

Comparison of Network-on-Chip Topologies for Multicore Systems Considering Multicast and Local Traffic*

Dietmar Tutsch
Bergische Universität Wuppertal
42119 Wuppertal, Germany
tutsch@uni-wuppertal.de

Mirosław Malek
Humboldt-Universität zu Berlin
12489 Berlin, Germany
malek@informatik.hu-berlin.de

ABSTRACT

Performance of two network-on-chip (NoC) topologies is compared for the use in multicore processors. The performance evaluation is supported by the *CINSim* simulator. This simulator has been developed to model a variety of network topologies that are based on atomic components such as buffers, routers, traffic generators, and target buffers. The development of this simulator was driven by the investigation of networks-on-chip. But off-chip networks can be examined as well. Two examples for NoC topologies, a mesh and a bidirectional interconnection network, are compared. Unicast traffic is used as well as multicast and local traffic, which both represent a significant part of the network traffic for evaluating multicore processors. In addition to the performance, the mean distance, the diameter, and the buffer cost are calculated for both network topologies. The results show that bidirectional multistage interconnection networks outperform meshes. A clearly better scalability is shown by the bidirectional multistage interconnection networks.

Categories and Subject Descriptors

C.1.2 [Processor Architectures]: Multiple Data Stream Architectures (Multiprocessors—*parallel processors*); C.2.1 [Computer Communication Networks]: Network Architecture and Design—*network topology*; C.4 [Performance of Systems]: Modeling Techniques

Keywords

network-on-chip, multicore processor, multicast, simulation, performance

1. INTRODUCTION

The ongoing improvement in VLSI technology leads to a further increase in the number of devices per chip. Since this increased density cannot longer be used to improve the performance of uniprocessor chips at a pace as in the past, multicore processors come to the center of interest [5].

*This research was sponsored in part by Intel Corporation.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIMUTools 2009, Rome, Italy

Copyright 2009 ICST ISBN 9-78963-97-9/94/55.

To enable cooperating cores on such a multicore processor, an appropriate communication structure among them must be provided. In case of a low number of cores (e.g. a dual core processor), a shared bus may be sufficient. But in the future, hundreds or even thousands of cores will collaborate on a single chip. Then, more advanced network topologies will be needed. Numerous topologies have been proposed for these so called networks-on-chips (NoCs) [1, 2, 3, 4, 6, 8, 15] and most of them are carried over from parallel computing [9]. For instance, this paper will compare meshes and multistage interconnection networks (MINs) as examples. But most other topologies can also be investigated by using the simulator introduced by this paper.

To map the communication demands of the cores onto predefined topologies like meshes, MINs, and other topologies, Bertozzi et al. [3] developed a tool called NetChip (consisting of SUNMAP [11] and xpipes [14]). This tool provides complete synthesis flows for NoC architectures.

Another example where MINs deal as NoC is given by Guerrier and Greiner [8] who established a fat tree structure using Field Programmable Gate Arrays (FPGAs). They called this on-chip network with particular router design and communication protocol Scalable, Programmable, Integrated Network (SPIN). Its performance for different network buffer sizes was compared.

Alderighi et al. [1] used MINs with the Clos structure. Multiple parallel Clos networks connect the inputs and outputs to achieve fault tolerance abilities. Again, FPGAs serve as basis for realization.

But previous papers only considered unicast traffic in the NoC. It is obvious that multicore processors also have to deal with multicast traffic. For instance, if a core changes a shared variable that is also stored in the cache of other cores, multicasting the new value to the other cores keeps them up to date. Thus, multicast traffic builds a non-negligible part of the traffic.

Furthermore, it is very likely that traffic in multicore processors will reveal some locality in its spatial distribution. Usually, an application will be distributed to some of the cores. But due to many available cores, more than a single application can be processed in parallel. Then, there will be much more communication between cores that process the same application than between cores of different applications. Thus, cores for the same application are chosen such that they are close together to achieve low communication latency. In consequence, local traffic dominates.

As a result, networks for multicore systems should support multicast traffic and local traffic as well. Investigating whether networks are suitable for multicore processors is usually performed by modeling them stochastically. Here, analytical methods as well as simulation are used.

This paper presents a simulator for modeling network-on-chip

topologies. The topology performance can be determined under various traffic patterns including traffic localities and multicast traffic. Thus, the performance of different network topologies can be compared. As an example, the paper evaluates mesh networks and bidirectional multistage interconnection networks. Besides performance in terms of delay and throughput, further parameters like the mean distance between network nodes, the diameter, and the cost in terms of number of buffers are compared.

The paper is organized as follows. Section 2 introduces the architectures of networks-on-chips, particularly multistage interconnection networks and meshes. The NoC simulator is presented in Section 3. Section 4 demonstrates the features of the simulator by comparing mesh networks and bidirectional multistage interconnection networks. Their performance is related to their topology parameters. In Section 5, summary and conclusions are given.

2. NETWORK-ON-CHIP

This section gives two examples for network-on-chip architectures. First, bidirectional multistage interconnection networks are discussed and then, mesh networks as a second approach are described.

2.1 Bidirectional Multistage Interconnection Networks

Multistage Interconnection Networks (MIN) are dynamic networks which are based on switching elements (SE). SEs are arranged in stages and connected by interstage links. The link structure and amount of SEs characterizes the MIN.

MINs [16] of size $N \times N$ (N inputs and N outputs) consist of $c \times c$ switching elements. The number of stages is given by $n = \log_c N$ (with $n, c, N \in \mathbb{N}$) in case of MINs with the banyan property which provide N disjoint paths and for each input-output pair, there exists only a unique path.

Bidirectional MINs (BMIN) [12] consist of at least $n = \log_c N$ stages to allow connections between each input and each output. Their interstage links and their SEs are bidirectional. That means packets can be transferred in both directions. In consequence, each input also represents the corresponding output. Furthermore, turnaround connections are allowed in the SEs resulting in bridged BMINs (in the sequel simply denoted as BMINs). Figure 1(a) depicts the structure of a bidirectional MIN with attached cores. The three transfer directions in bidirectional SEs are shown in Figure 1(b).

