A Communication-Aware Topological Mapping Technique for NoCs*

Rafael Tornero[1], Juan M. Orduña[1], Maurizio Palesi[2], and José Duato[3]

[1] Departamento de Informática - Universidad de Valencia - Spain
Juan.Orduna@uv.es
[2] DIIT - University of Catania - Italy
mpalesi@diit.unict.it
[3] DISCA - Universidad Politécnica de Valencia - Spain
jduato@gap.upv.es

Abstract. Networks–on–Chip (NoCs) have been proposed as a promising solution to the complex on-chip communication problems derived from the increasing number of processor cores. The design of NoCs involves several key issues, being the topological mapping (the mapping of the Intellectual Properties (IPs) to network nodes) one of them. Several proposals have been focused on topological mapping last years, but they require the experimental validation of each mapping considered.

In this paper, we propose a communication-aware topological mapping technique for NoCs. This technique is based on the experimental correlation of the network model with the actual network performance, thus avoiding the need to experimentally evaluate each mapping explored. The evaluation results show that the proposed technique can provide better performance than the currently existing techniques (in terms of both network latency and energy consumption). Additionally, it can be used for both regular and irregular topologies.

1 Introduction

Network-on-Chip (NoC) architectures [1] have been proposed as a promising solution to the complex on-chip communication problems derived from the increasing number of processor cores. In these architectures, each tile of the network contains a resource and a switch. Each switch is connected to a resource (a processor, a memory, or any other Intelectual Property (IP) compatible with the NoC interface specifications) and to some adjacent switches, depending on the NoC topology. The design flow for this architecture involves several steps. First, the application has to be split up into a set of concurrent communicating tasks. Then, the IPs are selected from the IP portfolio and the tasks are assigned and scheduled. Finally, the IPs have to be mapped onto the mesh in such a way that

* This work has been jointly supported by the Spanish MEC, the European Commission FEDER funds, the HiPEAC network of excellence, and the University of Valencia under grants Consolider-Ingenio 2010 CSD2006-00046, TIN2006-15516-C04-04, HiPEAC cluster 1169, and UV-BVSPIE-07-1884.

E. Luque, T. Margalef, and D. Benítez (Eds.): Euro-Par 2008, LNCS 5168, pp. 910–919, 2008.

the considered performance metrics are optimized. Different mapping techniques have been proposed last years [2,3,4], since the final mapping onto the mesh has a strong impact on typical performance metrics and the mapping problem is known to be NP-hard [5].

Some of the proposed techniques uses heuristic methods to search a mapping that optimizes the desired performance metrics (latency, power consumption, quality of service, etc.) with a reasonable computational cost [2]. Unfortunately, they require the experimental validation of each solution provided, making the design process very costly. Additionally, the mapping for NoCs with irregular topology like [6,7] is still an open issue.

In order to provide an efficient mapping technique, the experimental validation of each mapping considered should be avoided. On other hand, since the task scheduling is already performed in a prior step, exclusively a communication-based mapping technique should be used. In this paper, we propose a communication-aware topological mapping technique for NoCs. This technique is based on a task mapping technique that was proposed for irregular networks [8], and it consists on globally matching the communication requirements of the application(s) tasks running on the IPs with the existing network resources. Unlike other mapping techniques, the proposed method is based on correlating the model of network resources with the actual network performance. Since the correlation is high, the model is used to estimate the quality of each assignment. The evaluation results show that the proposed technique can provide better performance than other mapping techniques that require the experimental validation of each mapping considered.

The rest of the paper is organized as follows: Section 2 describes the proposed topological mapping technique, including the model of the NoC resources. Next, Section 3 shows the experimental correlation of the network model with the actual network performance. Section 4 shows the performance evaluation of the proposed technique. Finally, Section 5 concludes the paper and draw some directions for future developments.

2 A New Topological Mapping Technique

During the NoC design process, tasks mapping and tasks scheduling are carried out in order to properly balance the computational requirements of the application with the available resources in the existing IPs. These procedures are beyond the scope of this paper, since they have been addressed in the area of hardware/software co-design and IP-reuse [3,9]. However, after these steps, each IP should contain one (or more) task(s) that can exchange information with another tasks assigned to another IPs, generating a given network traffic pattern (due to one or more applications). The topological mapping (assigning each IP to a network node) should exclusively focus on this pattern. We propose a topological mapping technique that near-optimally fits the communication requirements of the IPs with the available network bandwidth in the different parts of the network. This technique consists of several steps:

first, the available network resources must be modeled. Second, the communication requirements of the tasks running on the different IPs must be modeled. Third, a criterion that measures the suitability of each mapping is needed. This criterion should use the model of the network resources and also the communication requirements of the tasks. Finally, some mapping technique that tries to minimize/maximize the previous criterion should be developed.

