Chip Multiprocessor Traffic Models Providing Consistent Multicast and Spatial Distributions

Dietmar Tutsch
Daniel Lüdtke
Technische Universität Berlin
Institute of Computer Engineering and Microelectronics
Sekr. FR 3-2, Franklinstr. 28-29
D-10587 Berlin, Germany
DietmarT@cs.tu-berlin.de

Chip multiprocessors (CMPs) have become the center of attention in recent years. They consist of multiple processor cores on a single chip. These cores are connected on-chip by a bus or, if many cores are involved, by an appropriate network. To investigate how a multicore processor behaves dependent on the chosen network-on-chip topology, a corresponding model must be established for performance evaluation. Modeling and simulating the entire system would lead to high model complexity. Thus, it is more reasonable to exclude the cores and to simply model stochastically the detached network. The cores are replaced by traffic generators which must provide reasonable CMP traffic. It usually consists of multicasts and a particular spatial distribution. Because the traffic is not known exactly, both multicasts and spatial traffic are described as stochastic distributions for model input. The easiest way is to specify the spatial distribution of the traffic and the kind of multicasts independently of each other. However, not all multicast distributions can be achieved with a particular desired spatial distribution and vice versa. It is therefore important to check for the compatibility of the spatial distribution and the multicasts that the modeler is willing to investigate. Such a compatibility check is provided by the algorithm presented in this paper. It prevents inconsistent traffic parameters while modeling.

Keywords: stochastic traffic modeling, network-on-chip, multicore processor, multicast distribution, spatial traffic distribution

1. Introduction

The ongoing improvement in very-large-scale integration (VLSI) technology has lead to an ever-increasing number of devices per chip. Since this increased density cannot be used to improve the performance of uniprocessor chips as in previous years, chip multiprocessors (multicore processors) are now the focus of interest [1].

To allow cooperating cores on such a multicore processor, an appropriate communication structure between them must be provided. In case of a low number of cores (e.g. a quad core processor), a shared bus may be sufficient. However, in the future, hundreds or even thousands of cores will collaborate on a single chip. More advanced network topologies will then be needed. Many topologies have been proposed for these so-called networks-on-chips (NoCs). An example is the multistage interconnection network (MIN) topology. Guerrier and Greiner [2] established a bidirectional MIN structure (equivalent to a fat tree) on a field-programmable gate array (FPGA). This on-chip network, with its particular router design and communication protocol, is referred to as Scalable, Programmable, Integrated Network (SPIN). The network operates by a wormhole-switching technique and with deterministic routing, although alternative paths exist in a bidirectional MIN. Its performance for different network buffer sizes has been compared.

Alderighi et al. [3] used MINs with the Clos structure. Multiple parallel Clos networks connect the inputs and the outputs to achieve fewer blockings. Again, FPGAs serve as a basis for realization.

Pande et al. [4] presented a fat tree network topology for NoCs. Virtual channels were introduced to reduce blockings. A VHDL model was established to estimate the area consumption on a chip.

SIMULATION, Vol. 84, Issue 2/3, February/March 2008 61–74

DOI: 10.1177/0037549708091638

Lahiri et al. [5] evaluated bus and ring topologies for NoCs. They investigated the performance of particular architectures of buses and rings dependent on spatial localities. Shared and hierarchical buses are used. Shared buses also build the basis of Wingard's research [6]. His SoC communication infrastructure, called STBUS, can also be used to set up crossbars.

Wiklund and Liu [7] proposed a mesh-based network-on-chip, named SoCBUS. It was developed especially for hard real-time embedded systems. Packets lock circuit parts while passing them. Another mesh NoC was developed by Kumar et al. [8]. Their project describes a design methodology for generating the mesh architecture and, in a second step, the application is mapped onto the mesh. Mesh networks were also investigated by Bononi and Concer [9]. They compared them to ring and spidergon networks. Besides homogeneous traffic, hot spot traffic was also applied for comparison. A prototype CMP with a mesh network interconnecting the cores has already been implemented by Intel [10].

Lee et al. [11] presented a star network architecture. It is realized as a globally asynchronous system. They compared it to several other topologies.

Sánchez et al. [12] described a reconfigurable NoC. In a two-dimensional torus topology, a node is able to exchange its position with a neighboring node.

To support the design of a network-on-chip for a particular application, some tools have been developed. They help in selecting a feasible network architecture and offer some assistance in hardware development. To map the communication demands of the cores onto predefined topologies such as meshes and MINs, Bertozzi et al. [13] developed a tool called NetChip (consisting of SUNMAP and ×pipes). This tool provides complete synthesis flows for NoC architectures. The authors investigated several topologies with their tool, including mesh, torus, hypercube and MINs.

Ching et al. [14] introduced a high-level NoC description language that eventually creates VHDL code. The related tool is also able to simulate the high-level description. Cycle-accurate performance results are obtained.

Most publications only consider unicast traffic in the NoC for choosing the optimal topology. It is obvious, however, 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.

Further on, 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. However, due to many available cores, more than a single application can be processed in parallel. There will then 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 outstandingly support multicast traffic and local traffic. Investigating whether networks are suitable for multicore processors is usually performed by stochastically modeling and simulating them. Therefore, offering multicast traffic and localities specifies an important feature of the modeling technique.

Since the traffic is not exactly known, both multicasts and spatial traffic are described as stochastic distributions for model input. The easiest way is to specify the spatial distribution of the traffic and the kind of multicasts independently of each other [15]. However, not all multicast distributions can be achieved with a particularly desired spatial distribution and vice versa. It is therefore important to check the compatibility of the spatial distribution with the multicasts that the modeler is willing to investigate. Such a compatibility check is provided by the algorithm presented in this paper. It prevents inconsistent traffic parameters while modeling chip multiprocessor NoCs. The basics of the algorithm are presented in Tutsch and Lüdtke [16]; this paper explains the algorithm more exhaustively and therefore reveals more features.

