Integer Linear Programming model for Reconfigurable Routing Problem in a Network-on-Chip with Guaranteed Traffic

[1] Boureima Zerbo, [2] Frédéric T. Ouédraogo, [3] Tiguiane Yelemou, [4] Abdoulaye Séré
[1] Université Ouaga2 - Ouagadougou – Burkina Faso
[2] Université de Koudougou – Burkina Faso
[3, 4] Université Polytechniques de Bobo – Burkina Faso
bzerbo@univ-ouga.bf

ABSTRACT

We define an integer linear programming model of combinatorial optimization problem that models reconfigurable multi-path routing in a Network-on-Chip with guaranteed traffic. Based on time division multiplexing, the model allows avoiding collisions, deadlocks and livelocks in irregular network topologies, while minimizing latency. The model allows dynamic reconfigurable routing. In that case, independent sets of valid routes are pre-computed in such a way they can be interchanged with no impact on the existing traffic, while reusing all the vacant time-slot resources.

Keywords: *Multiprocessor routing problem, network on chip, MPSoC, K-shortest paths problem, integer linear programming, modification, guaranteed traffic routing, reconfigurable routing*

1. INTRODUCTION

The concept of network-on-chip (NoC) is now an adopted solution in semiconductor technology in which networks principle play an important role [4, 2, 1]. As we shrink transistor size to nano scale, designers are developing integrated circuits including complex heterogeneous functional elements into a single device, known as a System-on-Chip (SoC) or Multi-Processor System-on-Chip (MPSoC). In such systems, traditional solutions based on shared-buses have hit their scalability limit. Hence, NoC architectures rather employ interconnection network with short links. As in computer networks or terrestrial transportation networks, the design of efficient routing schemes, for communication or transportation, is a critical issue to allow guarantee of traffic bandwidth, avoid collisions, deadlocks and livelocks. In such networks, Guaranteed Traffic (GT) approaches are often opposed to Best Effort (BE) methods [9]. A key feature of GT is to address conflict-free routing at the time of route computation, whereas BE deals with these problems only at the execution time of the routes and in a simple way. It is widely accepted that BE networks achieve good average performances, but that worst case performances are very hard to predict. GT methods often imply to over-dimension the traffic bandwidth between source-destination nodes, and their implementation suffers from a lack of dynamic adaptiveness to traffic fluctuations. But the GT approach ensures the respect of the application real-time requirements and avoids the possibility of contention and deadlocks while using irregular topologies that allow significant power savings.

In this paper, we present a GT approach where a conflict-free routing scheme is modeled as a mixed integer linear programming problem. We adopt wormhole routing, that is, one of the most widely used flow- control techniques in a NoC. A message being composed of a set of contiguous packets, often called flits (flow control units), wormhole routing operates by advancing the head of a message directly from incoming to outgoing links. Hence, packets are stored in the links while advancing through the network in a pipeline fashion. Only the header packet contains the specification of the path to follow, hence packets must remain in contiguous links of the network and cannot be interleaved with the packets of other messages. Wormhole avoids using memory buffers in the nodes through which messages are routed. We assume that transmissions are synchronous and cadenced by a common clock that defines the time-unit of the NoC, called a time-slot [20]. To efficiently use network resources and links, we adopt the technique of time division multiplexing. Messages are emitted repeatedly based on a temporal cycle of length T, and the message size defines a given bandwidth between a source and a destination nodes. At each source node, a TDMA (time division multiple access) table specifies the moments in time when to transmit messages toward their respective destinations. This technique of time-multiplexing in a NoC was previously presented and used in [16, 15, 7, 10, 13, 19].

Given a graph that represents a network topology, and a set of K source-destination messages of variable sizes that are periodically emitted based on a temporal cycle of length T, the standard integer linear programming problem had been modeled in [20]. This model addresses the computation of K source-destination paths according to the message emission dates, such that messages are transmitted across the network with no possible conflict between packets. The problem goal is to minimize the paths total length (network latency), while using network resources in the most efficient way to guarantee a given bandwidth. In this paper, we extend the standard model in order to allow dynamic reconfiguration of traffic bandwidth, at execution time. We assume that optimization takes place at design time in both cases, and that dynamic reconfiguration is achieved by the interchange at execution time of a set of predefined TDMAs assigned to some of the source nodes. Each TDMA specifies a given set of origin-destination messages and their related paths. Because the problem looks new, we will carefully state it and detail the terminology used in the next section.