If packet switching is applied buffers can be introduced. A packet is first routed from the network input to the right, denoted as forward direction. As soon as it reaches a stage from which a path exists in backward direction (that means from right to left) to its destination output, it turns around. This stage is called turnaround stage. Finally, the packet proceeds its way in backward direction to the desired output. This routing algorithm belongs to the shortest-path routing techniques.

During its movement in forward direction, the packet may choose any arbitrary SE output because each SE output offers a path to the network destination output via a turnaround stage. Moreover, all paths that a particular packet may choose reveal the same stage as turnaround stage due to the MIN structure. That means all redundant paths are of equal length.

In backward direction, only a single path through the network exists to reach a particular output.

2.2 Mesh Networks

A static network architecture for NoCs is a mesh [7]. In such an architecture, the cores are located at the crosspoints of the mesh.

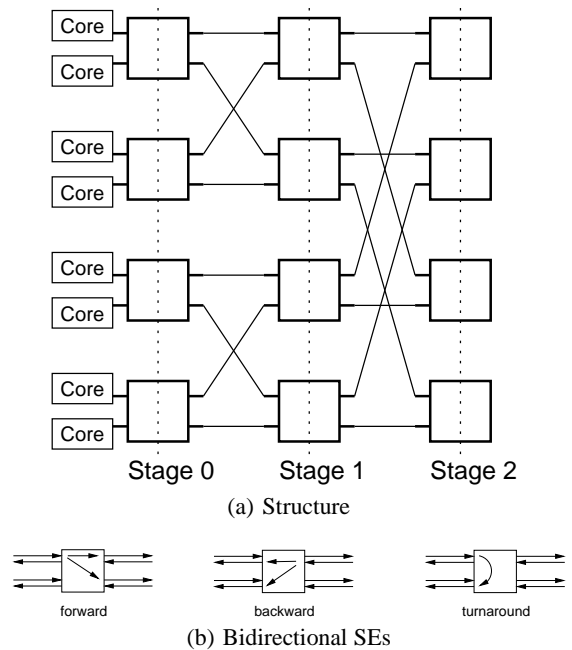


Figure 1: Bidirectional MIN

Three kinds of meshes are distinguished: one-dimensional meshes (also called chains), two-dimensional meshes (2-D meshes, grids), and three-dimensional meshes (3-D meshes). Figure 2 shows a 2-D mesh. The nodes of the mesh incorporate a core and a 5×5 SE (Figure 2(b)), optional with buffers. The SE connects all inputs and outputs of the node to allow packets to pass the node. Furthermore, the core is linked via the SE to the rest of the mesh.

Each node is connected to its two nearest neighbors in each dimension. For instance, four bidirectional links handle all communication of a node in a 2-D mesh (Figure 2(a)). The number of links per node does not change if additional cores (i.e. nodes) are added to the mesh. Therefore, a mesh offers very good scalability. Its blocking behavior reveals one of the most important disadvantages of meshes. Usually, messages pass several nodes and links until they reach their destination. As a result, the same link may be demanded by many connections: blocking may occur. Thus, messages are mostly transferred by packet switching to deal with the blocking by introducing buffers.

Meshes as well as BMINs reveal some locality. The next section discusses this locality and shows how to profit from it.

2.3 Locality

Two aspects of locality have to be considered. First, the locality of network traffic due to applications that are distributed to different set of cores. Traffic within a set of cores can be assumed to be more intensive than traffic between different sets representing different applications.

Second, the network topology reveals some locality in its structure. Figure 3 points out the locality of bidirectional MINs [10]. The structural locality for Core 0 (connected to Input/Output 0) is demonstrated. There is a very high locality for Core 0 with Core 1 (dark grey area). The communication path is very short (just a turnaround at Stage 0).

Less locality can be found between Core 0 and Core 2 or Core 3 (medium grey area). Here, packets must pass three stages to reach

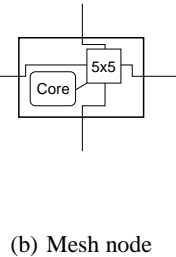
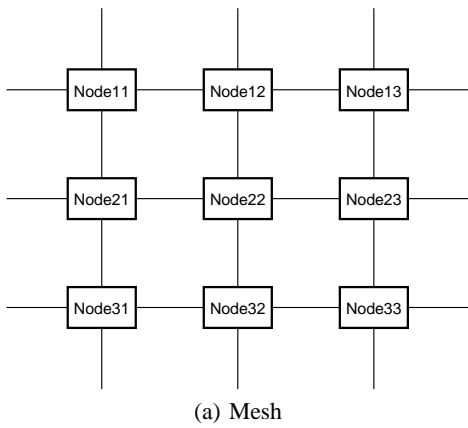


Figure 2: 2-D mesh architecture

the destination: Stage 0, a turnaround in Stage 1, and finally backwards via Stage 0. No locality can be seen for Core 0 when communicating with one of the cores numbered from 4 to 7 (light grey area) is initiated. All network stages are involved.

In meshes, it is obvious that the communication path to neighbor cores is much shorter than for instance the path between two cores in opposite corners.

In consequence, both aspects of locality should be mapped when applications are distributed to different cores: The cores should be chosen such that they reveal structural locality resulting in fast communication. However, sometimes it may not be possible to chose the cores in this way because either cores of structural locality are already occupied by other applications or the application is distributed to more cores than locally connected ones.

3. CINSIM SIMULATOR

The new *CINSim* simulator (Component-based Interconnection Network Simulator) supports modeling and performance evaluation of component-based interconnection networks. It is designed to provide a single simulator for different kinds of network architectures that are based on atomic components such as switches and buffers. Regular network topologies can be modeled as well as irregular ones. The development of this simulator was driven by the investigation of networks-on-chip. But off-chip networks can be examined as well.

The *CINSim* tool consist of two parts: a simulator core performing the simulation runs and a simulator graphical user interface (GUI) to design and draw the networks under investigation (see Figure 4).