The paper is organized as follows. Section 2 introduces some architectures of chip multiprocessor NoCs. Traffic characteristics such as multicasts and localities are described in Section 3 where the modeling of such traffic is also discussed. Section 4 presents a new algorithm to check for the compatibility of the desired spatial distribution and the multicasts for modeling and simulation. Section 5 summarizes and provides conclusions.

2. Chip Multiprocessor NoCs

This section provides some examples for network-on-chip architectures [17] to connect the cores of a chip multiprocessor.

2.1 Mesh Networks

A static network architecture for NoCs is, for instance, a mesh [8, 7]. In such an architecture, the cores are located at the crosspoints of the mesh. 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). Due to the two-dimensional chip area, 2-D meshes (Figure 1) are a good option for NoC realizations.

The nodes of the 2-D mesh incorporate a core and a 5 × 5 switching element (Figure 1b), optionally with buffers in case of packet switching. The switching element (SE) can connect each input to each output of the node to allow messages to pass the node. Further on, the core of the node is linked via the SE to the rest of the mesh.

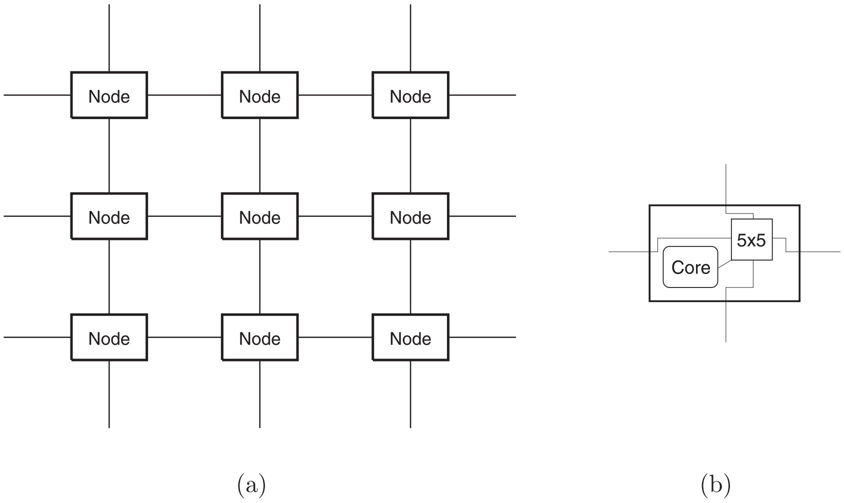


Figure 1. Two-dimensional mesh architecture: (a) mesh and (b) mesh node

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 1a). The number of links per node does not change if additional cores (i.e. nodes) are added to the mesh. Therefore, a mesh offers good scalability concerning its hardware structure. However, 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 is demanded by many connections and so blocking occurs. Messages are therefore mostly transferred by packet switching to deal with the blocking by introducing buffers.

2.2 Ring Networks

Another static network architecture for NoCs is given by a ring [5, 9]. With such an architecture each node is connected to exactly two other nodes, one on each side, leading to an overall structure of a closed loop (Figure 2). Having only two neighboring nodes keeps the amount of interfaces per node very small.

Messages are sent to the ring and usually circle in a common direction from node to node. Each node checks whether it is the receiver. The nodes of a ring look similar to Figure 1b except that (usually) only one single connection out of the node and one single connection into the node exists.

The main drawback is that the entire network is affected if any link fails. Further on, distances are far if messages are destined for nodes which are located close to the sender node in the negative circulation direction. Doubling each link reduces the problem of failures. Such an architecture is called dual ring. Allowing both (opposite) circulation directions reduces the distances. In the case of dual rings, two bidirectional connections from a network node to its environment exist. Similar blocking problems occur in rings as in meshes. Due to less links in the ring network, blocking may even occur more often.

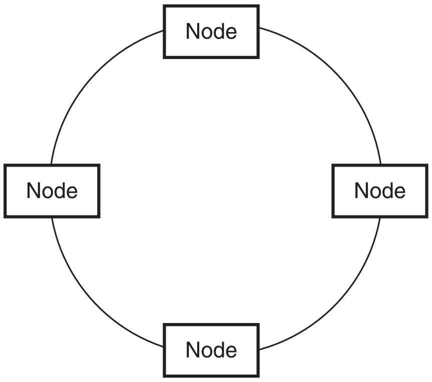


Figure 2. Ring architecture

2.3 Bidirectional MINs

Multistage Interconnection Networks [3, 18] are dynamic networks which are based on switching elements. SEs are arranged in stages and connected by interstage links. The link structure and size of the $c \times c$ SEs characterizes the MIN. MINs which connect N cores of a chip multiprocessor consist of at least $n = \log_c N$ stages to provide full connectivity.

In bidirectional MINs (BMINs), interstage links and their SEs operate in bidirectional mode. That means packets can be transferred in both directions. In consequence,

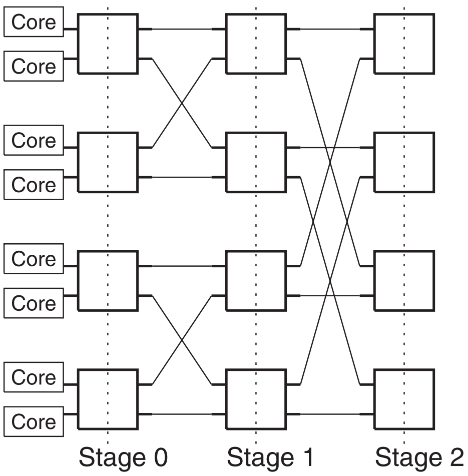


Figure 3. Bidirectional MIN

each core is only connected to a single input/output of the NoC. Figure 3 depicts the structure of a bidirectional MIN. If packet switching is applied, buffers can be introduced. A packet is first routed from the NoC input to the right, denoted as forward direction. As soon as it reaches a stage from which a path exists in the backward direction (from right to left) to its destination core, it turns around. This stage is called turnaround stage. Finally, the packet proceeds in backward direction to the desired core.