This problem of routing in a NoC has some analogies with routing problems in terrestrial transportation networks, as for example with Automated Guided Vehicles (AGVs) [17] technology for optimizing large scale production and logistic systems. Such AGVs are used for transportation tasks. According to [17], the task of the AGVs can be to transport containers between bridges for loading and unloading ships and a number of container storage areas. A key feature is also to find efficient conflict- free routing schemes that avoid collisions, deadlocks and livelocks. We also retrieve the GT and BE routing philosophies in such networks. While in AGVs, synchronization depends on vehicle speed and timewindows specified on network links, message packets in a NoC cross the network links at constant speed of a packet by time-slot. A message can be seen as a “train” of packets, each one occupying a given link in the network in sequential order, and preceded by a header packet which specifies the path to follow.

In Section 2, we state the problem, and give the notations that will be used through this study. In section 3, we present the model using of multiple TDMAs for dynamic routing reconfiguration as a integer linear programming, while reusing time-slots without conflicts. Section 4 is devoted to relate model with previous work. Then, the last section concludes the paper.

2. DEFINITION OF THE RECONFIGURABLE ROUTING PROBLEM

In this section, we first present reconfigurable NoC structure and notations will be use. In the second subsection, we state the reconfigurable routing problem.

2.1 Reconfigurable NoC structure and notations

First, we briefly present the structure of a network- on-chip as illustrated in Fig. 1. NoC architecture consists of a set of interconnected Intellectual Property (IP) components. Such IP components can be general-purpose processors, DSP blocks, memory blocks and embedded reconfiguration modules. These components are connected by routers according to a given network topology. The NoC in Figure 1 is composed of 15 routers, denoted from R_0 to R_{14}. They are interconnected by the arrows in the figure. To the routers, are connected 10 IP modules? They correspond to processing units or memory units. They are denoted IP_0, **...,** IP_9 ***in*** the figure, with specificity for IP_0, IP_1 and IP_2. Such IP modules are transmitters and/or receptors of messages, they are called "terminals”, and also source and destination nodes. These IP units are connected to the network by a standard network interface (NI). To a given IP, is exclusively attached a single router by its NI. Next in the paper, we will simply refer to source and destination nodes in the network.

A NoC can be modeled by a directed graph (V, A) where the set V of vertices represent the routers and IPs, and the set of arcs $V X V$ represent the directed transmission links between them. Here, we consider wormhole routing.

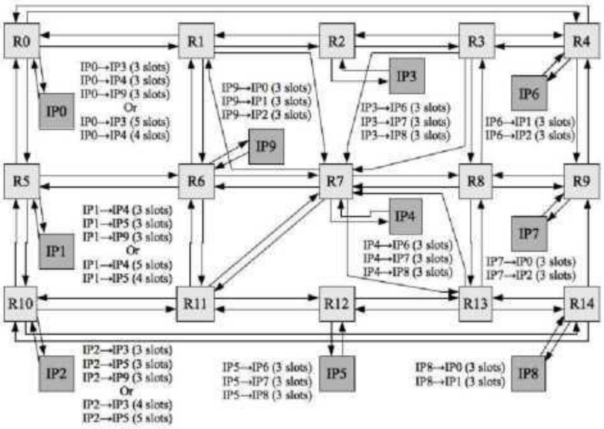

Figure 1: Example of reconfigurable NoC

Each message μ is made of a sequence of contiguous packets, with > 0, that are transmitted along the arcs of the network. The header packet contains a specification of the message origin/destination path, while other packets the application data. A router has no memorization capacity; it only retransmits packets as specified by the header packet. The NoC is synchronous; it is cadenced by a common clock shared by all its components. Each arc has a capacity of 1 packet by unit of time, called a time-slot or time-step. The packets are transmitted in a contiguous way. Hence, if a physical occurrence of a message emitted at time follows the path i, $i+1,\ldots, i+n$, with $i \in V$, its packets will cross the arc $(i, i+1)$ at the consecutive time-steps $+ i + q$, $q = 0,\ldots,$ $- 1$. We say that an arc may be “occupied” by a packet at a given time-slot, otherwise it is said “free” at a given time-slot.

To guarantee transfer rate, a source repeatedly emits messages based on a period or cycle T. A given bandwidth is specified by a message size, ***i.e.,*** the number of packets emitted by period T. Hence, a message can be seen as a class of its physical occurrences at each cycle.