In order to model the network resources, we have defined a metric that is based exclusively on the internode distances provided by the routing algorithm, since this metric is in turn inversely related with the amount of resources necessary for communicating two nodes. Concretely, our model proposes a simple metric, the *equivalent distance* between each pair of nodes (in what follows we will refer to a network router as a node). A *table of equivalent distances* can be obtained by computing the equivalent distance between each pair of nodes in the network. As a first approach, we assume that the link bandwidth is the same for all the links in the network. Therefore, we assign the unit cost for each link, but this limitation can be removed if necessary by assigning different link costs to different links. The equivalent distance for a pair of nodes is computed taking into account all the shortest paths between them supplied by the routing algorithm. The name of the metric is derived from the analogy to the electrical equivalent resistance. Indeed, we use the same rules as for electrical circuits to compute the total communication cost between nodes, applying Kirchoff's laws. The reason for using these rules is that the alternative paths increase the available network bandwidth between a pair of nodes, but the length of each path increases the number of nodes that share that path. The model of equivalent resistances properly reflects that situation, and it allows to model the network irregularities by containing different values in the table of distances.

Concretely, in order to compute the equivalent distance between each pair of nodes, the following method is used: if the routing algorithm provides only one shortest path between a given pair of nodes, then the equivalent distance is the sum of the costs of the links that form the path. This case is similar to computing the equivalent resistance of an electrical circuit consisting of serially arranged resistors. We have chosen only the shortest paths because usually NoCs work below the saturation point, and therefore it is not likely to use non-minimal paths. However, this model can be modified to take into account all the possible paths. If there exists more than one shortest path between a given pair of nodes, then the communication cost between them is computed similarly to the electrical equivalent resistance between two points of an electrical circuit, replacing each link in a shortest path with a unit resistor and applying Kirchoff's laws. The paths not supplied by the routing algorithm are not considered. As an example, let us consider the 2-D mesh network topology shown in Figure 1 a), and let us assume that the routing algorithm can provide two different paths for going from node 0 to node 6: 0-1-6 and 0-5-6. In this case, all the paths supplied by the routing algorithm are shortest paths.

Therefore, in order to compute the equivalent distance between nodes 0 and 6, the source and destination nodes must be considered as the V_{cc} and GND

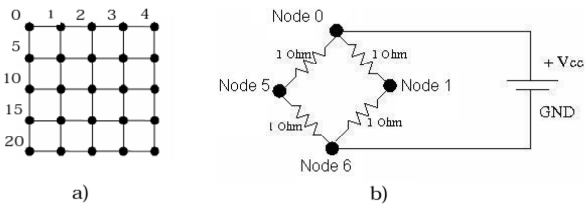

Fig. 1. a) A 2-D network topology b) Equivalent circuit for going from Node 0 to Node 6

points of an electric circuit. Nodes 1 and 5 must be considered as two different intermediate points, and each link must be considered as a resistor with unit resistance,as Figure 1 b) shows. This results in a circuit with two parallel branches, each one composed of two unit resistors serially arranged. Applying Kirchoff's laws to this circuit, an equivalent resistance of 1 Ohm is obtained. Thus, the equivalent distance from node 0 to node 6 would be set to 1. The elements in the table of distances should be computed in this way. If a given element in this table is located at row i and column j ($i, j \in [0..N$-1], where N is the number of rows and columns of the table, that is, the number of network nodes), then we will denote that element as d_{ij}. It represents the cost for communicating node i with node j.