During its movement in forward direction, the packet may choose any arbitrary SE output because each SE output offers a path to the destination core via a turnaround stage. Moreover, all paths that a particular packet may choose reveal the same stage as turnaround stage due to the regular 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 NoC output.

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

3. CMP Traffic Characteristics

As discussed in the introduction, networks for chip multiprocessors must be able to deal with multicast traffic. Traffic localities will be observed and therefore must also be represented in a NoC model. Both issues will be discussed in the following.

3.1 Multicast Traffic

Multicasting can be efficiently performed by copying the packets within the switches of a multicore processor network instead of copying them before they enter the network. This scheme is called message replication while

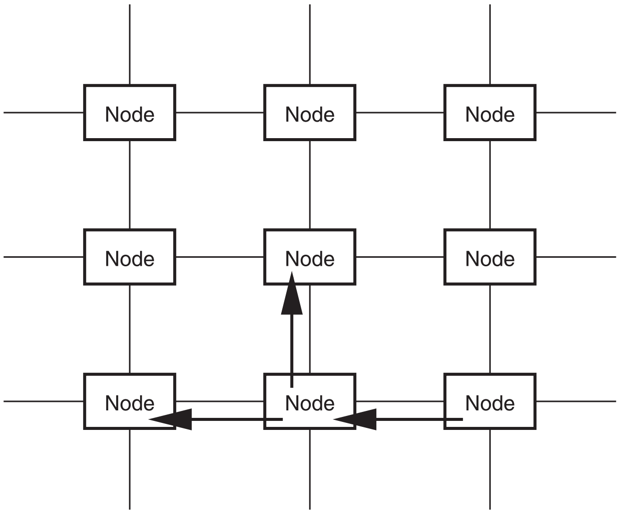


Figure 4. Multicast MRWR in mesh networks

routing (MRWR). For example, Figure 4 shows such a scenario for a mesh network.

A packet is sent by the lower rightmost core and is destined for the core in the center of the mesh and for the lower leftmost core. The packet is transferred as a single packet through the network. The copies for the different destinations are generated as late as possible. That means that the packet is not copied until it reaches the lower middle node. From there, the paths to the destinations differ and two copies of the packet proceed their way to the two destinations.

Message replication before routing (MRBR) applied to the example above would copy the packet at the sender core and send it twice into the mesh. Compared to this, applying MRWR copies the packets as late as possible to optimally reduce the number of packets in the network. MRWR also reduces the number of packets in most other NoC topologies, e.g. in BMINs and in ring networks.

3.2 Local Traffic

Two aspects of locality have to be considered. First, the locality of network traffic due to applications that are distributed to different sets of cores is important. 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 5 demonstrates the locality of bidirectional MINs [19]. The structural locality for Core 0 (connected to Input/Output 0) is depicted. There is a very high locality for Core 0 with Core 1 (dark gray area). The communication path is very short (just a turnaround at Stage 0).

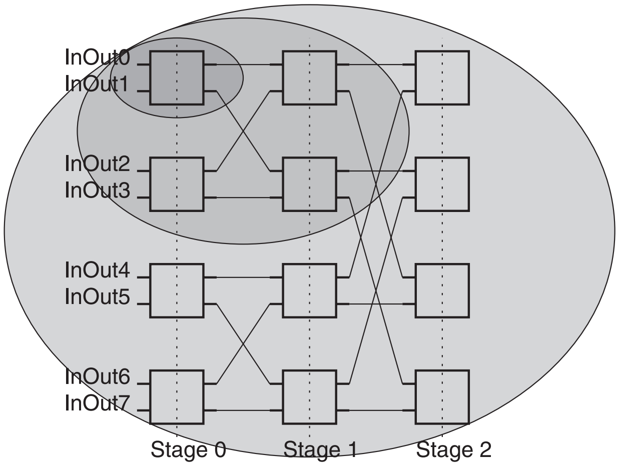


Figure 5. Locality in bidirectional MINs

A weaker locality can be found between Core 0 and Core 2 or Core 3 (medium gray area). Here, packets must pass three stages to reach 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 gray area). All network stages are involved.

In meshes and in ring networks, it is obvious that the communication path to neighboring cores is much shorter than e.g. the path between two cores in opposite corners of the mesh or the path between cores separated by half a ring, respectively.

In consequence, both aspects of locality should be mapped when applications are distributed to different cores. The cores are to be chosen such that they reveal structural locality resulting in fast communication. However, sometimes it may not be possible to choose 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 are locally connected.

3.3 Stochastic Traffic Modeling

To investigate how the behavior of a chip multiprocessor is dependent on the chosen network topology, a corresponding stochastic model can be established for performance evaluation. Modeling the entire system would lead to high model complexity. Thus, it is more reasonable to exclude the cores and to simply model stochastically the detached network. The cores are replaced by traffic generators which must produce reasonable CMP traffic. It usually consists of multicast and spatial traffic.

Because future CMP traffic is not exactly known, both multicasts and spatial traffic are described as stochastic distributions for model input. The easiest way is to specify the spatial distribution of the traffic and the kind of multicasts independently of each other [15]. But not all multicast distributions can be achieved with a particularly desired spatial distribution and vice versa: there are multicast distributions and spatial distributions that are incompatible.

