection**

---

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
Hence \( p(u, v) = \alpha \exp\left( -\delta(u, v) \right) \), which is a contradiction. Therefore \( i \leq L \). This gives a result that, there exists a path in \( G \) from a source node to a destination node with path length less than or equal to \( L \). Hence, whenever a new edge is added to the path, the path length should be computed and it does not exceed the limit \( L \).

5. CONGESTION MINIMIZATION

In this section, we investigate the multi-dimensional routing algorithm for minimizing the congestion factor in the entire network. Like single path or multi-path optimal flow problems, we can ensure by the constraint (1) that the total inflow and total outflow are equal at any internal node. Similarly, based on the constraint (2) the total outflow of the source nodes is equal to the total inflow at the destination nodes. That is, there is no loss of data packets during the routing.

Multi-Dimensional Flow Algorithm (MDFA) is a flow algorithm which will solve our mathematical model. We initialize all the flows as zero in the first step. From SAP, we find an augmenting path set \( A_P \) with path length restrictions. During the first iteration of MDFA, if \( P \in A_P \), then we will calculate the residual value \( \gamma(P) \), where \( \gamma(P) \) is minimum of the difference between capacity and flow of each link \( e \) in the path \( P \). The minimum among all \( \gamma \) is called gain and it is denoted as \( \Gamma \). This gain will refine the flow values of all the links in the augmenting paths. If a link \( e \) is in the forward direction, then the gain is added to the flow of \( e \), and if \( e \) is in the backward direction, then the new flow is the difference between the flow of \( e \) and the gain. This iteration process will be continued until no augmenting path exists. MDFA terminates with minimum flow congestion.

Let us now briefly discuss the time complexity of MDFA. Finding an augmenting path is culminating from the tree growing method. We can use the breadth first search method to find an augmenting path with the least number of links. In CAP, each iteration takes \( O(n+m) \), where \( n \) and \( m \) are the number of nodes and the number of links, respectively. The number of iterations in CAP is \( O(m) \). In SAP, we are constructing augmenting paths in a successive manner by deleting the links corresponding to previous augmenting paths. Hence the worst case time complexity of SAP is \( O(m(n+m)) \). In practice, the run time of SAP is less than few seconds for all the experimental studies. In MDFA, calculation of \( \gamma(p) \) and \( \Gamma \) will take linear time, and the number of refinement steps is the same as the number of links in the path. Hence, the overall time complexity of MDFA is polynomial.

6. EXPERIMENTAL RESULTS

6.1. Methodology

We consider a network with 200 nodes placed in a 2-D mesh as shown in Figure 1. The network size 200 is chosen, because it represents a substantial system complexity, and, furthermore, it enables convenient application of the probability function \( p(u, v) \) discussed later in this section (Banner & Orda, 2007; Waxman, 1988). We use a cycle-accurate network simulator that models all router pipeline delays and wire latencies. We use Orin (Wang, Zhu, Peh, & Malik, 2002) to estimate dynamic and static power consumption for the buffer, crossbar, arbiter, and link with 50% switching activity and 1V supply voltage in 65nm technology. We assume a clock frequency of 4GHz for the router and link.

In this paper, we are using the static routing strategies. That is, the bandwidth and the link capacities are fixed. Hence, we use the uniform distribution for bandwidth and link capacities. In particular, we assume that the link capacities are uniformly distributed in (5, 150) MB/s, the bandwidth request is uniformly distributed in (1, 5) MB/s, and length of each link is uniformly distributed in (1, 150). We generated a class of topologies, starting with Waxman topologies (Waxman, 1988). We first located the source nodes and destination nodes at the extreme
levels of a square area of unit dimension. Then the remaining nodes spread over the square. We introduce a link between a pair of nodes $u$ and $v$, with the probability function,

$$p(u,v) = \alpha \exp \left[ \frac{-\delta(u,v)}{\beta \sqrt{2}} \right]$$

where $\delta(u, v)$ is the length between $u$ and $v$ and using $\alpha = 2$ and $\beta = 0.2$. This leads to approximately 1800 links per network topology.