This allows considering classes of time-slots for arc occupations. If an arc is occupied at time-slot t, it will also be occupied at time-slots $t+\lambda T$, for all $\lambda > 0$. We say that an arc is occupied at t modulo T to express its recurrent occupation by a packet. We will often talk of a packet, or message, to refer to the class it represents.

When transmitted through the network, two packets are said conflict-free if they never clash, ***i. e.,*** cross an arc at the same time-slot. By extension, paths or messages are said “conflict-free” or “contention-free” when their related packets never conflict (two at a time).

Since paths may share common communication links, and since we want to achieve guaranteed traffic, paths and emission dates have to be stated such that two packets will never clash. We assume that paths may contain loops when necessary.

The messages managed by a transmitter and sent during a given cycle λT, $\lambda > 0$, are specified within a TDMA (time division multiple access) table of size . A TDMA δ can be seen as a specification of a set of messages together with their related emission dates into the interval [0, ***T*** - 1]. The message set is specified by a set of triples = $\{(s, d_k, l_k): d_k \in V, l_k > 0,\ k = 1, \dots, K_\delta\}$, where V is the source node of messages with target nodes and packets each. Because of cyclic and sequential emission, we must have $\leq \sum l_k \leq T$. Given two messages with their respective emission dates , $[0, T-1]$, and respective number of packets , , such that t_2 , we necessarily have $l_1 < t_2$, and $+ l_2) mod\ T \leq t_1$ whenever $+ l_2 \geq T$. This means that at most ***T*** packets are sent within a cycle of length ***T*** with their emission dates arbitrarily stated in $-1]$.

We say that the system is pipeline-based. About the receiver nodes, it is also true that at most ***T*** different packets can be received within a cycle T. This can be verified in the example of Figure 1. With an emission cycle of ***T*** = 9 time-slots, IP_0, IP_1 and IP_2 sources emit three messages with three packets each or two messages with five and four packets each. Whereas IPs, IP_4, IP_5, IP_9 emit three messages with three message each and IP_6, IP_j, IP_8 emit two messages with three packets each. The TDMA's specifications are beside each source node in the figure. One could verify that each IP receiver never receives more than 9 packets from the different sources.

2.2 Problem statement

In the integer linear programming modeling the standard problem considered in [20] we assume that each source node emits messages according to a single TDMA table. In this problem of reconfigurable routing, we assume that a set of TDMA tables can be assigned to a each IP source node, such that the IP's can interchange their applications at any cycle λT, $\lambda \geq 0$ asynchronously and independently from one each other. The decision is local; a source node may interchange its emission table at any cycle, individually and without regards to what other source nodes do. This is illustrated in Fig. 2. In (a) are shown TDMA tables assigned to IP components. In (b), are illustrated the possible interchanges of TDMA emissions at each cycle. In this figure, IP i have three TDMA and IP k have two TDMA. $_{ij}$ being the maximal number of packets that IP_j can receive from IP_i. Note that the messages of the new table at cycle $- 1)T$ must be sent once all the packets from the previous table have been emitted. It should be the case that the remaining packets of the last emitted message at cycle are set after the time $-(T+1)$, which is the latest possible emission date for the header packet.

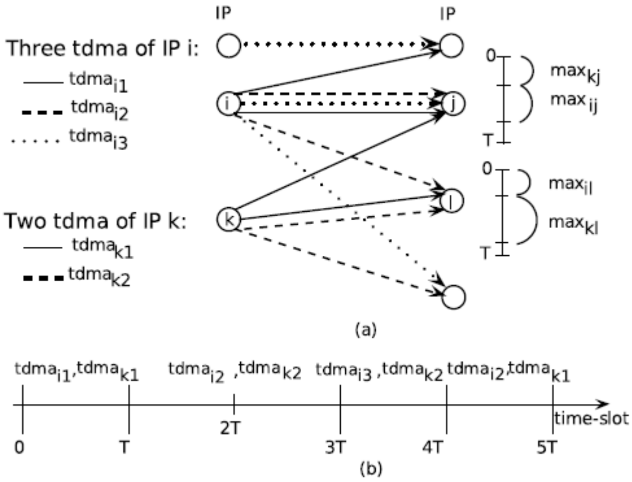

Figure 2: Reconfigurable routing, (a) Emitters may have multiple TDMA configurations, (b) Emitters can interchanged TDMA emissions at any cycle *T*

We distinguish different cases of possible conflicts. These conflicts can be inter-message conflicts that arise between packets of distinct messages and intra-message conflicts between packets of a same message class. These conflicts could arise at a given emission cycle, or between different emission cycles depending on the path lengths along which packets travel. The example in Fig. 3 illustrates conflict-free message transfers in the NoC with possibly dynamic TDMA interchanges. We can see how the origin/destination paths may be modified dynamically and the vacant time-slots reused. We can now define the reconfigurable routing problem.