For example, if the multicast distribution is specified such that only broadcasts to all network outputs occur, all outputs receive the same number of packets. As previously proposed, the spatial distribution is now specified independently of this multicast distribution. If the spatial distribution is defined, for example, such that a particular output receives twice as many packets as all others, it contradicts the broadcast specification which is that all outputs receive the same number of packets! For broadcasts, anything other than a uniform spatial distribution is feasible.

It is therefore important to check for the compatibility of the spatial distribution and the multicast distribution that the modeler is willing to investigate. Such a compatibility check avoids inconsistent traffic parameters while modeling.

4. Compatibility of Multicast and Spatial Distribution

In this section, an algorithm is invented that checks whether a network input is able to generate traffic with a given stochastic multicast distribution by fulfilling the desired stochastic spatial distribution. It is assumed that the investigated input can reach N outputs of the network. The multicast distribution is described by the multicast probabilities $a(i)$ with $i \in \mathbb{N}, 1 \leq i \leq N$. The multicast probability $a(i)$ gives the probability with which a generated packet is destined to i outputs. For instance, a network with $N = 4$ outputs may have a multicast distribution $a(1) = 0.1$, $a(2) = 0.6$, $a(3) = 0.1$ and $a(4) = 0.2$. This means that 10% of all incoming (generated) packets are unicast packets (destined to $i = 1$ output), 60% of all incoming packets are multicast packets (destined to $i = 2$ outputs), 10% are multicast packets (destined to $i = 3$ outputs) and 20% are broadcast packets (destined to all $i = 4$ outputs).

The spatial distribution is described by the local probabilities $\ell(h)$ with $h \in \mathbb{N}, 0 \leq h < N$. The local probability $\ell(h)$ gives the probability with which a generated packet is destined to Output h. Figure 6a gives an example for a network with $N = 4$ outputs: Output 0 is to be one of the packet destinations with a probability of 35% ($\ell(0) = 0.35$), Output 1 with 10% ($\ell(1) = 0.1$), Output 2 with 40% ($\ell(2) = 0.4$), and Output 3 with 15% ($\ell(3) = 0.15$).

4.1 Compatibility

To demonstrate the idea of the new algorithm, we first consider the particular multicast distribution where there

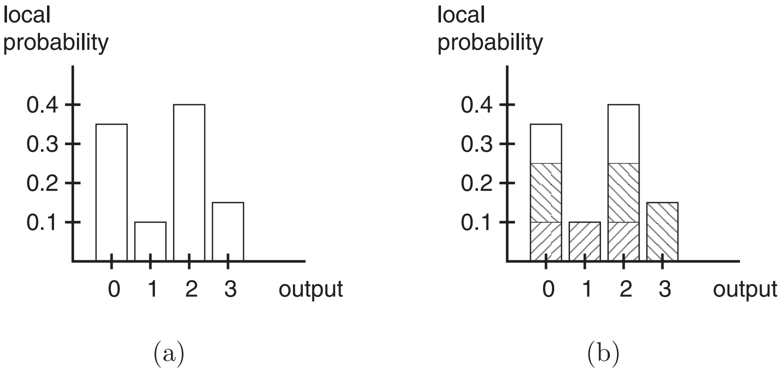


Figure 6. Spatial distribution for a network with 4 outputs: (a) desired distribution and (b) feasible with *a*(3)

are only multicasts to g outputs and unicasts (a single output):

$$a(i = g) := a(g) \tag{1}$$

$$a(1) := 1 - a(g) \tag{2}$$

$$a(1 \neq i \neq g) := 0. \tag{3}$$

The only critical issue is then the multicast to g outputs. It has to be checked if the multicasts can be performed in the required ratio without violating the given spatial distribution. The unicasts can always be distributed among the outputs for any given spatial distribution. Figure 6a gives an example.

Figure 6 depicts the desired spatial distribution (as mentioned above) for a network with 4 outputs. For instance, if there were *only* multicasts to $g = 4$ outputs (that means $a(4) = 1$ and $a(1) = 0$) then the distribution of Figure 6a could not be reached because all packets were destined to *all* outputs and that means the spatial distribution was uniform (all bars of Figure 6a are of the same height: 0.25).

A similar problem would arise if there were only multicasts to $g = 3$ outputs ($a(3) = 1$ and $a(1) = 0$). The uniform spatial distribution could then be avoided if e.g. half of the packets are sent to Outputs 0, 1 and 2 and half of them are sent to Outputs 0, 2 and 3. In this case, Outputs 0 and 2 would be the destination of packets twice as often as Outputs 1 and 3: the spatial distribution $\ell(0) = 0.\overline{3}$, $\ell(1) = 0.1\overline{6}$, $\ell(2) = 0.\overline{3}$ and $\ell(3) = 0.1\overline{6}$ would result. However, the distribution of Figure 6a could not be reached by this multicast distribution either. In the best case, the filled part of the bars (Figure 6b) would be feasible but not the unfilled part by multicasts to 3 outputs.

Consequently, the filled part of the bars of Figure 6b give the maximum amount of multicast traffic to 3 outputs (received by the network outputs) that is allowed if the given spatial distribution is to be fulfilled (the unfilled parts can be contributed by multicasts to 2 outputs and to a single output). The phrase 'amount of multicast traffic to i outputs' represents the fraction of received traffic (by all network outputs), divided by i, that originates from sending multicast traffic to i outputs.