Let $\rho(M)$ be the ratio between the total congestion and the total capacity using MDFA and let $\rho$ be the ratio between the total congestion and the total capacity using the single path flow algorithm. From the constraint (4), it is easy to conclude that $C_e - f_e \leq C_e, \forall e \in E$ and hence it is easy to see that $0 \leq \rho(M), \rho \leq 1$. Using Dijkstra’s shortest path algorithm (Gross & Yellen, 2006) we find all possible shortest paths from the nodes in $S$ to the nodes in $T$. Now, let $L^*$ be the minimum length among all shortest paths from $S$ to $T$ and $L$ be the normalized path length in $\{ L^*, 1.1 \cdot L^*, 1.2 \cdot L^*, \ldots, 2 \cdot L^* \}$ (x-axis in Figure 5).

Figure 5 shows that $\rho(M)$ is smaller than $\rho$ and hence MDFA produces minimum network congestion in comparison with the single path flow algorithm. MDFA will drastically reduce the network congestion by routing all possible paths with shortest paths.

### 6.2. Deadlock Avoidance

Deadlock freedom is an important feature for every routing algorithm. Most of deadlock avoidance techniques are designed to remove cyclic dependencies as in Dally and Seitz (1987). Without any turn restriction in routing scheme, flows in the network may cause a

---

**Algorithm 1. Multi-Dimensional Flow Algorithm (MDFA)**

Input the network $G$ as direct graph

$S$ - set of sources and $T$ - set of destinations

Initialize: $f_e = 0$ for all links $e$

Find a set of augmenting path $A_p$ from SAP

While $A_p \neq \emptyset$

{ 
  
  $P$ = augmenting path in $A_p$
  
  $\gamma(P) = \min\{C_e - f_e / \text{for all } e \in P\}$
  
  $\Gamma = \min\{\gamma(P) / \text{for augmenting paths } P \text{ in } A_p\}$
  
  Refinement of the flow $f_e$:
  
  $f_e + \Gamma$ for all forward links $e \in A_p, f_e - \Gamma$ for all backward links $e \in A_p$

End
cyclic dependency, which freezes movement of all the packets in a cycle. Our MDFA depends on generating paths in $A_p$ by SAP and CAP. These two algorithms are free from cycles. When an augmenting path is generated from CAP, it will be a simple path from the source node to the destination node without any node repetition. If the nodes are not repeated, then there is no cyclic dependency and hence there is no deadlock either.

7. CONCLUSION

In this paper, we have proposed a multi-dimensional flow algorithm for network congestion in NoC. We have taken path length and residual capacity as relevant measures of the quality for multi-dimensional flow routing. Our result indicates that congestion reduces significantly and the NoC is free from deadlock. We have established an algorithm in polynomial time.
from the well known maximum flow algorithm with path length restrictions. Our MDFA will minimize the congestion in the network with a given hop count. Moreover, comparing to a single source-destination flow algorithm our MDFA reduces the power dissipation, since the number of augmenting paths generated from SAP is smaller than in any traditional flow algorithms.

If we take the congestion factor as a ratio between capacity and flow then finding an optimal congestion factor for multi-dimensional network is a more challenging problem. In this case, the algorithm becomes NP-complete (Baner & Orda, 2007) and hence we need to find new schemes for optimizing the congestion factor in multi-dimensional networks. Also, finding low-power solutions for multi-dimensional flows with minimum congestion may be another interesting problem for future research.

REFERENCES


K. Somasundaram received a MS degree in mathematics from St. Joseph’s College, Trichy, India, in 1995 and a doctoral degree in mathematics from Bharathiar University, Coimbatore, India, in 2003. He completed his post doctoral research fellowship from Turku Centre for computer science, University of Turku, Finland, in 2009. Currently he is a professor in the department of mathematics, Amrita Vishwa Vidyapeetham, Coimbatore, India. His research interests include linear algebra, graph theory and network on chip.

Juha Plosila received his MS and PhD degrees in electronics and communication technology from the University of Turku, Turku, Finland, in 1993 and 1999, respectively. He is an adjunct professor in digital systems design at the University of Turku, Department of Information Technology, and holds currently a five year position of Academy Research Fellow at the Academy of Finland (2006-2011). He directs, at the University of Turku, an active research group focusing on modeling, design, and verification of network-on-chip (NoC) and 3D integrated systems at different abstraction levels. Plosila’s research interests include formal methods for NoC and embedded system design, reliable on-chip communication techniques, on-chip thermal monitoring and control methods, dynamically reconfigurable systems, and application mapping on NoC based systems.