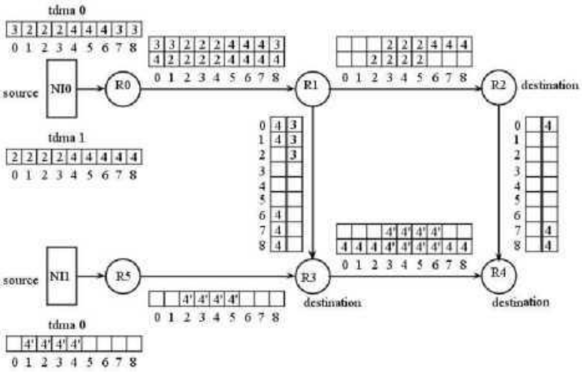

Figure 3: Message transfers in a NoC with multiple TDMA configurations

Reconfigurable routing problem

A problem instance consists in a directed graph ***G = (V, A),*** an emission cycle of length T, and a finite set of TDMA tables. A given source node s can have one or more tables attached to it that may be interchanged at each cycle to be emitted. A TDMA table δ attached to a source nodes is specified by its related message set $= \{(s, d_k, l_k): d_k \in V, l_k > 0,\ k = 1, \dots, K_\delta\}$, where $_k)$ is an origin/destination pair, and the number of packets. The goal is to find the departure time of each message within the interval $' - 1]$ and the origin/destination paths of minimum total length for conveying the messages, such that the paths are conflict-free whichever the TDMA table that is emitted at each cycle.

We should note that the objective of the problem is the total length of the paths, whereas the constraints reside in the temporal occupation of the arcs such that paths are conflict-free. The specificity of the reconfigurable routing problem is to allow the interchanged of multiple tables attached to each source node.

3. RECONFIGURABLE ROUTING INTEGER LINEAR PROGRAMMING MODEL

The model is packet oriented since the routing concerns the packets of message. A NoC is composed of IPs, routers, and packets travel the NoC across arcs. There are ***P*** IPs that may contain ρ TDMA, each TDMA denoted δ such as $1, \dots \rho\}$. ***P*** IPs and ***N*** routers denoted ***i*** or ***j***. The set of IPs and routers is $\{0, \dots, P-1, P, \dots, P+N-1\}$. The directed arcs are denoted ***(i, j)***. The set of arcs is denoted $VX\,V$. For every IP source, there are packets by TDMA δ denoted ***q*** with $1, \dots, Q_\delta\}$. Where $= \sum_{k=1}^{K_\delta} l_k$, with the size of message of the TDMA and the total number of messages of the TDMA such as $1, \dots, K_\delta\}$. Each packet has and origin IP s ∈ {0, …, ***P*** - 1} and a destination IP ***d*** ∈ {0,..., ***P*** - 1} with s ≠ d. Time is discretized, ***t*** ∈ {0, ..., ***T*** - 1} is a time slot, and ***T*** is the TDMA size. The system is pipeline-based, thus, time slots ***t*** and ***t*** + ***T*** are identical for all t. Let $_{,\delta,q,t}$ be a Boolean variable that is set to one if packet ***q*** of the TDMA in the source ***s*** uses arc ***(i, j)*** during time slot t, it is set to 0 otherwise. We can now state constraints and objective, for NoC can have a maximum of two TDMA by IP, as a integer linear program in equations (1-13) below.