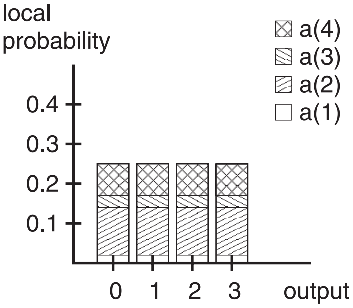


Figure 7. Example with uniform spatial distribution

4.2 Algorithm

The basic idea of the algorithm is based on the following observation: a given multicast to i outputs of the network can always be performed in a uniform spatial distribution. This means that each of the N network outputs receives the fraction of i/N packets (copies) per multicast packet entering the network. (Note that each multicast packet destined to i outputs generates i copies, including the original packet, while passing the network.)

As an example, Figure 7 shows such a multicast for a uniform spatial distribution. For this network with four outputs, a uniform spatial distribution is the desired one. Any multicast distribution $a(i)$ can achieve it. For example, in the depicted distribution where each output receives a copy of the incoming broadcast packets ($i = 4$), $3/4$ of the copies of the incoming multicast packets are sent to $i = 3$ outputs, $2/4$ of the copies of the incoming multicast packets are sent to $i = 2$ outputs and $1/4$ of the unicast packets is sent to $i = 1$ output.

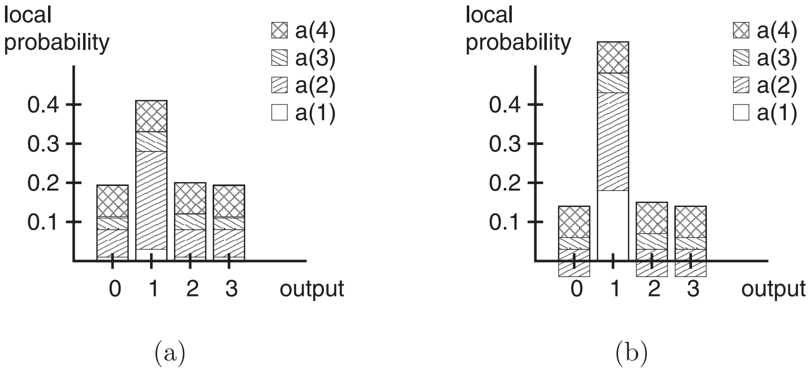


Figure 8. Optimal distribution among outputs: (a) feasible and (b) not compatible

This characteristic of a uniform spatial distribution is used for checking the compatibility of a given multicast distribution $a(i)$ to a desired spatial distribution. Step by step, each given multicast i is tested as follows. It is started with the desired spatial distribution and any multicast probability to $i = i_1$ outputs. This multicast to i_1 outputs is assumed to go to the b outputs (with $b \geq i_1$) of highest local probabilities.

In Figure 6, these b highest local probabilities would be Output 2 and Output 0 if $b \geq 2$, and additionally Output 3 if $b \geq 3$, and so on.

These b highest local probabilities are reduced according to this particular multicast probability. The number b is chosen such that the spatial distribution comes as close as possible to the uniform distribution after the described reduction of local probabilities. This allows the highest possible amount of multicast to another number i_2 of outputs.

With the reduced local probabilities, the next multicast probability (e.g. to $i = i_2$ outputs), is dealt with in the same way as above. It is assumed that it is directed to the outputs of the (now) highest local probabilities, and so on.

This reduction of the (previously calculated) local probabilities is performed for all multicasts i. If in any step the reduction leads to a negative local probability, it can be concluded that this reduction is not possible and therefore this multicast cannot be realized, i.e. the spatial distribution and the multicast distribution are not compatible.

Figure 8 shows how to distribute the multicasts. The multicast distribution is set to $a(1) = 0.1$, $a(2) = 0.6$, $a(3) = 0.1$ and $a(4) = 0.2$. In Figure 8a, the spatial distribution is given by $\ell(0) = 0.195$, $\ell(1) = 0.41$, $\ell(2) = 0.2$ and $\ell(3) = 0.195$. First, the probabilities of the spatial distribution are reduced by the multicast to $i = 4$ outputs. Because that is a broadcast, all outputs receive the same amount of packet and thus, all probabilities must be reduced by the same value, corresponding to $a(4) = 0.2$. (As already mentioned, this 20% is related to the input traffic. Since $i = 4$, each incoming packet results in four outgoing packets.) For $i = 3$, the probabilities are further reduced such that Output 1 is reduced the most trying to reach a uniform distribution. All multicast output sets include Output 1 and two of the others. (A uniform distribution of Outputs 0, 2 and 3 can now be reached but Output 1 cannot be reduced very much). For $i = 2$ the same procedure is repeated: Output 1 is always one of the two multicast destinations. Because the unicast ($i = 1$) is able to fulfill any distribution, the remaining probabilities can be covered by it and the desired spatial distribution is compatible to the given multicast distribution.

In Figure 8b, the spatial distribution is slightly changed in the way that the probabilities of Outputs 0, 2 and 3 are decreased by 0.05 (5%) and Output 1 is increased by this traffic. It can be seen that one fails to distribute the multicast to $i = 2$ outputs (after $i = 4$ and $i = 3$ have been applied). Even if Output 1 is always one of the two multicast outputs, the second one must be chosen from the others. However, there are not enough resources left to cover this 60% multicast traffic to two outputs. The resulting distribution would lead to negative probabilities.