The simulator core contains the implementation of network components and their behavior. Any network can be modeled if based on switches (routers), buffers, sources (traffic generators), destina-

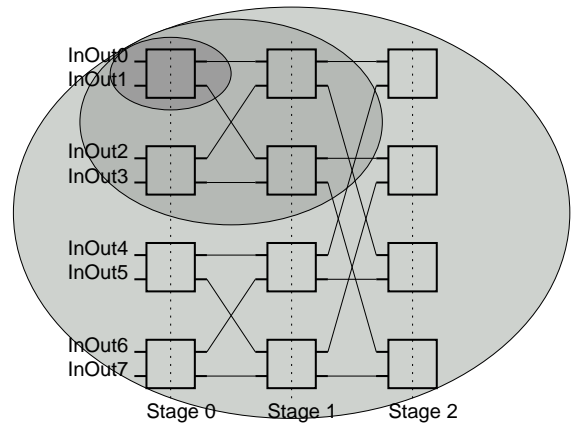


Figure 3: Locality in bidirectional MINs

tions (target buffers), and routes (links) connecting them.

Switches are components to realize dynamically changing connections between switch inputs and outputs. Inputs and outputs are connected according to the requested network output of the message. Thus, they perform some routing and may also be called routers. If multiple inputs contain messages destined to the same output, a scheduling algorithm chooses one of the messages. Currently, random choice, round-robin, least recently used, most recently used, least frequently used, and most frequently used are implemented.

Buffers store packets if packet switching is applied. Shared buffers connected to multiple switch inputs/outputs are implemented as well as non-shared (single-queued) buffers.

Sources produce traffic which is offered to the network. Various destination traffic patterns and time-dependent traffic patterns can be generated, both combined with an arbitrary offered load. The traffic generators are driven by a random number generator.

Destinations represent the outputs of the network. They are in charge to remove the messages from the outputs as soon as they arrive.

Additionally to these components, *CINSim* also offers analyzers for performance measurement. Analyzers can be connected via observer lines to buffers, sources, or destinations to determine the source or destination throughput, the delay, or the buffer queue sizes.

Various traffic interarrival times, like heavy tailed distributions and geometric distributions, can be chosen. Besides the distribution in time, *CINSim* also supports traffic distributions in space. For instance, traffic locality and multicast traffic can be simulated, which are mainly investigated in the sequel of this paper.

Due to the complex stochastic events, confidence levels and estimated precisions must be observed during simulation to achieve a given accuracy. *CINSim* provides exhaustive functionality for accuracy prediction. The simulation is observed by permanently collecting the measured performance results and by calculating the confidence level and precision. If the termination criteria are met, *CINSim* stops the simulation. Besides mean values, quantiles can also be determined for characterizing the distribution of the measure in question.

Steady-state simulation is supported as well as terminating simulation. Terminating simulation is used to investigate the transient behavior of the networks in question.

The simulator also offers a random number generator with very

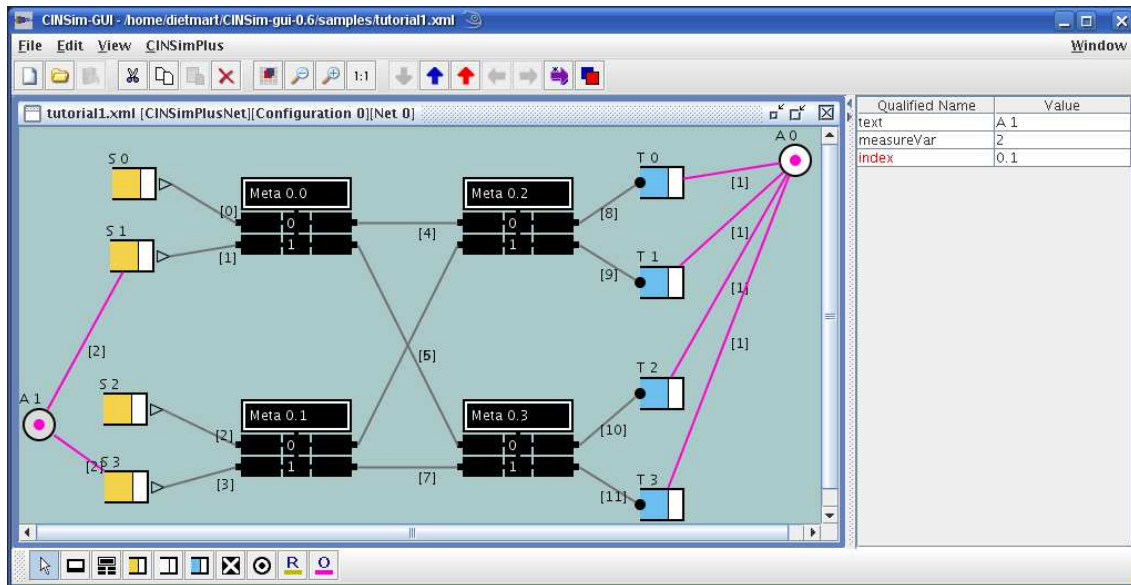


Figure 4: GUI of *CINSim*

long cycles. It also supports distributed simulation to accelerate the simulation runs by starting multiple replications of the simulation in parallel on connected computers or a multicore processor.

The graphical user interface (GUI) as shown in Figure 4 provides a comfortable editor to draw the network that is to be investigated. The predefined components like buffers, switches, etc. can be added to the drawing area to construct the network. Copying parts of the current drawing is supported as well as creating meta components with underlying subnetworks. A meta component can again consist of meta components. Thus, a hierarchical drawing and modeling can be realized.

Furthermore, the *CINSim* simulator allows to model the dynamic reconfiguration of networks. The dynamic reconfiguration of network architectures seems to be a promising way for network performance enhancement [10]. Dynamic network reconfiguration is not a topic of this paper.

4. MESH VERSUS BMIN

Comparative analysis are carried out using *CINSim* to evaluate mesh networks and bidirectional multistage interconnection networks. Besides performance in terms of delay and throughput, further parameters like the mean distance between network nodes, the diameter, and the cost in terms of number of buffers are compared. Also, the problems of scalability are discussed.

