A Multi-level Routing Scheme and Router Architecture to Support Hierarchical Routing in Large Network on Chip Platforms

Rickard Holsmark[1], Shashi Kumar[1], and Maurizio Palesi[2]

[1] School of Engineering, Jönköping University, Sweden
{rickard.holsmark,shashi.kumar}@jth.hj.se
[2] DIIT, University of Catania, Italy
mpalesi@diit.unict.it

Abstract. The concept of hierarchical networks is useful for designing a large heterogeneous NoC by reusing predesigned small NoCs as subnets. In this paper we show that multi-level addressing is a cost-effective implementation option for hierarchical deadlock-free routing. We propose a 2-level routing scheme, which is not only efficient, but also enables co-existence of algorithmic and table-based implementation in one router. Synthesis results show that a 2-level hierarchical router design for an 8*x*8 NoC, can reduce area and power requirements by up to ~20%, as compared to a router for the flat network. This work also proposes a new possibility for increasing the number of nodes available for subnet-to-subnet interfaces. Communication performance is evaluated for various subnet interface set-ups and traffic situations.

Keywords: Networks on Chip, Hierarchical Networks, Router Architecture.

1 Introduction

NoC will be the ideal communication infrastructure for next generation SoCs with hundreds of cores as predicted by ITRS. The concept of hierarchy will be helpful in designing and using such NoC platforms with growing number of cores. Whether hierarchical or not, the formation of packet deadlocks may be fatal to any network communication. To avoid this, several deadlock-free routing schemes have been proposed in literature, e.g. Turn model [1], Odd-Even [2] and Up*/Down* [3]. Deadlock freedom may be compromised when combining different networks, each with its own deadlock-free routing algorithm. Therefore, an important new issue in hierarchical NoCs is the design of deadlock-free routing algorithms.

Holsmark et al. [4] proposed hierarchical deadlock-free routing and showed that if subnets are interconnected by "safe boundary" nodes, it is possible to design a deadlock-free global routing algorithm without altering any internal subnet routing algorithm. But Holsmark et al. [4] assumed a flat implementation with a common address space for all network nodes. In this work we propose that a hierarchical routing function is implemented in two levels. The higher level routing function will determine if the destination for a packet is inside or outside the local subnet. If the destination is outside the current subnet, the address of the entry node of the

M.R. Guarracino et al. (Eds.): Euro-Par 2010 Workshops, LNCS 6586, pp. 153–161, 2011.

destination subnet is used for routing the message. Upon reaching the destination subnet, the lower level routing function determines the route using the local address of the destination node. Hence, multi-level routers need only to store addresses to subnet entry-nodes for destinations in other subnets, rather than addresses to all nodes. This reduces router table-sizes and we show that the 2-level router architecture indeed enables significant reduction in area and power consumption.

One important parameter which affects performance is the number of safe boundary nodes of a subnet [4]. Since some routing algorithms provide very few safe nodes, we propose the concept of “safe channels” to attain higher connectivity, and hence higher performance of a network. We have compared the performance of 2-level routing with common deadlock-free routing algorithms and explored the effect of varying the number of boundary nodes.

Recently the topic of hierarchical NoCs has caught the attention of researchers. Several aspects have been studied, for example Bourduas et al. [5] have proposed a hybrid ring/mesh interconnect topology to remove limitations of lengthy diameter of large mesh topology networks. Deadlock-free routing in irregular networks often implies a strongly limited set of routing paths. To increase the available paths, Lysne et al. [6] developed a routing scheme, which avoids deadlock by assigning traffic into different layers of virtual channels.

2 Safe Channels for Increased Connectivity in Hierarchical Routing Algorithms

The methodology for hierarchical routing algorithms [4] used the concept of safe boundary nodes to ensure deadlock freedom. Whether a node is “safe” or not depends on each subnet routing algorithm and is checked by analysis of internal CDG (Channel Dependency Graph [7]) paths. If there are no internal paths from any output to any input of a node, it is *safe* (see Fig. 1). If such a path exists, the node is *unsafe* and may enable formation of CDG cycles with paths in other subnets. The requirement that all boundary nodes should be safe often reduces the number of possible boundary nodes in a network.