The communication requirements of the tasks running on the different IPs can be estimated by measuring the number of bytes exchanged by the tasks being executed on each IP. Although this estimation can vary due to the existence of hot-spots, the bandwidth required by two given tasks within a given period is related to the number of bytes exchanged between them. Particularly, this estimation is valid for those applications whose communication patterns do not significantly change in different executions with different input data. For example, self-similar traffic has been observed in the bursty traffic between on-chip modules in typical MPEG-2 video applications [10]. Therefore, the proposed technique also computes a table of communications between IPs. If the application(s) tasks are mapped on N IPs, then this table will consist of $N \times N$ elements. We have denoted each element (i, j) in this table as com_{ij}. This value represents the number of bytes that the task(s) mapped onto IP i send(s) to the task(s) mapped onto IP j. These values can be measured from different executions of the application.

Also, we need a quality function to measure the suitability of each mapping. We will represent a given mapping of IPs (hosting one or more tasks) to network nodes as an array of N elements. We will denote each element i of this mapping array as m_i. This value means that IP i is assigned to the network node m_i. Thus, for example, if the second element of this mapping array contains the value 5, this will mean that the IP 1 (if they are numbered from 0 to $N - 1$, or IP 2 if they are numbered from 1 to N) should be located at the network node 5. The quality function is denoted as the *mapping coefficient* M_c, and it is defined as

$$M_c = \sum_{i=1}^{N} \sum_{j=1}^{N} com_{ij} \cdot d_{m_i m_j}$$

where $d_{m_i m_j}$ is the element in the table of equivalent distances in the m_i row and in the m_j column. M_c represents the sum of all the bytes exchanged between the existing tasks, weighted by their corresponding cost of sending these bytes across the network according to the mapping array.

In order to match the communication requirements of the applications with the existing resources in the network, we propose a heuristic search method. In this case, the target function is M_c. If the heuristic search minimizes this function, then the overall cost for transmitting all the application messages will be minimized. The purpose of the proposed mapping technique is to search the best mapping array for a given network topology and for a given communication pattern. Since the communication cost is defined as inversely proportional to network bandwidth [11], we are actually mapping the IPs hosting the tasks that communicate more frequently to the network nodes with the higher network bandwidth between them. By doing so, we fully exploit the existing network resources, delaying network saturation as much as possible. We have tested different heuristic techniques for implementing the search method, including Genetic Algorithms (GA) [12], Simulated Annealing (SA) [13] , and Greedy Randomized Adaptive Search Procedures (GRASP) [14]. However, due to the global nature of the optimization problem, we have obtained the best results with the a random search method. This random search method starts with a random mapping array. Each iteration consists of exchanging two randomly selected values of the mapping array. If the resulting mapping array shows a better mapping coefficient, then this permutation is saved. If not, then it is discarded. The stop condition for the algorithm is to perform a given number of consecutive permutations without decreasing the resulting mapping coefficient. At this point, the minimum reached mapping coefficient and its corresponding mapping array are saved, and another seed (random mapping array) is tried. The algorithm stops when a number of different seeds has been explored. The result of this algorithm is the mapping array with the lowest mapping coefficient reached until that moment. In order to keep diversity, different seeds can be swept.

3 Correlation of the Model of Network Resources

In order to validate the model of network resources, we propose the correlation of the model with actual network performance. Since the quality function M_c uses this model, if the correlation is high then the experimental validation of each mapping will be no longer needed.

For correlation purposes, we have considered a 2-D mesh network topology with X-Y routing and wormhole switching. The table of equivalent distances for that network is computed as described above. We have studied the correlation between each distance in the table of distances and the average latency of the messages exchanged between the corresponding pair of nodes. The performance evaluation methodology used is based on the one proposed in [15].

We have considered different network sizes. However, for the sake of shortness, we show here the correlation results for a 64x64 2-D mesh, since this size is large enough to be representative of the NoCs sizes expected for next years. The correlation results obtained for other network sizes were very similar. We have used different patterns for message generation (uniform, bit reversal, perfect, shuffle, etc). We have considered a fixed packet size of 8 flits. Each router has an input buffer capable of allocating 3 flits. We have made simulations with different injection rates, in order to simulate the network with different workloads (ranging from low load to saturation). The simulator provides the global average latency value obtained for each simulation. This value is computed as the average latency value obtained of all messages transmitted through the network during a given simulation. Additionally, the simulator provides the average latency value for each source-destination pair in the network. From these measurements, we have computed the correlation between the table of distances and network latency.