4.1 NoC Hardware Cost

The investigated NoC architectures use packet switching. Thus, the switching elements in the BMIN and the mesh nodes provide buffers: a buffer is located at each SE input. Buffers are the main factor for the hardware cost of NoCs: For fully static standard cell-based CMOS $0.18\mu\text{m}$ technology, the consumed silicon area of a FIFO buffer is, for instance, around 10.000 equivalent two-input NAND gates for a single flit FIFO with a flit size of 35 bit [13]. In some switching techniques like virtual cut-through switching, packets are divided into flits (flow control units). In Pande et al. [13], a flit size of 90 bit leads to around 24.000 equivalent two-input NAND gates.

Compared to this, realizing a switching element needs only around 1200 equivalent two-input NAND gates per input, an order of magnitude less consumed silicon area than a buffer occupies.

Therefore, the number of buffers will represent the network cost in the sequel. In off-chip networks, the number of pins also takes an important part of the network cost. But on-chip networks need no pins to connect the network and the attached processor cores.

In the following, a mesh and a BMIN consisting of a similar number of buffers and, thus, of similar cost are compared. Considering the given buffer distribution, networks connecting, e.g., $N = 16$ nodes (processor cores) results in comparable cost. A 16×16 mesh results in 64 buffers and a 16×16 BMIN with 4×4 SEs in slightly less, in 48 buffers.

In general, the number of buffers B_m of meshes adds up to five buffers for each node (one for the four external inputs and one for the input from the core). The unused inputs at the four edges of the mesh can be subtracted. Assuming a mesh of quadratic geometry (with side length $w = \sqrt{N}$), the number of buffers is yield by

$$B_m(N) = 5N - 4\sqrt{N}. \quad (1)$$

The number of buffers B_b of a BMIN is given by the number of stages n where each of the N bidirectional input-output rows of a stage consists of two buffers, one for each direction. But the last stage has only a single input direction and, thus, only a single buffer is located in each row:

$$B_b(N) = (n - 1) \cdot 2N + N = N \cdot (2 \log_c N - 1) \quad (2)$$

Figure 5 shows the number of buffers dependent on the network size N . Smaller network sizes are scaled and depicted in Figure 6. The SE size of the BMIN is set to $c = 4$. For smaller networks, the number of buffers differs only slightly between mesh and BMIN. For larger networks, BMINs suffer from higher buffer cost. But the differences between both curves are moderate. Furthermore, one should be aware that the number of SE inputs was counted to obtain the number of buffers. This gives also the number of links between the nodes and SEs in the network and, therefore, represents the bandwidth of the network: The bandwidth of larger BMINs outperforms the bandwidth of meshes. The performance

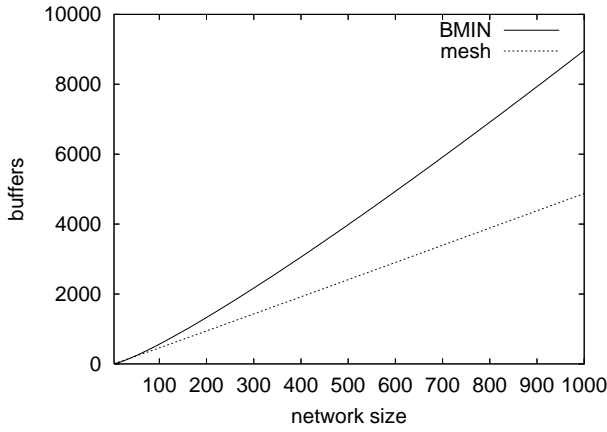


Figure 5: Comparison of number of buffers

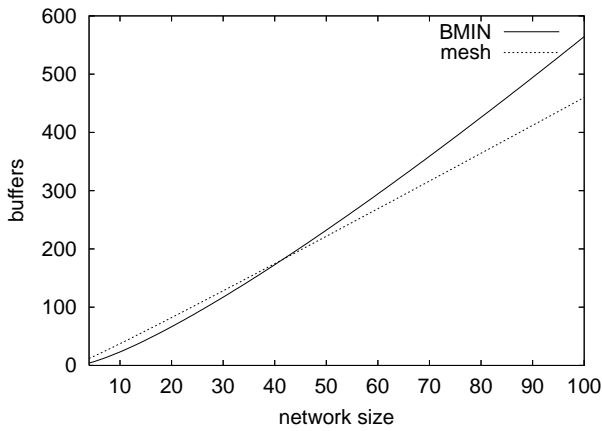


Figure 6: Number of buffers in small NoCs

results determined by the simulator later in this section will confirm this.

4.2 Mean Distance and Diameter

An important measure to estimate the latency of messages in the NoC are the mean distance and the diameter. The mean distance \bar{r} represents the average path length between two nodes of the network in hops. The diameter \emptyset gives the path length in hops for the two nodes with the highest distance.

The mean distance of a mesh with a quadratic geometry (side length $w = \sqrt{N}$) is yield by averaging distances between all node pairs $((x_1, y_1), (x_2, y_2))$ of the mesh with $1 \leq x_1, x_2, y_1, y_2 \leq w$:

$$\bar{r}_m(N) = \frac{\sum_{x_1=1}^w \sum_{y_1=1}^w \sum_{x_2=1}^w \sum_{y_2=1}^w |x_1 - x_2| + |y_1 - y_2|}{(w^2 - 1)w^2} = \frac{2}{3}\sqrt{N} \quad (3)$$

The diameter of such a mesh is given by

$$\emptyset_m(N) = 2(w - 1) = 2\sqrt{N} - 2 \quad (4)$$

In case of a BMIN with $c \times c$ SEs, the mean distance can be obtained by considering that in a subnetwork of c^i nodes, each node can be reached by passing i stages forward, turning at this stage, and

passing backward: $2i - 1$ hops are needed. But c^{i-1} of these nodes are again located in a subnetwork of this $c^i \times c^i$ one which means that they can reach each other within this subnetwork and need less than $2i - 1$ hops. Considering all subnetwork sizes of c^i leads to the mean distance of an $N \times N$ BMIN with $n = \log_c N$ stages:

$$\begin{aligned} \bar{r}_b(N) &= \frac{\sum_{i=1}^n (c^i - c^{i-1}) (2i - 1)}{N - 1} \\ &= \frac{2N \log_c N}{N - 1} - \frac{c + 1}{c - 1} \end{aligned} \quad (5)$$