For deterministic routing algorithms, like XY, all boundary nodes are safe. Partially adaptive algorithms provide only a few safe boundary nodes, e.g. an *NxN* network with Odd-Even [2], or West-First [1] provides only *N*, whereas Negative-First [1] provides *N*+ (*N*-1) nodes. To remedy this situation we propose the concept of

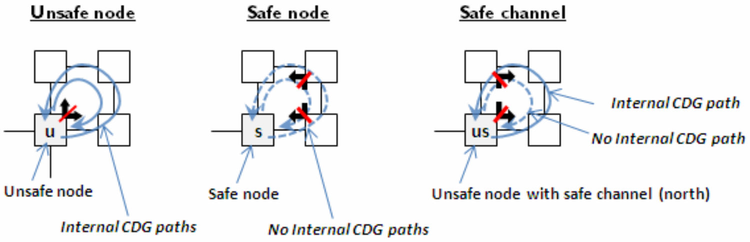


Fig. 1. Examples of unsafe boundary nodes, safe boundary nodes and safe channels

safe *channels*. Given a node *n*, and an internal output channel *c* of node *n*, *c* is a safe channel if there does not exist an internal CDG path from channel *c* to any input channel of *n*.

Fig. 1 illustrates the differences between unsafe nodes, safe nodes and safe channels. The node ***u*** is *unsafe* because the routing restriction allows CDG paths between its internal ports. The restrictions in the *safe* node example prevent such CDG paths and ***s*** is therefore safe. In the safe *channel* example, it is straightforward to see that only one of the internal output channels of node ***us*** (unsafe with safe channel) is on a CDG path to an input channel of ***us*** itself. Using this safe channel and restricting the use of the other channel would, from a deadlock freedom perspective, be equivalent to using a safe node. Note that safe channels cannot relax the requirement of at least one safe boundary node in each subnet. The effect of adding unsafe nodes with safe channels is explored in the evaluation section.

3 Two-Level Routing Scheme

3.1 Addressing and Routing Protocol

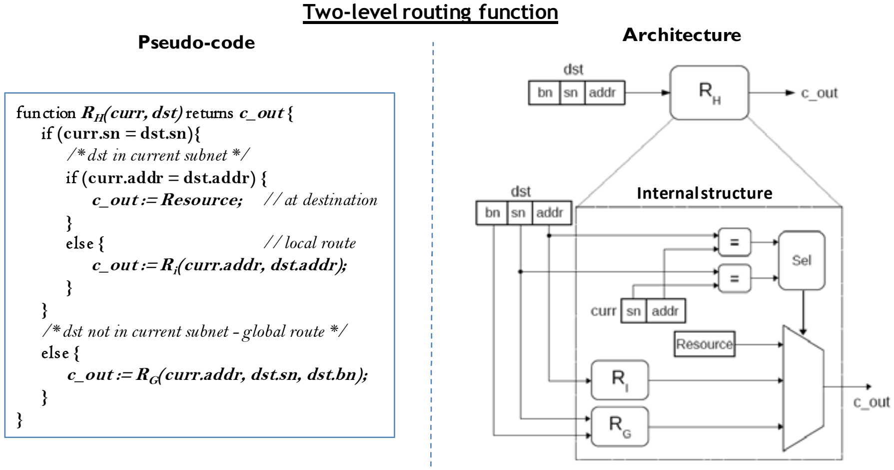
Availability of multiple boundary nodes requires that information of the destination subnet boundary node is added (see example in section 3.3). Therefore a source node tags the header destination address with three fields [*subnet id, boundary node, node id*]. The routing protocol is identical for all nodes. Each node first checks to which subnet a packet is destined. If it is in the current subnet, an internal routing function determines the route. Otherwise, it is forwarded by an external routing function.

If subnets are heterogeneous, the encoding of node address in the source subnet may differ from the encoding in the destination subnet, both with respect to size and topology. In general, the header field for node address must be adjusted according to the subnet requiring largest number of bits for node address. The size of the field for subnet addressing depends on the number of subnets.