1. Arc capacity when s ≠ *s'* : Each arc is used for conveying at most one packet per unit of time.

$$\sum_{q\in\{1,\dots,Q_\delta\}} x_{i,j,s,\delta,q,t} + \sum_{q'\in\{1,\dots,Q_{\delta'}\}} x_{i,j,s',\delta',q',t} \le 1$$

$$\begin{gathered}\forall (i,j) \in A,\\ \forall t \in \{0, \dots, T-1\},\\ \forall s \in \{0, \dots, P-1\},\\ \forall s' \in \{0, \dots, P-1\},\\ \forall \delta \in \{0, \dots, \rho_s - 1\},\\ \forall \delta' \in \{0, \dots, \rho_{s'} - 1\}\end{gathered} \tag{1}$$

2. Arc capacity when s = s'**:** Each arc is used for conveying at most packet(s) per unit of time.

$$\sum_{q\in\{1,\dots,Q_\delta\}} x_{i,j,s,\delta,q,t} + \sum_{q'\in\{1,\dots,Q_{\delta'}\}} x_{i,j,s',\delta',q',t} \le \rho_s$$

$$\begin{gathered}\forall (i,j) \in A,\\ \forall t \in \{0, \dots, T-1\},\\ \forall s \in \{0, \dots, P-1\},\\ \forall \delta \in \{0, \dots, \rho_s - 1\},\\ \forall \delta' \in \{0, \dots, \rho_s - 1\} - \{\delta\}\end{gathered} \tag{2}$$

3. Packet origin: For all $1, \dots, Q_\delta\}$, packet originates from his IP s∈$.., P-1\}$.

$$\sum_{\substack{t\in\{0,\dots,T-1\}\\ j\in\{P,\dots,P+N-1\}|(s,j)\in A}} x_{i,j,s,\delta,q,t} = 1$$

$$\begin{gathered}\forall q \in \{1, \dots, Q_\delta\},\\ \forall s \in \{0, \dots, P-1\},\\ \forall \delta \in \{0, \dots, \rho_s - 1\}\end{gathered} \tag{3}$$

4. Packet destination: For all $q \in \{1, \dots, Q_\delta\}$, packet ***q*** has to reach his IP d∈$\{0, \dots, P-1\}$.

$$\sum_{\substack{t\in\{0,\dots,T-1\}\\ i\in\{P,\dots,P+N-1\}|(i,d)\in A}} x_{i,j,s,\delta,q,t} = 1$$

$$\begin{gathered}\forall q \in \{1, \dots, Q_\delta\},\\ \forall s \in \{0, \dots, P-1\},\\ \forall \delta \in \{0, \dots, \rho_s - 1\}\end{gathered} \tag{4}$$

5. Packets conservation: For all instant time $), \dots, T-1\}$, for all $1, \dots, Q_\delta\}$ and for $P, \dots, P+N-1\}$, if packet reaches router at time , then it has to leave that router at time 1 .

$$\sum_{j\in V|(j,i)\in A} x_{i,j,s,\delta,q,t} + \sum_{j\in V|(i,j)\in A} x_{i,j,s,\delta,q,(t+1) \bmod T}$$

$$\begin{gathered}\forall t \in \{0, \dots, T-1\},\\ \forall q \in \{1, \dots, Q_\delta\},\\ \forall i \in \{P, \dots, P+N-1\},\\ \forall s \in \{0, \dots, P-1\},\\ \forall \delta \in \{0, \dots, \rho_s - 1\}\end{gathered} \tag{5}$$

6. Origin IPs are not reentrant: For all instant time $t \in \{0, \dots, T-1\}$, for all $q \in \{1, \dots, Q\}$, packet q cannot enter in his IP $s \in \{0, \dots, P-1\}$.

$$\sum_{i\in\{1,\dots,P+N-1\}|(i,s)\in A} x_{i,j,s,\delta,q,t} = 0$$

$$\begin{gathered}\forall t \in \{0, \dots, T-1\},\\ \forall q \in \{1, \dots, Q_\delta\},\\ \forall s \in \{0, \dots, P-1\},\end{gathered} \tag{6}$$

$$\forall\delta\in\{0,\dots,\rho_s-1\}$$

7. Destination IPs are not reentrant: For all time slot $,\dots,T-1\}$, for all $1,\dots,Q\}$, packet cannot leave his IP $0,\dots,P-1\}$.

$$\sum_{j\in\{P,\dots,P+N-1\}|(d,j)\in A} x_{i,j,s,\delta,q,t} = 0$$

$$\forall t\in\{0,\dots,T-1\},\quad \forall q\in\{1,\dots,Q_\delta\},\quad \forall s\in\{0,\dots,P-1\},\quad \forall\delta\in\{0,\dots,\rho_s-1\} \qquad (7)$$