The diameter is simply the length of the way to the last stage and back again:

$$\emptyset_b(N) = 2n - 1 = 2 \log_c N - 1 \quad (6)$$

The Figures 7 and 8 depict the mean distance for up to $N = 1000$ nodes and for smaller networks, respectively. Here, the bidirec-

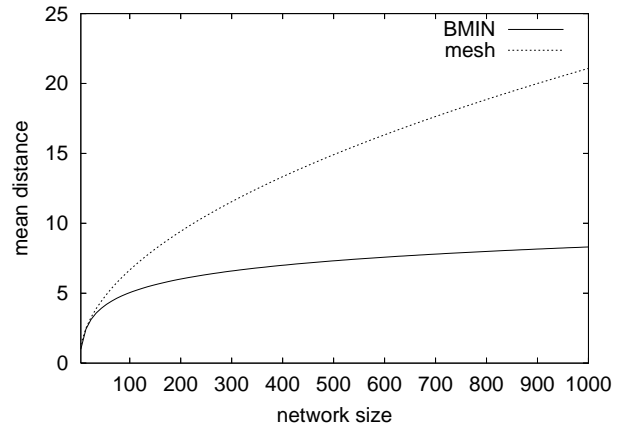


Figure 7: Comparison of the mean distance

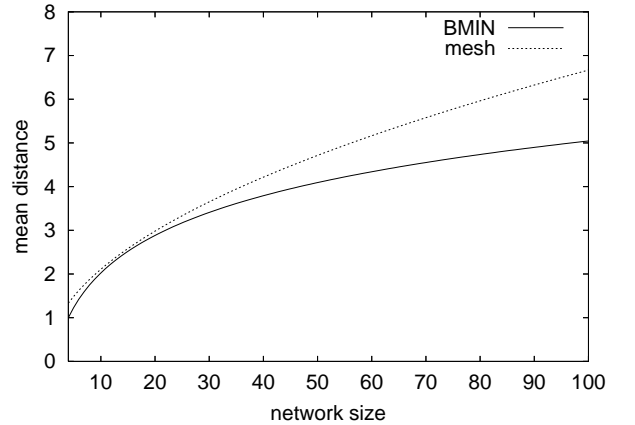


Figure 8: Mean distance in small NoCs

tional MINs are built with SEs of size $c = 4$. The figures show that BMINs outperform meshes in terms of mean distance. The mean distances in BMINs is always smaller than in meshes. This is particularly true for larger NoCs because the mean distance in BMINs grows only logarithmically with the network size while in meshes it grows polynomially.

When developing an NoC, the hardware cost as well as the mean distance are to be minimized. Thus, Figure 9 depicts the product of these parameters for both network topologies. The BMIN topol-

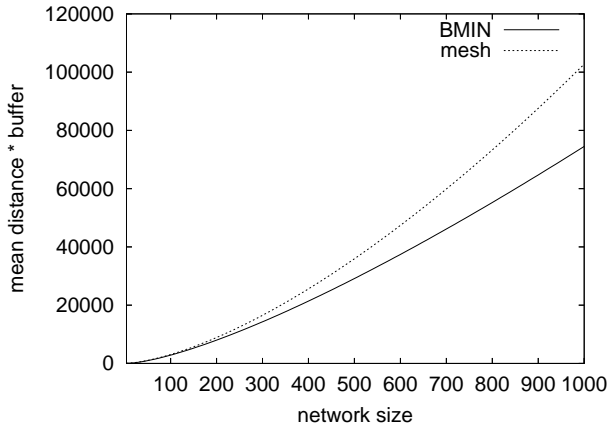


Figure 9: Product of mean distance and number of buffers

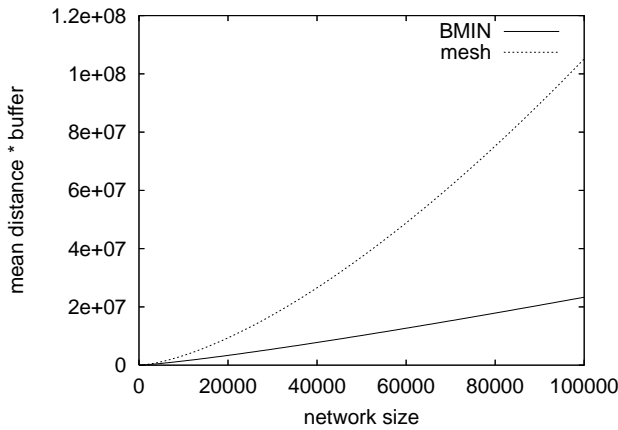


Figure 10: Mean distance and buffer product for large NoCs

ogy clearly shows lower cost delay product. The BMIN becomes increasingly superior to the mesh, the larger the network size (Figure 10).

4.3 Performance

The performance of the mesh and the BMIN topology was determined using the *CINSim* simulator. The topologies were compared for connecting $N = 16$ cores of a multicore processor. For this size, both topologies have comparable hardware cost: the mesh consists of 64 buffers and the BMIN with 4×4 SEs of 48 buffers, respectively.

Both networks operate in virtual cut-through switching with each packet consisting of five flits. The buffers can accommodate two packets. This means that each buffer is of a size to accept 10 flits. Virtual cut-through switching is combined with the local backpressure (clear to send) mechanism to avoid packet loss in case of occupied buffers.

The following performance results are obtained by a scheduling algorithm that solves packet conflicts at SE inputs for the same out-

put randomly.

In our study, a network traffic generator produces by randomization packets with a geometric distribution in time. The network performance is determined dependent on the average offered load to the NoC inputs. The packet destinations are uniformly distributed over the NoC outputs, first. Then, traffic localities as significant multicore traffic patterns are investigated. Such communication between the closest neighbors is examined by starting with only a single communication partner. Then, more and more communication partners are added.