3.2 Two-Level Routing Function

The 2-level routing function is partitioned into an external routing function R_G and a subnet internal routing function R_i. The internal routing function is identical to the routing function as if the subnet is a stand-alone network. One feature which is enabled by 2-level routing is the possibility to mix implementation techniques of the internal routing functions in different subnets. This implies that routers in some subnets may be table-based while other routers may implement algorithmic routing.

Fig. 2 gives pseudo-code of the main hierarchical routing function R_H and the proposed router architecture. The routing function takes *dst* which contains the destination subnet *dst.sn*, destination boundary node *dst.bn* and node address *dst.addr*. If both destination subnet and node addresses match with current subnet and node addresses, the function returns the local resource channel. If the destination resides in the same subnet as the current node, the local routing function R_i is called with the destination node address *dst.addr*. The returned output channel *c_out* will in this case always be internal. Should the subnets not match, the external routing function is invoked with destination subnet *dst.sn* and boundary node *dst.bn*. The external routing function can return both external and internal channels if current node is a boundary node. If current node is not a boundary node it will only return internal channels. The 2-level router tables are built similarly to the flat router tables, using breadth-first search for computing paths. The main difference is that only paths to destination subnets and boundary nodes are stored in the external table. This means that during the search, for each source-destination pair, the node where the last transition between different subnets was made is stored as boundary node for the destination. This information is used for addressing by the source node. Simultaneously, the output channel from which the boundary node can be reached is stored in the router table.



```
function $R_H$(curr, dst) returns c_out {
    if (curr.sn = dst.sn){
        /* dst in current subnet */
        if (curr.addr = dst.addr) {
            c_out := Resource;    // at destination
        }
        else {                    // local route
            c_out := $R_I$(curr.addr, dst.addr);
        }
    }
    /* dst not in current subnet - global route */
    else {
        c_out := $R_G$(curr.addr, dst.sn, dst.bn);
    }
}
```

Fig. 2. Pseudo-code and architecture of two-level routing function

Since all paths are obtained using the hierarchical deadlock-free routing methodology [4], it can be shown that the 2-level scheme is deadlock free and connected as well. If the destination is in another subnet, such paths must traverse a boundary node in the source subnet and a boundary node in the destination subnet.

3.3 A Small Example of Routing in Two-Level Router Networks

This small example illustrates routing in 2-level networks as well as the requirement of boundary node id for external destinations. Study Fig. 3 where each subnet S_1, S_2 and S_3 is a 2*x*2 mesh with routing algorithms XY, YX and XY respectively. The external algorithm in this case is assumed to be YX. Boundary nodes are indicated by double border.

Consider routing a message from source node $n_{1,1}$ in subnet S_1 to the destination node $n_{2,2}$ in subnet S_2. The source node is identified with subnet and node address, *src* = (S_1, $n_{1,1}$). Destination address contains subnet, boundary node and node address, *dst* = (S2, b2, n2,2). When the routing function is called in src, the subnet fields do not match and the external function will be used. The external function returns the *East* channel, i.e. $R_G((n_{1,1}), S_2, b_2)$ = *East*. Note that this is the only allowed route according to the internal XY algorithm. At node *curr* = (S_1, $n_{1,2}$), the external algorithm returns *South*. Using *East* would neither violate the internal algorithm restriction. However, this shows the necessity for boundary node specification in the packet header. If the external address is specified using subnet id alone, it would not be possible to distinguish between destinations in row 1 and row 2 in subnet S_2.

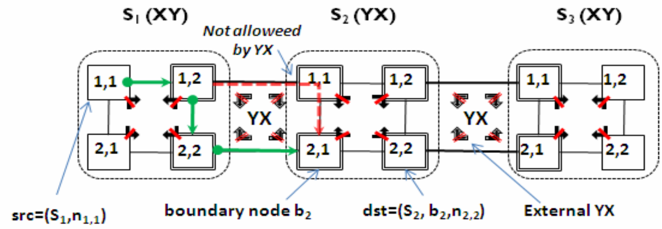


Fig. 3. Example of two-level addressing

In this case, for reaching node *dst* the *only* allowed route is *South*, as the packet cannot make this turn at row 1 in subnet S_2 since both the internal algorithm and external algorithm is YX. After turning south, eventually the packet arrives at node $n_{2,1}$ in subnet S_2. Since the current subnet is now the same as the destination subnet, the node address and local algorithm is used for routing to the destination, i.e. R_i = ($n_{2,1}$, $n_{2,2}$) = *East.*

4 Synthesis Results

We evaluate area and power requirements for the network structures given in Fig. 5(right). Straightforwardly compared, flat addressing in this case requires 64 (8x8) entries per router whereas, e.g., 2-level addressing with one boundary node requires one table of 4 entries for subnet addresses and one table with 16 (4x4) entries for local addresses.