8. Origin IPs generate at most one packet at a time: For all time slot $0,\dots,T-1\}$, for all IP i$,\dots,P-1\}$, the sum of the packet sent out of and such that i must be less than .

$$\sum_{\substack{q\in\{1,\dots,Q_\delta\}|s=i\\ j\in\{P,\dots,P+N-1\}|(i,j)\in A}} x_{i,j,s,\delta,q,t} + \sum_{\substack{q'\in\{1,\dots,Q_{\delta'}\}|s=i\\ j\in\{P,\dots,P+N-1\}|(i,j)\in A}} x_{i,j,s',\delta',q',t} \le \rho_s$$

$$\forall t\in\{0,\dots,T-1\},\quad \forall i\in\{0,\dots,P-1\},\quad \forall s\in\{0,\dots,P-1\},\quad \forall\delta\in\{0,\dots,\rho_s-1\},\quad \forall\delta'\in\{0,\dots,\rho_s-1\}-\{\delta\} \qquad (8)$$

9. Destination IPs consume at most one packet at a time when $'$: For all time slot $),\dots,T-1\}$, for all IP j$,\dots,P-1\}$ the sum of the packets sent to IP and such his destination j must be less than one.

$$\sum_{\substack{q\in\{1,\dots,Q_\delta\}|d=j\\ i\in\{P,\dots,P+N-1\}|(i,j)\in A}} x_{i,j,s,\delta,q,t} + \sum_{\substack{q'\in\{1,\dots,Q_\delta\}|d'=j\\ i\in\{P,\dots,P+N-1\}|(i,j)\in A}} x_{i,j,s',\delta',q',t} \le 1$$

$$\forall t\in\{0,\dots,T-1\},\quad \forall j\in\{0,\dots,P-1\},\quad \forall s\in\{0,\dots,P-1\},\quad \forall s'\in\{0,\dots,P-1\},\quad \forall\delta\in\{0,\dots,\rho_s-1\},\quad \in\{0,\dots,\rho_{s'}-1\} \qquad (9)$$

10. Destination IPs consume at most one packet at a time when s' : For all time slot $),\dots,T-1\}$, for all IP j$,\dots,P-1\}$ the sum of the packets sent to IP and such his destination j must be less than .

$$\sum_{\substack{q\in\{1,\dots,Q_\delta\}|d=j\\ i\in\{P,\dots,P+N-1\}|(i,j)\in A}} x_{i,j,s,\delta,q,t} + \sum_{\substack{q'\in\{1,\dots,Q_{\delta'}\}|d'=j\\ i\in\{P,\dots,P+N-1\}|(i,j)\in A}} x_{i,j,s',\delta',q',t} \le \rho_s$$

$$\forall t\in\{0,\dots,T-1\},\quad \forall j\in\{0,\dots,P-1\},\quad \forall s\in\{0,\dots,P-1\},\quad \forall\delta\in\{0,\dots,\rho_s-1\},\quad \in\{0,\dots,\rho_s-1\}-\{\delta\} \qquad (10)$$

11. Message constraints: A message is modeled as a sequence of packets that must have the same route in the NoC. More formally, message $1,\dots,K_\delta\}$ of TDMA δ in source s is an ordered set of packets $_1,\dots,\omega_{k,l(k)}\}$ where is the length of message of the TDMA δ in source ***(i.e.,*** the number of packets in the message). $\in\{1,\dots,Q\}$ is the first packet and $_{k)}$ is the last packet of message of the TDMA δ in source s **.** In any consistent instance, all the packets of a message must have the same origin and destination router, ***i.e.,*** they must satisfy: $= s_{q'}$ and $= d_{q'}$ $\in\{1,\dots,K_\delta\}$, $q'\{\omega_{k,1},\dots,\omega_{k,l(k)}\}$

The message constraints are enforced as follows:

$$_{,\delta,\omega_{k,q},t} = x_{i,j,s,\delta,\omega_{k,q+1},(t+1)\bmod T}$$

$$\forall k\in\{1,\dots,K_\delta\},\quad \forall q\in\{1,\dots,l(k)-1\},\quad \forall t\in\{0,\dots,T-1\},\quad \forall(i,j)\in A,\quad \forall s\in\{0,\dots,P-1\},\quad \forall\delta\in\{0,\dots,\rho_s-1\} \qquad (11)$$

These constraints say that if packet of the TDMA δ in source occupies arc at time , then packet $_{+1}$ of the TDMA δ in source s must occupy this arc at time $1)\bmod T$. The equality also enforces that if packet of the TDMA δ in source s does not occupy arc at time , then packet $_{+1}$ of the TDMA δ in source s cannot occupy this arc at $1)\bmod T$. Thus, all the packets of a message move along the same route within a time shift.