As routing algorithm, the BMIN performs shortest-path routing. That means packets turn as soon as possible from the forward direction to the backward one. The mesh network operates in xy routing.

The *CINSim* simulator obtained the following results by simulating the networks until a confidence level of 98% and an estimated precision of 1% was achieved. To reach this confidence and precision, a simulation run time of less than a minute in case of rare events (e.g. low network load) and of only a few seconds in most other cases has been needed. Simulation has been run on a 2.0 GHz PC. Compared to simulation run time, model set-up time is more time intensive because the automatic generation of (larger) NoC models is still under development. Setting up descriptions by hand needs several minutes or even more dependent on the NoC size. Thus, only smaller NoCs have been evaluated in the following. Automatic model generation will be available soon.

4.3.1 Uniformly Distributed Traffic

Figures 11 to 13 show the performance for unicast traffic in the NoC. The throughput (Figure 11) is given in received packets per

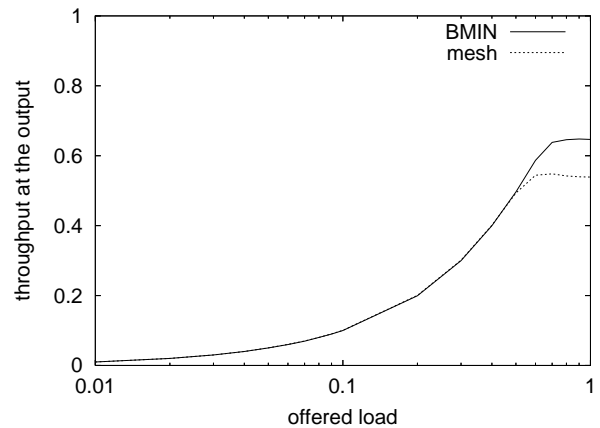


Figure 11: Unicast traffic: throughput

NoC output and per five network clock cycles (needed to receive a single packet consisting of five flits). As a consequence, a maximum throughput of 1 can be theoretically reached. The offered load is similarly defined for the NoC inputs. As the figure shows, there is no significant difference in throughput between mesh and BMIN except for a very high load where the network becomes saturated. Usually, networks are to be dimensioned such that no saturation occurs. Note that the offered load is logarithmically scaled in the figures.

Figure 12 depicts the average delay of the packets in network clock cycles. Here, differences between mesh and BMIN are clearly visible. The BMIN outperforms the mesh for any network load. In case of no saturation, the delay of the mesh is about 30% higher

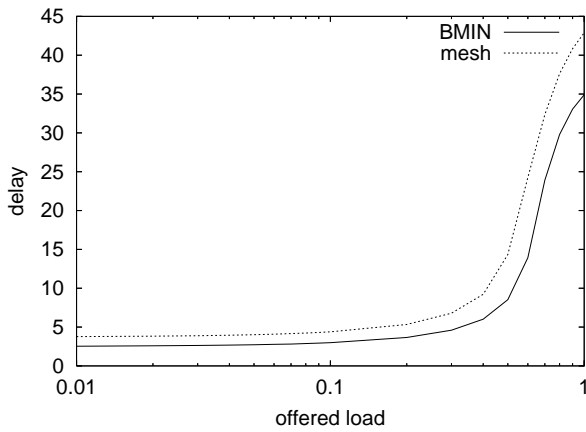


Figure 12: Unicast traffic: delay

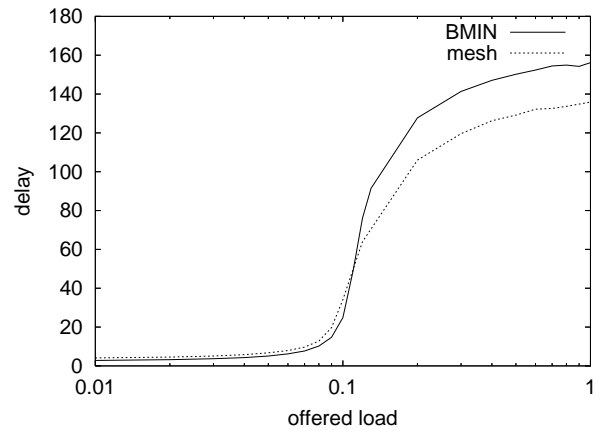


Figure 14: Multicast traffic: delay

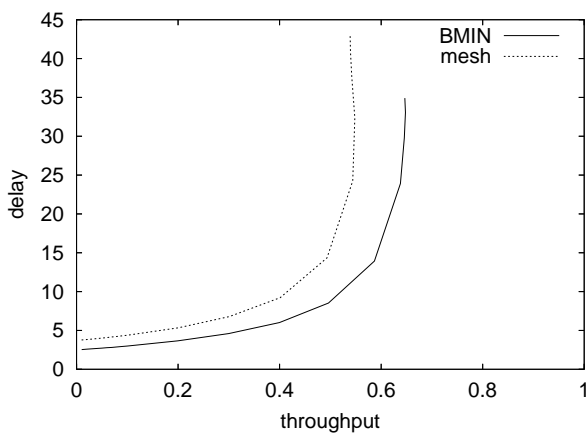


Figure 13: Unicast traffic: delay versus throughput

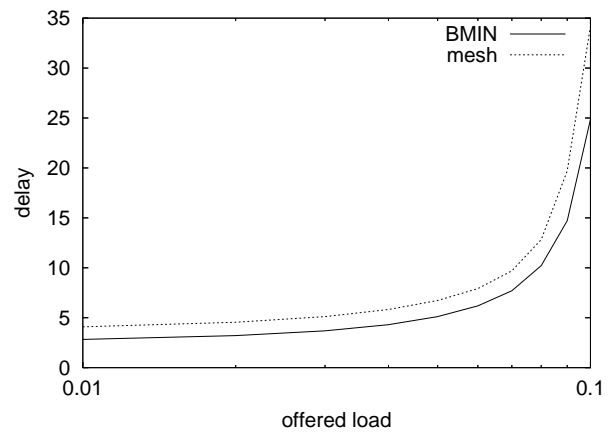


Figure 15: Multicast traffic: delay for light loads

than the delay of the BMIN. In Figure 13, the delay times are compared dependent on the throughput.