Area and power reduction (routing function) for 2-level routers w.r.t. flat router ⇨

Synthesis Results

Router	Description	Routing Function		Complete Router	
		Area	*Power (uW)*	*Area*	*Power (uW)*
RF	*Flat 8x8 mesh*	3928	2993,4	21781,5	19884
RH-1bn	*2L table 1 bn*	1268,2	1176,1	19121,6	18066,7
RH-4bn	*2L table 4bn*	1974,4	1749,1	19827,8	18639,6
RH-7bn	*2L table 7 bn*	2675,2	2306,2	20528,6	19196,8
RH-1bn-xy	*2L tbl/alg 1 bn*	317,1	332,7	17569	16426,7

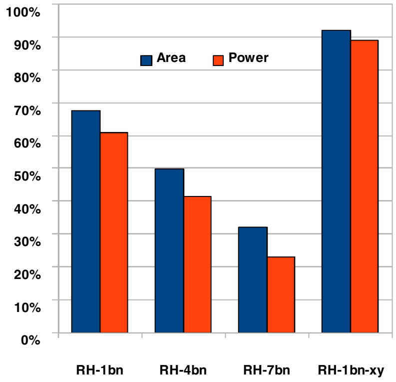


Fig. 4. Area and power for different two-level router versions (RH-*x*bn) and a flat router (RF)

Fig. 4 presents synthesis results from UMC 65nm technology library, assuming 1 GHz clock frequency. The results for implementation of a flat routing function are indicated by the label *RF*. The 2-level variants are indicated according to the number of boundary nodes (*RH-1bn*, *RH-4bn* and *RH-7bn*). The table also provides data for 2-level routing with one boundary node and algorithmic XY routing (*RH-1bn-xy*). Results are given for one routing function per router. The table gives area and power consumption separately for the routing function as well as the whole router. The main share of cost of the complete router is dominated by input buffers of 4 flits for each input port. Fig. 4 also summarizes the percentage of area and power reduction of the 2-level routing functions as compared to the flat routing function.

The largest reduction in area, about 65 percent for the routing function (and ~12 percent for the complete router), is obtained by the configuration with one boundary node (*bn1)*, which only needs to store one entry per subnet. As the number of boundary nodes increase so do the resource requirements of the routing function. Power reduction is slightly less than area reduction for all configurations. Considering the algorithmic implementation with XY as local routing function, it is shown that area and power for the routing function can be reduced by about 90 percent.

5 Simulation Based Performance Evaluation

The evaluations are performed with a simulator designed in SDL using Telelogic SDL and TTCN Suite 6.2. Wormhole switching is employed, with packet size fixed at 10 flits. Routers are modeled with input buffers of size 4 and flit latency of 3 cycles per router. Packet injection rate *pir* is given in average number of packets generated per cycle. Thus at *pir*=0.02, each node generates on average 2 packets per 100 cycles.