12. Packet destination: The destination of packet is a terminal node of the path of the packet . For all time slot $),\dots,T-1\}$, for all $1,\dots,Q_\delta\}$, the packet cannot go to another destination other than its destination .

$$\sum_{\substack{j\in\{0,\dots,P-1\}-\{d\}\\ (i,j)\in A}} x_{i,j,s,\delta,q,t} = 0$$

$$\forall t\in\{0,\dots,T-1\},\quad \forall q\in\{1,\dots,Q_\delta\},\quad \forall i\in\{P,\dots,P+N-1\},\quad \forall s\in\{0,\dots,P-1\},\quad \forall\delta\in\{0,\dots,\rho_s-1\} \qquad (12)$$

13. Objective: Minimize the total length of all paths for all packets.

$$\sum_{s} \sum_{\delta} \sum_{(i,j)\in A} \sum_{q\in Q_{\delta}} \sum_{t\in T} x_{i,j,s,\delta q,t}$$

Constraints 8, 9 and 10 are specifically added to take into account the hardware conditions of emission and reception. These conditions state that each IP node implements TDMAs (Time Division Multiple Access) table that can only send one packet at each time slot occurrence. A router node, at the difference of an IP node, allows packets to cross at a given time slot following the different directions stated by the communication graph. An IP node will always be connected to a single router and its emission capability restricted by the TDMA mechanism such that no more than one packet will be emitted or received at each time slot.

4. RELATED WORK

In this section, we relate the cyclic reconfigurable ***K***- conflict-free paths problem to other standard problems from the literature in order to gauge its computational complexity. The cyclic K-conflict-free paths problem (CKPP) was presented first in [15, 6, 3], but it was stated informally and presented in the context of MPSoC technology. The cyclic reconfigurable K-conflict-free paths was stated formally as a combinatorial optimization problem first in [20] Here, we adopt a combinatorial optimization viewpoint through a integer linear programming model.

We should note that the objective of the problem is the total length of the paths, whereas the constraints reside in the temporal occupation of the arcs such that paths are conflict-free. It is worth noting that the problem is NP-hard in the strong sense for general graphs as well as for planar graphs or grid networks. This can be seen by relating the problem to the ***K***-Edge- Disjoint Shortest Paths problem (EDP) [13]. This well known problem has different versions depending whether we ask for vertex or edge disjoint paths between a set of ***K*** source-sink pairs.

This is one of Karp's [12] original NP-hard problems. By restricting CKPP to only those instances for which 0 , we retrieve the EDP. This problem is known to be NP-hard in the case of planar graphs [14], even when stated in the grid and when path lengths are constrained by a constant [8]. Then, since CKPP remains NP-hard when the number is bounded by a constant, it follows that it is NP- hard in the strong sense. As well, the CKPP can be seen as an extension of a Bin Packing problem, where arcs stand for bins of capacity , and the messages for items of size .

A similar problem is the unsplittable flow problem (UFP). It is a generalization of EDP where every edge has a positive capacity ; and every pair has a demand 0 . The demand from two has to be routed in an unsplittable manner, ***i.e.,*** along a single path from two For every edge the total demand routed through that edge should be at most . The problem adds a capacity constraint to the EDP. It is different from CKPP since it allows variable arc capacities, discarding temporal aspects on arc occupation. Generally, classical flow models only deal with static situation. A well-known problem that introduces time dependent transit constraints is the one-to-one shortest path problem with time windows (SPPTW) [5]. The aim is to compute a shortest path respecting the given time-windows on arcs occupation. This kind of problem often arises in terrestrial transportation, road traffic control, and vehicle routing applications. An example is Automated Guided Vehicles (AGVs) [17] technology for optimizing large scale production and logistic systems. The SPPTW is NP-hard but several pseudo-polynomial time algorithms are available to solve it exactly [5, 11, 18]. A sub-problem of our CKPP is to compute a one-to-one source-destination path in an already occupied NoC. The available ("free") time-slots on each arc stand for time-windows

5. CONCLUSION

We have presented a integer linear programming model of combinatorial optimization problem which models reconfigurable routing in a network-on-chip with guaranteed traffic. Based on TDMA techniques, the problem can be seen as an extension of a classical K-shortest paths problem combined with a bin packing problem, where time plays an important role. Messages are emitted in a cyclic way and the problem goal mainly resides in the efficient allocation of the available timeslots for conflict-free routing, while minimizing path lengths. This problem takes into account the possibility of reconfigurable routing at execution time. It becomes unnecessary to stop the system while modifying the traffic bandwidth between nodes. The method performs multi-path computation in a NoC, taking care of asynchronous TDMA's interchanges at runtime.