Figure 2 a) shows the performance evaluation results. This figure shows on the X-axis the traffic injected to the network, measured in packets per cycle and per node. On the Y-axis, it shows the global average message latency. Also, average message latencies for each pair of nodes were computed. However, they are not shown here due to space limitations. Each value in this Figure was computed as the average value of fifty different simulations. Figure 2 a) shows the typical behavior expected for an interconnection network [15]. While the injected traffic is kept below the saturation point, the average latency slightly increases (points $S1$ to $S7$). However, when the injected traffic makes the network to reach saturation (from point $S8$ up) the average latency starts to greatly increase with the traffic rate.

In order to establish the correlation between the table of distances and these performance evaluation results, we have first computed the least square linear adjustment for each point in Figure 2 a). Although they are not shown here due to space limitations, the correlation index was above 85% for the first eight simulation points. From that point up, the correlation coefficient highly

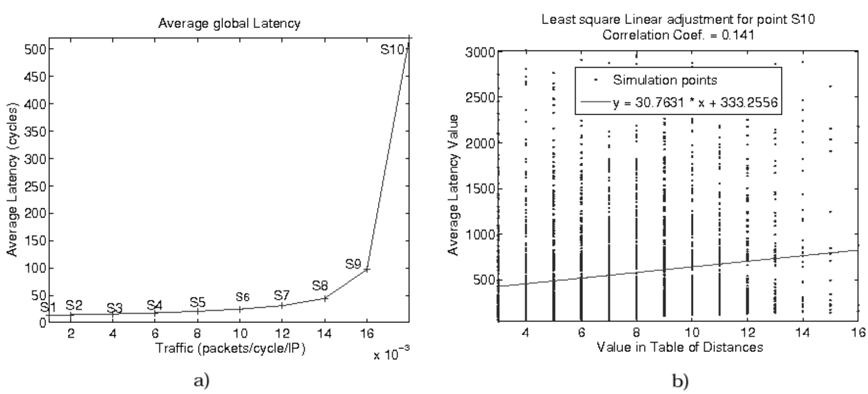

Fig. 2. a) Global performance results for a 8x8 topology b) Correlation for simulation point S1

decreased. For illustrative purposes, Figure 2 b) shows the regression curve for point S10. In this figure, the X-axis shows the values in the table of distance, while the specific latencies obtained by the simulator for each pair of nodes are shown in the Y-axis. The correlation index is 14% in this case, showing that (as it could be expected) when the network is under deep saturation then the contention prevents the network performance to correlate with inter-node distance.

Nevertheless, in order to study the practical correlation of the proposed metric with real performance, when different values are obtained in different simulations (as it is the case) then we should also consider the long-term network behavior. Therefore, we have also studied the correlation between the equivalent distances and the mean values of the average latencies supplied by the simulator. Figure 3 a) shows the corresponding regression curve for the simulation point $S10$. In this figure, there is a point for each different value in the table of distances. These values are represented on the X-axis. For information purposes, the number of occurrences for each value in the table of distances is shown on the right side of the figure. For example, point P4 represents value 6 in the table of distances and appears 552 times in the table of distances. Thus, the long-term network behavior for this value in the table of distances is much more stable than the network behavior for point P12, whose mean value is computed only with 40 different average latencies. As shown in Figure 3 a), the mean values of the average latencies show a much higher correlation with the table of distances (96%) than the average latencies (Figure 2 b), corr. coef. of 14%). The reason for this behavior is that network contention cannot affect this performance metric as it can do to the average latencies.

In order to show the correlation results in a unified way, Figure 3 b) shows the correlation coefficients provided for each of the points shown in Figure 2 a). The correlation COEFFICIENTS for latency values remain about about 86–99% when the network is under a low and a medium load (points $S1$ to $S8$).

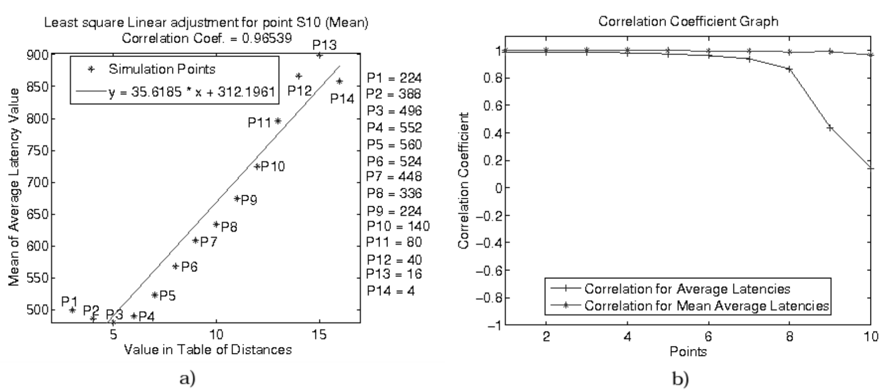

Fig. 3. a) Least square linear adjustment for averages value of average latencies values (point S10) b) Correlation of average latencies values and average values of average latencies for all the points