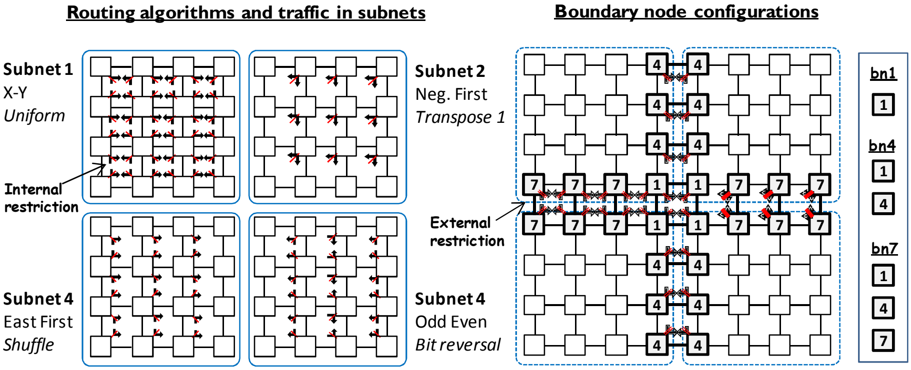


Fig. 5. A network with four subnets and boundary node configurations

Different levels of external subnet traffic w.r.t. local subnet traffic are used. This means that out of the total, 75% is local traffic and 25% of the traffic is sent outside the source subnet. External traffic destinations are uniformly distributed over the whole network. The used subnet configurations are given in Fig. 5(left). Each subnet exhibits a specific traffic type, which in the case of hierarchical routing is matched with a suitable routing algorithm, (e.g. Subnet 2: Negative First, Transpose1). Fig. 5(right) shows the three configurations of boundary nodes and external routing restrictions in the evaluations. Nodes labeled 1, represent the one boundary node per subnet case (bn1).The four boundary node set-up (bn4) additionally uses the nodes labeled 4, while the seven boundary node instance (bn7) includes all numbered nodes.

The bn1 and bn4 set-ups utilize only safe nodes, where bn4 represents the maximum attainable connectivity with safe nodes. The bn7 case allows safe channels of unsafe nodes in subnets 3 and 4 and achieves the maximum connectivity of the topology. Flat algorithms are global, e.g. in the case of XY this means that XY is used for routing all messages. Note that XY is only applicable to the bn7 configuration. The Up*/Down* algorithm is applicable to all different configurations and is annotated similarly to the hierarchical cases, i.e. ud_bn*x*.

5.1 Simulation Results

Fig. 6(left) compares average latency (duration from when a packet was generated at the source to when its tail flit was received at the destination) of the hierarchical *hr_bn7* configuration with XY and Up*/Down* for 100, 95 and 75 percent of message subnet locality. Performance is adversely affected for all algorithms when reducing the internal traffic. The highest performance is obtained by *hr_bn7* with 100% local traffic (hr_bn7_100). Notable is that *hr_bn7* also for 95 % local traffic performs considerably better than both XY and Up*/Down* for 100% local traffic.

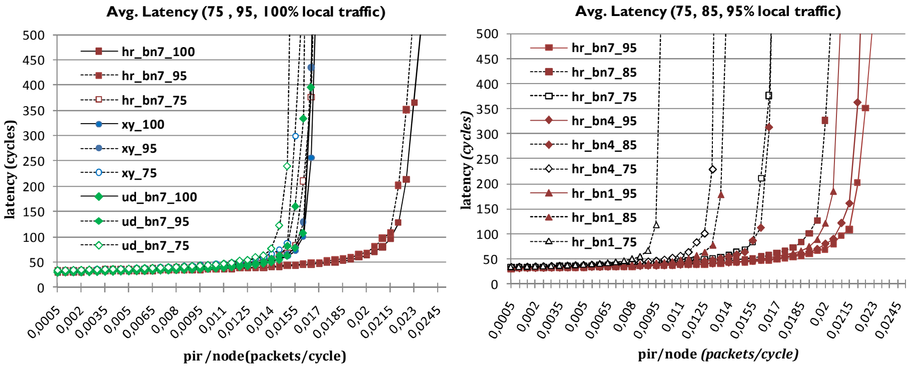


Fig. 6. Average latency: hr vs. other algorithms (left), different hr configurations (right)

For 75% of local traffic the differences are reduced, especially compared to XY. This is quite expected, since XY is known to be a very good algorithm for uniformly distributed traffic (which is the distribution of the external traffic). Studying the results for four- and one- boundary node hierarchical (hr_bn4_95 and hr_bn1_95 respectively) in Fig. 6(right), we see that both outperform XY for 95% local traffic (Fig. 6(left)), even though their average distances increase due to longer routes. As the local traffic is reduced, fewer boundary nodes result in significantly higher average latency than XY. The very few external links in *hr_bn1* are effective bottlenecks and the congestion rapidly propagates into the internal subnet traffic.