REFERENCES

[1] Agarwal A, Iskander C, Shankar R (2009) Survey of network on chip (NoC) architectures and contributions. Journal of Engineering, Computing, and Architecture 3(1):1-15

[2] Benini L, Micheli G (2002) Networks on Chips: A New SoC Paradigm. Computer 35(1):70-78

[3] Dafali R (2011) Conception des réseaux sur puce reconfigurables dynamiquement. Phd dissertation, University of South Brittany, France

[4] Dally W, Towles B (2001) Route packets, not wires: On-chip interconnection networks. In: Proceedings of the 38th Design Automation Conference

[5] Desrochers M, Soumis F (1988) A generalized permanent labelling algorithm for the shortest path problem with time windows. INFOR 26:191-212

[6] Evain S (2006) µSpider Environnement de

Conception de Reseau sur Puce. Phd dissertation, Institut National des Sciences Appliquees de Rennes, France

[7] Evain S, Diguet JP (2007). Efficient space-time NOC path allocation based on mutual exclusion and prereservation. In: Proceedings of the 17th ACM Great Lakes symposium on VLSI, Italy

[8] Gonzalez TF, Serena D (2004) Complexity of pairwise shortest path routing in the grid. Theoretical Computer Science 326:155-185

[9] Goossens K, van Meerbergen J, Peeters A, Wielage R (2002) Networks on Silicon: Combining Best-Effort and Guaranteed Services. In: Automation and Test in Europe Conference and Exhibition, pp 423-425

[10] Hansson A, Goossens K, Radulescu A (2007) A unified approach to constrained mapping and routing on network-on-chip architectures. In: CODES+ISSS'05 Proceedings of the 3rd IEEE/ACM/IFIP international conference on Hardware/software code sign and system synthesis, pp 75-80

[11] Ioachim I, Geiinas S, Soumis F, Desrosiers J (1998) A dynamic programming algorithm for the shortest path problem with time windows and linear node costs. Networks 31(3):193204

[12] Karp RM (1975) On the computational complexity of combinatorial problems. Networks 5:45-68

[13] Kolliopoulos SG (2007) Edge-disjoint paths and unsplittable flow. In: T. F. Gonzalez (ed) Handbook of Approximation Algorithms and Metaheuristics, Chapman & Hall/CRC

[13] Lu Z, Jantsch A (2008) TDM virtual-circuit configuration for network-on-chip. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 16(8):1021-1034

[14] Lynch JF (1975) The equivalence of theorem proving and the interconnection problem. ACM SIGDA Newsletter 5:31-36

[15] Marescaux T, Bricke B, Debacker P, Nollet V, Corpo- raal H (2005) Dynamic time-slot allocation for QoS enabled networks on chip. In: 3rd Workshop on Embedded Systems for Real-Time Multimedia, pp 47-52

[16] Millberg M, Nilsson E, Thid R, Jantsch A (2004) Guaranteed bandwidth using looped containers in temporally disjoint networks within the Nostrum network on chip. In: Proceedings of the conference on Design, automation and test in Europe

[17] Mohring RH, Kohler E, Gawrilow E, Stenzel B (2004) Conflict-free Real-time AGV Routing. In: Operations Research Proceedings, pp 18-24

[18] Powell WB, Chen ZL (1998) A Generalized Threshold Algorithm for the Shortest Path Problem with Time Windows. In: DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol 40

[19] Stefan R, Goossens K (2011) Enhancing the security of time-division-multiplexing networks-on-chip through the use of multipath routing. In: Proceedings of the 4th International Workshop on Network on Chip Architectures, ACM New York

[20] Zerbo B (2012) Problèmes de plus courts chemins dans les NoC et leurs extensions aux cas difficiles. PhD thesis, Université de Bretagne Sud – Université de Ouaga