Besides unicast, multicast traffic patterns were also investigated due to their importance in multicore processors. The following figures were obtained by choosing a multicast traffic pattern with uniformly distributed destination sets. This means that any possible combination of NoC outputs was chosen with equal probability as a multicast destination of a newly generated packet at the sources.

The shape of the throughput in case of multicasting is similar to Figure 11 except that the saturation of the NoC is starting at a lower offered load of approximately 0.1. The related figure is omitted here. Figure 14 depicts the delay of both network topologies while Figure 15 scales the area where no saturation occurs. In this case, the BMIN again outperforms the mesh with its lower delay: again, the mesh copes with an about 30% higher delay. In case of saturation, the lower delay is shown by the mesh. Up to now, no explanation was found for this behavior. Changing the routing algorithm from xy to west-first routing only slightly changes the shape of the delay curve. Thus, the routing algorithm seems not to be the reason for the given observation. Further investigation is needed.

In Figure 16, the delay times dependent on the throughput are compared to show their interdependence. The figure confirms the

higher performance of BMINs.

An extreme case of multicasts are broadcasts. A traffic pattern where all sources generate only broadcast packets was also investigated. The results do not differ qualitatively from the presented multicast case.

4.3.2 Traffic Localities

Figures 17 to 19 depict the performance of both network topologies if local traffic is involved. Local traffic means that each node only communicates with its closest neighbors. The figures start with the case of communicating to only a single neighbor. Further communication partners are added till a number of five partners is reached.

The most interesting steps are those from three to four communication partners and from four to five. That is because in a BMIN with 4×4 SEs, increasing the number of partners from three to four means that the fourth one must be located at another SE and thus, an additional network stage becomes involved. Increasing the number of communication partners from four to five leads in meshes to the situation that one of the partners is no longer a direct neighbor of the sending node.

In Figures 17 and 18, the local traffic is fed into to network with a high offered load of 1.0 while Figure 19 investigates a weak offered

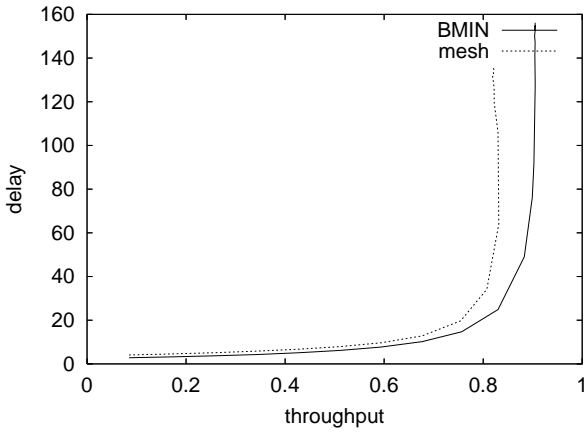


Figure 16: Multicast traffic: delay versus throughput

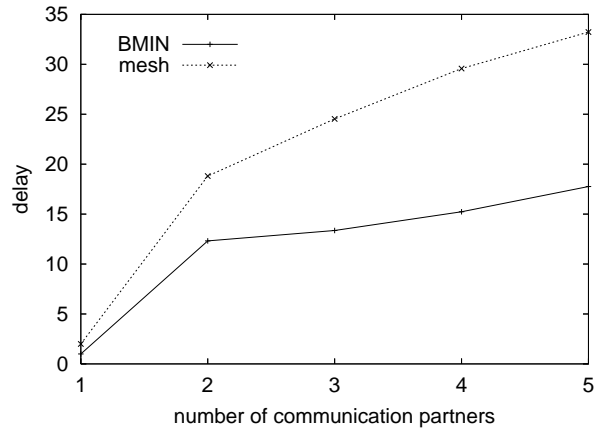


Figure 18: Delay dependent on the traffic locality

load of 0.1. If every node only communicates to a single neigh-

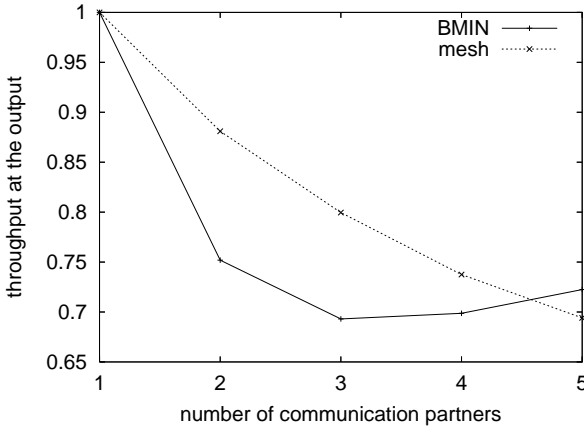


Figure 17: Throughput dependent on the traffic locality

bor (who is different to the partners of the other communications), then, the communication paths through the network do not interfere and the throughput is at maximum and the delay at minimum. No offered packets to the network are rejected and for an offered load of 1.0, a throughput of 1.0 results. The delay equals the number of hops needed in the network. That is for MINs, only a single SE is involved and one hop leads to the destination. In case of meshes, two SEs are involved (this one at the sender node and this one at the receiver node). Thus, two hops are needed leading to a delay of 2.