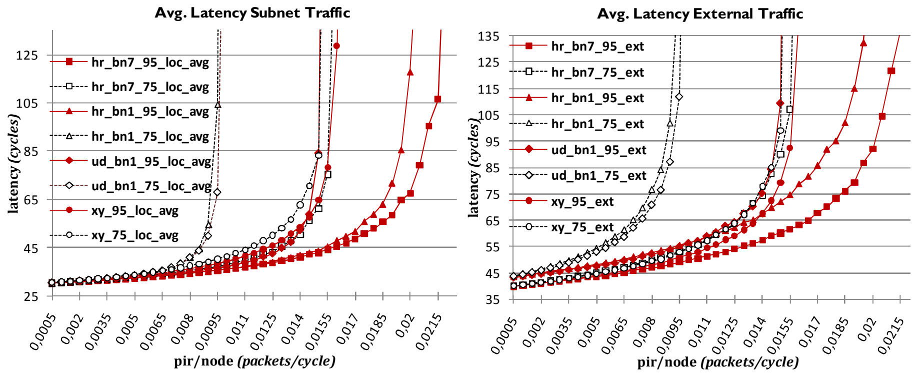


Fig. 7. Average latency for internal subnet traffic (left), external traffic (right)

Fig. 7.(left) compares average latency for different algorithms and internal subnet traffic. Both *hr_bn7* and *hr_bn1* show considerably lower latency values for high load in the 95% local traffic scenario. Note that XY is on a higher curve than Up*/Down* (ud_bn1_95) at low *pir* but improves as *pir* is increased. This indicates that Up*/Down* may have advantage of adaptive routes at lower *pir* compared to XY routing algorithm. Fig. 7(right) complement the subnet latency by showing the latency of the external traffic. The higher base latency for *ud_bn1*and *hr_bn1*, due to fewer external links, is visible at both 75% and 95% of local traffic. But, in spite of lower initial latency, *xy_95* rapidly increases above the latency of *hr_bn1_95* at *pir* of 0.015.

6 Conclusions

In this paper we have proposed both a new routing scheme as well as a structured router design to support deadlock-free routing in a 2-level hierarchical NoC. One important hierarchical network parameter is the number of safe boundary nodes. We have synthesized a router for various values of this parameter and results show that 2-level routing is less costly for area and power consumption as compared to a flat solution. The importance of this advantage will increase with network size and number of boundary nodes. We also observe through simulation that 2-level hierarchical routing with maximum number of boundary nodes, in general, provides higher performance compared to flat routing algorithms. Although it seems that the proposed 2-level scheme will recursively extend itself to *n*-levels, implementation of such schemes will open new challenges.

References

1. Glass, C., Ni, L.: The Turn Model for Adaptive Routing. In: Proceedings of the 19th Annual International Symposium on Computer Architecture, 1992, pp. 278–287 (1992)
2. Chiu, G.-M.: The odd-even turn model for adaptive routing. IEEE Transactions on Parallel and Distributed Systems 11, 729–738 (2000)

3. Schroeder, M., Birrell, A., Burrows, M., Murray, H., Needham, R., Rodeheffer, T., Satterthwaite, E., Thacker, C.: Autonet: a high-speed, self-configuring local area network using point-to-point links. IEEE Journal on Selected Areas in Communications 9, 1318–1335 (1991)
4. Holsmark, R., Kumar, S., Palesi, M., Mejia, A.: HiRA: A methodology for deadlock free routing in hierarchical networks on chip. In: 3rd ACM/IEEE International Symposium on Networks-on-Chip, NoCS 2009, pp. 2–11. IEEE Computer Society, Los Alamitos (2009)
5. Bourduas, S., Zilic, Z.: A Hybrid Ring/Mesh Interconnect for Network-on-Chip Using Hierarchical Rings for Global Routing. In: Proc. of the ACM/IEEE Int. Symp. on Networks-on-Chip (NOCS) (2007)
6. Lysne, O., Skeie, T., Reinemo, S., Theiss, I.: Layered Routing in Irregular Networks. IEEE Trans. Parallel Distrib. Syst. 17, 51–65 (2006)
7. Duato, J.: A new theory of deadlock-free adaptive routing in wormhole networks. IEEE Transactions on Parallel and Distributed Systems 4, 1320–1331 (1993)