After these short examples, the algorithm is explained in detail now. First the local probabilities $\ell(h)$ are sorted such that $\ell(h_1) \leq \ell(h_2) \leq \ldots \leq \ell(h_N)$. That means all h_j with $1 < j < N$ must be determined such that

$$\{h_j\} = \{h | \ell(h_{j-1}) \leq \ell(h) \leq \ell(h_{j+1})\} \tag{4}$$

and

$$\{h_1\} = \{h | \ell(h) \leq \ell(h_2)\}, \tag{5}$$

$$\{h_N\} = \{h | \ell(h_{N-1}) \leq \ell(h)\}. \tag{6}$$

Then, for each non-zero multicast probability $a(i)$, the following steps are performed as outlined above. First, the number b of outputs is chosen such that the spatial distribution comes as close as possible to the uniform distribution after the described reduction of local probabilities. If b is chosen too small, one of the b local probabilities falls below one of the remaining $N - b$ local probabilities.

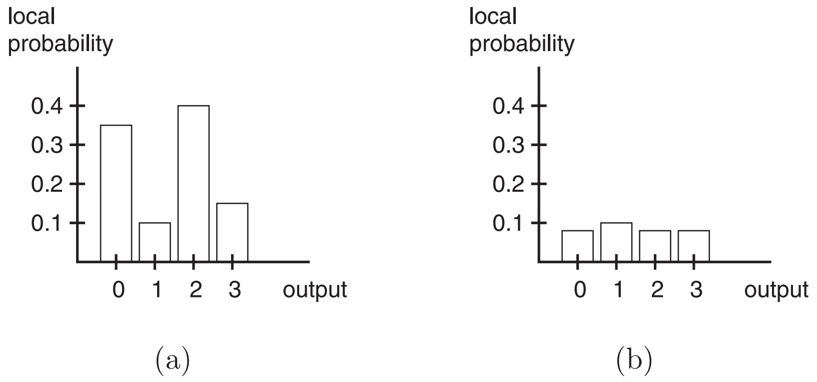


Figure 9. Violation of the algorithm: (a) desired distribution and (b) reduction of three highest probabilities

Figure 9 demonstrates a violation of the previously presented method. In the desired spatial distribution (Figure 9a), the $b = 3$ outputs of the highest local probabilities (Outputs 0, 2 and 3) are reduced by a certain required amount (assumed 0.55 overall) leading to Figure 9b. Now, the formerly lowest probability of Output 1 is higher than the others. A better solution could have been found by choosing $b = 4$ and decreasing all four probabilities to the same level, which means reaching a uniform distribution. In conclusion, if the maximum number of highest outputs is known for which the reduction fails in the previously described way, this leads to the searched b by simply incrementing this maximum number by one.

The maximum number for which reduction fails is found as follows. The overall reduction of the b local probabilities by a multicast to i outputs is given by $i \cdot a(i)/\sum_{g=1}^{N} g \cdot a(g)$. The normalization of $a(i)$ by $\sum_{g=1}^{N} g \cdot a(g)$ must be done because all probabilities $a(i)$ are related to the network inputs but the spatial probabilities $\ell(h)$ are related to the outputs. With probability $a(g)$, an incoming packet is copied g times in the network and g packets reach the output. Thus, $i \cdot a(i)/\sum_{g=1}^{N} g \cdot a(g)$ gives the output ratio of packets generated by a multicast to i outputs. This output traffic ratio is considered to be directed to the b outputs of highest local probability. Decreasing these local probabilities overall by the given output traffic ratio must not lead to a probability falling below one of the remaining $N - b$ outputs as described above. Therefore, b is determined by the following equation considering that each of the highest local probabilities is allowed to be decreased by $a(i)/\sum_{g=1}^{N} g \cdot a(g)$ at most (otherwise, there would exist multicasts where an output is involved more than once):

$$b = \begin{cases} N & \text{if } i = N \\ b_{\max} & \text{else} \end{cases} \tag{7}$$

with

$$\begin{aligned} b_{\max} &= \max_{i \le b_x \le N} \left\{ b_x \,\middle|\, \frac{i \cdot a(i)}{\sum_{g=1}^{N} g \cdot a(g)} \right. \\ &\geq \sum_{j=N-b_x+2}^{N} \min\left\{ \ell(h_j) - \ell(h_{N-b_x+1}), \right. \\ &\qquad \left. \left. \frac{a(i)}{\sum_{g=1}^{N} g \cdot a(g)} \right\} \right\}. \end{aligned} \tag{8}$$

It is obvious that for a multicast to $i = N$ outputs, all $b = N$ local probabilities must be decreased.

Equation (7) gives the number of outputs to which the multicasts to i outputs are assumed to be directed to. Thus, their local probabilities must be reduced to an average value of

$$\ell_{\text{av}} = \frac{\sum_{j=N-b+1}^{N} \ell(h_j) - \frac{i \cdot a(i)}{\sum_{g=1}^{N} g \cdot a(g)}}{b}. \tag{9}$$

In other words, the local probability $\ell(h_j)$ with $N - b < j \le N$ must be reduced by $\ell(h_j) - \ell_{\text{av}}$. But as already mentioned, the maximum amount of reduction must not exceed $a(i)/\sum_{g=1}^{N} g \cdot a(g)$ to avoid an output to be involved more than once. In consequence, the reduction of Output h_j is given by

$$\ell_r \leftarrow \min\left\{ \ell(h_j) - \ell_{\text{av}}, \frac{a(i)}{\sum_{g=1}^{N} g \cdot a(g)} \right\} \tag{10}$$

and a reduced local probability results: $\ell(h_j) \leftarrow \ell(h_j) - \ell_r$. If ℓ_{av} was not reached because a reduction of only $a(i)/\sum_{g=1}^{N} g \cdot a(g)$ took place, the number of remaining local probabilities (denoted by b^*) that have to be reduced must cover the missing difference and their average value to be reached must be adapted to $\ell_{\text{av}} \leftarrow \ell_{\text{av}} - (\ell(h_j) - \ell_{\text{av}})/b^*$. That is why the reduction must start with the highest local probability (i.e. $j = N$) and end with the lowest (i.e. $j = N - b + 1$).