However, when the network enters deep saturation, the correlation coefficients decrease to values about 40%, and even reaching values of 15%. However, the average values of average latencies obtain a correlation coefficient of about 95% even the network is under a deep saturation, showing that actually the network model properly correlates with the actual network performance.

4 Performance Evaluation

We propose the evaluation of the proposed technique by simulation. For comparison purposes, we have also evaluated the performance of a currently proposed mapping technique based on Genetic Algorithm [2], since to our knowledge this is the mapping technique for NoCs that provide the best performance results (we have denoted this technique as the GA-based technique). Thus, as the application running on the machine we have considered the generic Multi-Media System (MMS). We started from the Communication Task Graph (CTG) of the MMS [2]. From this CTG, that contains 40 different tasks assigned and scheduled onto 25 IPs, we have computed the table of communication requirements (that is, the 40 original tasks are finally grouped in 25 IPs). Concretely, we have simply taken the communication bandwidth required between each pair of processing nodes as the number of messages exchanged between that pair of processing nodes.

Using the local search method described in the previous section, we have obtained a near-optimal mapping of IPs to network nodes. Using the same NoC simulator used for evaluating the GA-based mapping technique [2], we have evaluated the performance of the mapping provided by the proposed mapping technique and also the mapping provided by the GA-based technique. Additionally, for comparison purposes we have evaluated the performance provided by a random mapping as well as the sequential mapping of the MMS.

Figures 4 a) and b) show the comparative results for the MMS application mapped onto a 5x5 mesh. Each point in these figures has been computed as the average value of 50 different simulations. The plots corresponding to the mapping provided by the communication-aware mapping technique have been labeled as "Analytical mapping", while the plots corresponding to the mapping provided by the GA-based technique [2] have been labeled as "GA-Based mapping". Figures 4 a) and b) show on the X-axis the packet injection rates. These values are the scale factor of the different communications which form the MMS. These communications have been equally scaled up to analyze the network behaviour at different traffic loads. On the Y-axis, Figure 4 a) shows the average message latencies obtained during the simulations, while Figure 4 b) shows the power consumption required by the NoC when using each of the mappings for executing the MMS.

Figure 4 a) shows that the mapping provided by the proposed technique obtains the lowest latencies. Although the improvement achieved is not significant if compared with the GA-based mapping, the main advantage of the proposed technique is that it is capable of obtaining better latencies without testing each mapping explored. Figure 4 b) shows that the mappings provided by the

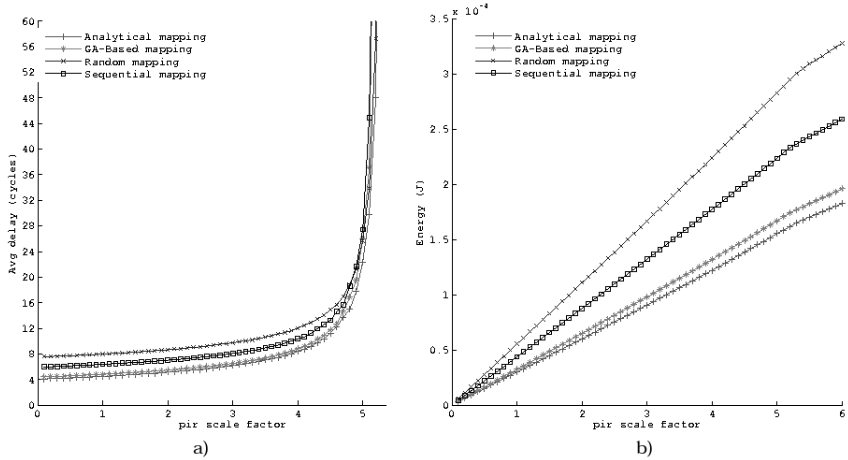

Fig. 4. a) Average latencies b) Energy consumption provided by the mapping techniques

proposed technique require the lowest power consumption. The differences among the considered mapping increase as the traffic injection rates are higher. This behavior is due to the fact that the proposed mapping technique maps closer those tasks that communicate more frequently. As the traffic generated by the tasks increases, the energy saving with respect to other techniques is greater.

