# Dynamic Routing: A Prerequisite for Reliable NoCs

Muhammad Ali, Michael Welzl
Institute of Computer Science
University of Innsbruck, Austria
{Muhammad.Ali, Michael.Welzl}@uibk.ac.at

Sybille Hellebrand
Institute of Electrical Engineering
University of Paderborn, Germany
Sybille.Hellebrand@date.upb.de

## Abstract

*With an increase in the number of transistors on-chip, the complexity of the system also increases. In order to cope with the growing interconnect infrastructure, the "Network on chip (NoC)" concept was introduced. With network methodologies coming on-chip, various characteristics of traditional networks come into play. So far, failures that are common in regular networks were hardly considered on-chip; this paper introduces ideas of dynamic routing in the context of NoCs and explains how they could be applied to cope with adverse physical effects of deep sub-micron technology.*

**Keywords:** NoC, fault-tolerant, self-healing, routing

## 1. Introduction

Recent advances in technology are driving the current silicon dies to accommodate billions of transistors in the near future leading to designs of very complex Systems-on-Chip (SoCs). However, a study of current SoC infrastructure reveals that buses – existing interconnect medium for SoCs – are posing a serious threat toward achieving the era of billion transistors because of their poor scalability beyond a certain number of partners [1]. To counter such a situation, researchers from the chip design area have explored the computer networks area which has already dealt with issues such as scalability (e.g, Internet technology scales very well).

The idea of NoCs has been exploited by many researchers and many design templates have been introduced for this novel interconnect structure for SoCs. This paper focuses on a major area regarding the communication on-chip; that is, how to deal with broken links or/and malfunctioning routers on chip. To the best of our knowledge, this area is not yet addressed by the research community which is probably due to the assumption that networks on chip are considered to be stable and, hence, no such failures are supposed to occur [2]. We however believe that, as the number of transistors on a chip increases, the problems associated with deep sub-micron will become more pronounced and may therefore pose problems of link and/or router failures [3]. In order to cope with link failures we propose techniques for dynamic routing in NoCs which are well known to the network community.

## 2 Routing in NoCs

In the Internet, packet based communication was introduced to replace the circuit switching in order to bring reliability to communication. However, time proved that it played a vital role in the scalability of the Internet. Packet based communication has been brought to NoCs but looses the original advantage of reliability in the absence of dynamic routing as is being implemented in the Internet [4]. Currently, most of the proposals for routing in NoCs are based upon static routing mechanisms which are based upon a XY-coordinate discipline [2], [5]. This is a simple mechanism but in case of link failures, a mechanism is needed to route packets dynamically. Dynamic routing, as the name shows, is used to dynamically discover routes in case of path changes. Figure 1 shows the basic concept of routing based on shortest path selection.

In the following we describe the two most popular dynamic routing algorithms in networks and discuss how they can be utilized in case of NoCs.

### 2.1 Distance Vector Routing

This is quite a simple routing mechanism where each router maintains a table about the known destinations available to it along with the link to get there, and it sends this information to its neighbours. In this way distance vector routing keeps information about neighbours and destinations only, which makes it a simple mechanism (e.g., node A in figure 1 has no knowledge about the link between C and D). However, distance vector routing may lead to a counting-to-infinity problem where packets keep on bouncing between two nodes infinitely in case of a broken link.

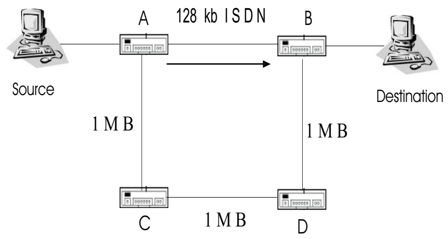


**Figure 1. Shortest Path Routing**

Some solutions for this problem are also suitable for NoCs. Examples are "split horizon with poisoned reverse" limiting the maximum count (e.g., the "Routing Information Protocol" defines the maximum count as 16), and triggered updates [4], [6].

### 2.2 Link State Routing

In contrast to distance vector routing, which shares its routing table with immediate neighbours only, the link state routing algorithm tells every router on the network about its neighbour routers. The mechanism is based upon "a distributed map", where each router has a copy of the map and it is regularly updated. The goal of link state routing is for each peer to have an identical picture of the state of the entire network; link state protocols require each router to know whether a link is up or down and which cost it has, and then calculate the total cost to reach a destination.

However, in an NoC context, the complexity of link state routing can be simplified by implementing static routing tables. These tables will never grow based on the fact that no new nodes will be added once the chip is off the factory. In such a case, when a link goes down, only relevant entries need to be removed from the tables. Also there are no manual interventions to change the topology of the NoCs as in case of regular networks. Hence, this way we can counter the inherent problem of scalability of link state routing and can make it a feasible solution for NoCs.

The simplified form of link state routing that suits for NoCs is as follows:

1. The network configuration can be set in the beginning, as the topology, distance between the nodes and their number are known in advance. The cost of each neighbour can be calculated in the beginning. Based on this information, routing tables are built and stored in every router.

2. Whenever there is any significant change in the network – a link is down or a router malfunctions – ECHO packets can be sent to calculate the cost toward the alternate paths and the network configuration can be changed accordingly; otherwise, routing tables will be built using the last known configuration. There is no need to frequently send the updates over the network.

3. In case of a topology change, best paths can be calculated using Dijkstra's "Shortest Path First" algorithm.

## 3 Conclusion

In this paper we introduced the idea of dynamic routing in NoCs. We have argued that, in order to produce fault-tolerant future SoCs, it would be inevitable to incorporate routing mechanisms on a chip. This fact is underlined by the increasing physical effects of deep sub-micron technology as a chip scales in size.

In this paper, we have discussed two popular dynamic routing algorithm classes; distance vector routing and link state routing with their pros and cons. In the context of NoCs, distance vector routing seems to be a better choice because of its simplicity and presumably low power consumption. However, link state routing may become a much better choice by introducing static routing tables.

We hope that our ideas will not only benefit the NoC designers to come up with fault tolerant designs but may also inspire the network research community to actively contribute their ideas.

## References

[1] Luca Benini and Giovanni De Micheli, "Networks on Chips: A New SoC Paradigm", *DATE, Design, Automation and Test in Europe, Paris, March 3 - 7, 2002*

[2] Bart Vermeulen, John Dielissen, Kees Goossens, Kalin Ciordas, "Bringing Communication Networks On Chip: Test and Verification Implications", *IEEE Communications Magazine, Vol. 41, No. 9, September 2003, pp. 74-81*

[3] R. Ho, K. Mai, and M. Horowitz, "The Future of Wires", *Proceedings of the IEEE, Vol. 89, No. 4, April 2001, pp 490 - 504*

[4] Christian Huitema, "Routing in the Internet", *Second Edition, 2000, Prentice Hall, New Jersey*

[5] Cesar A. Zeferino, Marcio E. Kreutz, Luigi Carro, Altamiro A. Susin, "A Study on Communication Issues for System-on-chip", *Proceedings of 15th Symposium on Integrated Circuits and System Design, Porto Alegre, Brazil, 2002, pp. 121-126*

[6] C. Hendrick, "Routing Information Protocol", *RFC 1058, IETF Network Working Group, June 1988. Category: Historical*