If ℓ_{av} results in a negative value or there are no more remaining local probabilities to cover a missing difference ($b^* = 0$), then this demonstrates that the given spatial distribution cannot deal with the given multicast distribution and they are incompatible.

If all multicast probabilities $a(i)$ can be handled without showing any incompatibility, the entire multicast distribution is compatible to the spatial distribution. In this case, the algorithm can even additionally determine a spatial distribution $\ell_i(h)$ for the multicasts to each number i of outputs which overall fulfill the spatial distribution $\ell(h)$. The probability $\ell_i(h_j)$ of Output h_j being a destination of a multicast to i outputs is simply the reduction ℓ_r for this multicast and output as calculated above. Additionally, it must be normalized considering the message replication in the network from input to output due to the multicasts:

$$\ell_i(h_j) = \ell_r \cdot \frac{\sum_{g=1}^{N} g \cdot a(g)}{a(i)}. \tag{11}$$

However, one must be aware that there is usually no unique solution for the spatial distributions $\ell_i(h)$ and Equation (11) gives only one of all solutions.

Figure 10 summarizes the entire algorithm. The algorithm was validated by comparing the results with a second approach for checking compatibility. This second approach [20] is based on closed form equations. These equations describe the problem in a reverse way. For a given multicast distribution, they allow the calculation of a single compatible spatial distribution determined by some input parameters. Inverting such a calculation leads either to the related input parameters or to no solution because no input parameters exist for the given multicast and spatial distribution; they are then incompatible.

Unfortunately, the types of equations only allow inversion and solution by a computer algebra system. Due to the complexity of the equations, the computer algebra system has not been able to produce results for a larger number N of network outputs. Both approaches yield results for small values of N, however. In contrast to Lüdtke and Tutsch [20], the algorithm presented in this paper returns results for any number N.

4.3 Examples

Two examples demonstrate the algorithm. The first example network consists of $N = 4$ outputs, e.g. a 4 × 4 BMIN. The desired spatial distribution is that shown in Figure 11a: $\ell(0) = 0.195$, $\ell(1) = 0.41$, $\ell(2) = 0.2$ and $\ell(3) = 0.195$. The multicast distribution is given by $a(1) = 0.1$, $a(2) = 0.6$, $a(3) = 0.1$ and $a(4) = 0.2$. It is now investigated whether an input that can reach all outputs is able to generate network traffic that fulfills both the desired spatial distribution and the given multicast distribution.

Ordering the local probabilities leads to $\ell(h_1 = 0) = 0.195 \leq \ell(h_2 = 3) = 0.195 \leq \ell(h_3 = 2) = 0.2 \leq \ell(h_4 = 1) = 0.41$. The algorithm starts with the multicast to e.g. all $i = N = 4$ outputs (but any other i is also possible). That means the $b = N$ highest local probabilities must be reduced. Their average value is desired to be

$$\begin{aligned} \ell_{\mathrm{av}} &= \frac{(0.195 + 0.195 + 0.2 + 0.41) - \frac{0.2}{2.4} \cdot 4}{4} \\ &= 0.1\overline{6} \end{aligned} \tag{12}$$

as shown in Figure 11a by the dashed line.

However, this value cannot be reached if the probabilities were to reduce for more than the maximum allowed reduction (given by the fact that no output is allowed to appear more than once in a multicast message). The maximum allowed reduction is obtained by $a(4)/\sum_{g=1}^{N} g \cdot a(g) = 0.08\overline{3}$. This is why Output $h_j = h_4 = 1$ is only reduced for the filled area to $\ell(1) = 0.32\overline{6}$ and the desired ℓ_{av} is not reached. In consequence, the remaining three outputs must also cover the missing amount and a new $\ell_{\mathrm{av}} = 0.1\overline{6} - (0.32\overline{6} - 0.1\overline{6})/3 = 0.11\overline{3}$ is determined (dotted line in Figure 11a).

For Output $h_3 = 2$, this is again not possible and the maximum reduction of $0.08\overline{3}$ leads to $\ell(2) = 0.11\overline{6}$ and a new $\ell_{\mathrm{av}} = 0.11\overline{3} - (0.11\overline{6} - 0.11\overline{3})/2 = 0.111\overline{6}$ (slightly below the dotted line in Figure 11a) for the remaining two outputs. This value can be reached by Output $h_2 = 3$ as well as by Output $h_1 = 0$ by a reduction of $0.08\overline{3}$ of both $\ell(3) = \ell(0) = 0.111\overline{6}$.

Calculating a spatial distribution $\ell_4(h)$ for the multicast to $i = 4$ outputs according to Equation (11) results in $\ell_4(1) = \ell_4(2) = \ell_4(3) = \ell_4(0) = 1$ (in the order of calculation). This is obvious because this multicast is a broadcast and all four outputs must be a destination.