These results show that the communication-aware topological mapping can significantly improve the network performance, particularly in terms of energy consumption. Moreover, the proposed technique does not need to test each mapping considered, thus requiring a much lower computational effort than the GA-based method. Additionally, this communication-aware mapping technique can be applied to NoCs with adaptive routing algorithms.

5 Conclusions and Future Work

In this paper, we have proposed a communication-aware topological mapping technique for NoCs. This technique consists of globally matching the communication requirements of the application(s) tasks running on the IPs with the existing network resources. Unlike other mapping techniques, the proposed method does not need to experimentally validate each mapping considered, thus requiring a much lower computational effort.

The evaluation results show that the proposed technique can improve the network performance in regard to other mapping techniques that require experimental validation. Additionally, the proposed technique can be applied to both regular and irregular network topologies.

As a future work to be done, we plan to study the application of this technique to irregular NoC topologies. Also, we plan to analyze the joint application of the proposed technique with recently proposed adaptive routing algorithms.

References

1. Dally, W.J., Towles, B.: Route packets, not wires: On-chip interconnection networks. In: Proceedings of Design Automation Conference, pp. 684–689 (2001)
2. Ascia, G., Catania, V., Palesi, M.: Mapping cores on network–on–chip. International Journal of Computational Intelligence Research 1(1–2), 109–126 (2005)
3. Lei, T., Kumar, S.: A two–step genetic algorithm for mapping task graphs to a netword on chip architecture. In: Proceedings of the Euromicro Symposium on Digital System Design (DSD 2003), IEEE Computer Society Press, Los Alamitos (2003)
4. Hu, J., Marculescu, R.: Energy-and-performance-aware mapping for regular noc architectures. IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems 24(4), 551–562 (2005)
5. Garey, M.R., Johnson, D.S.: Computers and intractability: a guide to the theory of NP-completeness. Freeman and Company, New York (1979)
6. Schafer, M., Hollstein, T., Zimmer, H., Glesner, M.: Deadlock-free routing and component placement for irregular mesh-based networks-on-chip. In: Proceedings of IEEE/ACM International Conference on Computer-Aided Design, 2005 (ICCAD 2005), pp. 238–245 (2005)
7. Bolotin, E., Cidon, I., Ginosar, R., Kolodny, A.: Routing table minimization for irregular mesh nocs. In: Proceedings of Design, Automation & Test in Europe Conference & Exhibition (DATE 2007), pp. 1–6 (2007)
8. Orduña, J., Silla, F., Duato, J.: On the development of a communication-aware task mapping technique. Journal of Systems Architecture 50(4), 207–220 (2004)
9. Chang, J.M., Pedram, M.: Codex-dp: Co-design of communicating systems using dynamic programming. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 19(7), 732–744 (2002)
10. Varatkar, G., Marculescu, R.: Traffic analysis for on-chip networks design of multimedia applications. In: ACM/IEEE Design Automation Conference, June 2002, pp. 510–517 (2002)
11. Arnau, V., Orduña, J., Ruiz, A., Duato, J.: On the characterization of interconnection networks with irregular topology: A new model of communication cost. In: Proceedings of IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 1999), pp. 1–6 (1999)
12. Haupt, R.L., Haupt, S.E.: Practical Genetic Algorithms. Willey (1997)
13. Laarhoven, P.J., Aarts, E.: Simulated Annealing: Theory and Applications. Mathematics and its Applications, vol. 37. Springer, Heidelberg (1987)
14. Festa, P., Resende, M.: Grasp: An annotated bibliography. In: Hansen, P., Ribeiro, C. (eds.) Essays and Surveys on Metaheuristics, pp. 325–367. Kluwer Academic Publishers, Dordrecht (2002)
15. Duato, J., Yalamanchili, S., Ni, L.: Interconnection Networks: An Engineering Approach. IEEE Computer Society Press, Los Alamitos (1997)