If more than only a single node is the destination of each communication, conflicts for the destinations occur and thus, blockings in the SEs in front of the destination node. In consequence, throughput decreases and delay increases. For multistage interconnection networks, the throughput grows again slightly if more than three nodes are the communication partners of each node (Figure 17): an additional network stage is needed for the communication as mentioned above. This additional stage offers redundancy and additional bandwidth. In case of four communication partners, only every fourth communication uses the additional stage and bandwidth. Due to the high network load in Figure 18 and therefore, due to the

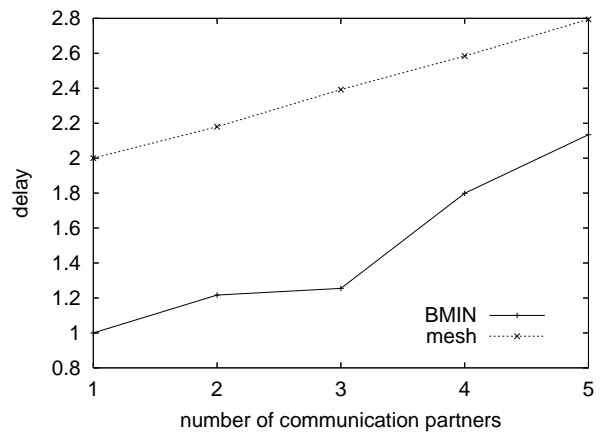


Figure 19: Traffic locality with weak traffic load

occupied buffers and high delay in the first network stage, the additional second stage delay of every fourth communication does not strongly influence the overall delay. In contrast, if the traffic load is weak (Figure 19) and delay is low, the step from three to four communication partners (and the additional stage delay) is clearly visible in the figure.

Comparing the mesh network and the BMIN, the delay of the BMIN outperforms the mesh for any investigated number of communication partners. The multistage interconnection network shows lower delay (less than 70% of the mesh's delay). The mesh reveals a higher throughput in case of a very strong locality in traffic. If more than four communication partners are involved, the BMIN's throughput becomes dominant.

5. CONCLUSION

The performance of two network-on-chip topologies was compared for use in multicore processors. Particularly, multicast traffic patterns and traffic localities were investigated which represent a significant part of multicore traffic. The performance evaluation was supported by the *CINSim* simulator. This simulator has been developed to model all kinds of network topologies that are based on atomic components such as buffers, routers, traffic generators and target buffers. The development of this simulator was driven

by the investigation of networks-on-chip. But off-chip networks can be examined as well.

The network performance was described in terms of throughput and delay. Specifically, a mesh topology was compared to a bidirectional interconnection network topology. Unicast traffic was used as well as multicast and local traffic, which both are an important part of the network traffic for evaluating multicore processors. The simulation results show that BMINS outperform meshes: meshes cope with delays that are about 30% higher than those of BMINS.

Besides the performance, the mean distance, the diameter, and the buffer cost were calculated for both network topologies. The NoC silicon area consumption is dominated by the buffers.

Again, the BMIN reveals better results except for the buffer cost. Nevertheless, the higher buffer cost is more than compensated by the higher performance, shorter distance and diameter of the BMIN. BMINS main advantage seems to be scalability as with the increase in the number of cores, the distance and diameter remain logarithmic.

6. REFERENCES

- [1] M. Alderighi, F. Casini, S. D'Angelo, D. Salvi, and G. R. Sechi. A fault-tolerant FPGA-based multi-stage interconnection network for space applications. In *Proceedings of the First IEEE International Workshop on Electronic Design, Test and Applications (DELTA'02)*, pages 302–306, 2002.
- [2] L. Benini and G. De Micheli. Networks on chips: A new SoC paradigm. *IEEE Computer*, 35(1):70–80, 2002.
- [3] D. Bertozzi, A. Jalabert, S. Murali, R. Tamhankar, S. Stergiou, L. Benini, and G. De Micheli. NoC synthesis flow for customized domain specific multiprocessor systems-on-chip. *IEEE Transactions on Parallel and Distributed Systems*, 16(2):113–129, Feb. 2005.
- [4] L. Bononi and N. Concer. Simulation and analysis of network on chip architectures: Ring, spidergon and 2d mesh. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE 2006)*, pages 154–159. ACM, 2006.
- [5] W. J. Dally and S. Lacy. VLSI architecture: Past, present, and future. In *Proceedings of the 20th Anniversary Conference on Advanced Research in VLSI*, pages 232–241, 1999.
- [6] W. J. Dally and B. Towles. Route packets, not wires: On-chip interconnection networks. In *Proceedings of Design Automation Conference (DAC 2001)*, pages 684–689, 2001.
- [7] C. de Rose and H.-U. Heiß. Dynamic processor allocation in large mesh-connected multicomputers. In *Proceedings of the EURO-PAR 2001; Manchester; Lecture Notes in Computer Science (LNCS 2150)*. Springer Verlag, 2001.
- [8] P. Guerrier and A. Grenier. A generic architecture for on-chip packet-switched interconnections. In *Proceedings of IEEE Design Automation and Test in Europe (DATE 2000)*, pages 250–256. IEEE Press, 2000.
- [9] G. J. Lipovski and M. Malek. *Parallel Computing: Theory and Comparisons*. John Wiley & Sons, New York, 1987.
- [10] D. Lüdtke, D. Tutsch, A. Walter, and G. Hommel. Improved performance of bidirectional multistage interconnection networks by reconfiguration. In *Proceedings of 2005 Design, Analysis, and Simulation of Distributed Systems (DASD 2005); San Diego*, pages 21–27. SCS, Apr. 2005.
- [11] S. Murali and G. De Micheli. SUNMAP: A tool for automatic topology selection and generation for NoCs. In *Proceedings of the 41st Design Automation Conference (DAC 2004)*, pages 914–919. ACM, 2004.
- [12] L. M. Ni, Y. Gui, and S. Moore. Performance evaluation of switch-based wormhole networks. *IEEE Transactions on Parallel and Distributed Systems*, 8(5):462–474, May 1997.
- [13] P. P. Pande, C. S. Grecu, A. Ivanov, and R. A. Saleh. Switch-based interconnect architecture for future systems on chip. In *Proceedings of the SPIE*, volume 5117, pages 228–237, 2003.
- [14] S. Stergiou, F. Angiolini, S. Carta, L. Raffo, D. Bertozzi, and G. De Micheli. xpipes lite: A synthesis oriented design library for networks on chips. In *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE'05)*, volume 2, pages 1188–1193. IEEE, 2005.
- [15] D. Tutsch. *Performance Analysis of Network Architectures*. Springer Verlag, Berlin, 1 edition, 2006.
- [16] Y. Yang and J. Wang. A class of multistage conference switching networks for group communication. *IEEE Transactions on Parallel and Distributed Systems*, 15(3):228–243, Mar. 2004.