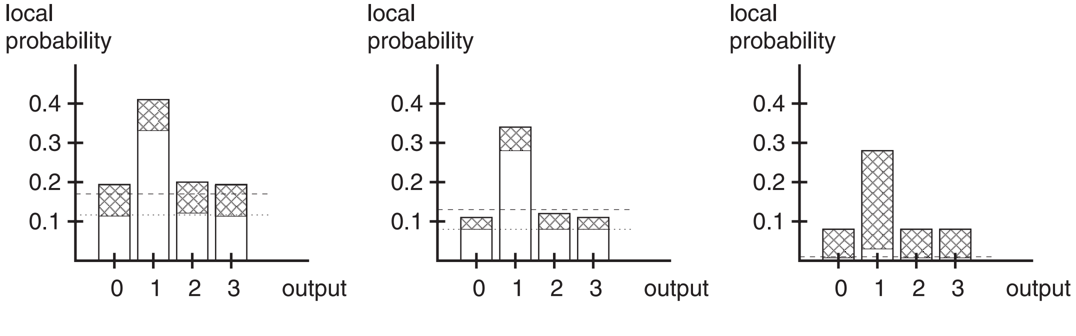
Now, multicasts to $i = 3$ outputs are investigated using the remaining local probabilities as depicted in Figure 11b. The maximum b_x for which Equation 7 is fulfilled is $b_x = 4$. This means that again the $b = 4$ highest local probabilities (i.e. all) must be reduced. The average value is now desired to be

$$\begin{aligned} \ell_{\mathrm{av}} &= \frac{(0.111\overline{6} + 0.111\overline{6} + 0.11\overline{6} + 0.32\overline{6}) - \frac{0.1}{2.4} \cdot 3}{4} \\ &= 0.135417 \end{aligned} \tag{13}$$

as shown in Figure 11b by the dashed line. But this value cannot be reached if the probabilities were to reduce for more than the maximum allowed reduction of $a(3)/\sum_{g=1}^{N} g \cdot a(g) = 0.041\overline{6}$. Again, Output $h_j = h_4 = 1$ cannot reach ℓ_{av} and is only reduced by the maximum allowed value to $\ell(1) = 0.285$. To cover this, the three remaining outputs try to reach a new average value given by

```
Data: multicast distribution $a(i)$ with $i \in \mathbb{N}, 1 \leq i \leq N$
Data: spatial distribution $\ell(h)$ with $h \in \mathbb{N}, 0 \leq h < N$
Result: compatibility of multicast and spatial distribution
Result: spatial distributions $\ell_i(h)$ for multicasts to $i$ outputs
begin
    letall $\ell_i(h) \leftarrow 0$ ;                    /* initialization of result */
    $\{h_j\} \leftarrow \begin{cases} \{h|\ell(h) \leq \ell(h_2)\} & \text{if } j = 1 \\ \{h|\ell(h_{N-1}) \leq \ell(h) & \text{if } j = N \\ \{h|\ell(h_{j-1}) \leq \ell(h) \leq \ell(h_{j+1})\} & \text{else} \end{cases}$   with $1 \leq j \leq N$ ;
    /* sort local probabilities */
    foreach $a(i) > 0$ do
                         /* for each non-zero multicast probability */
        $b \leftarrow \begin{cases} N & \text{if } i = N \\ \max_{i \leq b_x \leq N} \left\{ b_x \middle| \frac{i \cdot a(i)}{\sum_{g=1}^{N} g \cdot a(g)} \geq \sum_{j=N-b_x+2}^{N} \right. \\ \left. \quad \times \min\{\ell(h_j) - \ell(h_{N-b_x+1}), \frac{a(i)}{\sum_{g=1}^{N} g \cdot a(g)}\} \right\} & \text{else} \end{cases}$ ;
        /* number of local probabilities to be equalized */
        $\ell_{\text{av}} \leftarrow \frac{\sum_{j=N-b+1}^{N} \ell(h_j) - \frac{a(i)}{\sum_{g=1}^{N} g \cdot a(g)} \cdot i}{b}$ ;      /* target mean of $b$ local probabilities */
        for $j \leftarrow N$ downto $N - b + 1$ do   /* for the $b$ highest local probabilities */
            $\ell_r \leftarrow \min\left\{ \ell(h_j) - \ell_{\text{av}}, \frac{a(i)}{\sum_{g=1}^{N} g \cdot a(g)} \right\}$ ;   /* reduct. to mean or with allowed maxim. */
            $\ell(h_j) \leftarrow \ell(h_j) - \ell_r$ ;         /* local probability reduced */
            $\ell_i(h_j) \leftarrow \ell_r \cdot \frac{\sum_{g=1}^{N} g \cdot a(g)}{a(i)}$ ;         /* spatial distribution for multicast to $i$ outputs */
            if $\ell(h_j) > \ell_{\text{av}}$ then     /* average could not be reached */
                $\ell_{\text{av}} \leftarrow \begin{cases} \ell_{\text{av}} - \frac{\ell(h_j) - \ell_{\text{av}}}{b-N-1+j} & \text{if } b - N - 1 + j \neq 0 \\ -1 & \text{else} \end{cases}$ ;
                /* result: new mean */
                if $\ell_{\text{av}} < 0$ then /* negative mean cannot be reached */
                    output INCOMPATIBILITY ;   /* distributions are not compatible! */
                endif
            endif
        endfor
    endfch
end
```

Figure 10. Short sketch of the algorithm



(a) (b) (c)

Figure 11. Desired average value and real reduction for multicasts to i outputs (Example 1): (a) $i = 4$; (b) $i = 3$; and (c) $i = 2$.

$\ell_{\text{av}} = 0.135417 - (0.285 - 0.135417)/3 = 0.08\overline{5}$ (dotted line in Figure 11b). This value can be reached by all three outputs resulting in $\ell(2) = \ell(3) = \ell(0) = 0.08\overline{5}$.

The spatial distribution $\ell_3(h)$ for the multicast to $i = 3$ outputs results in $\ell_3(1) = 1$ (i.e. Output 1 is always one of the destinations), $\ell_3(2) = 0.74\overline{6}$, and $\ell_3(3) = \ell_3(0) = 0.62\overline{6}$ (in the order of calculation).

Figure 11c shows the remaining local probabilities for multicasts to $i = 2$ outputs. The maximum b_x for which Equation 7 is fulfilled is again $b_x = 4$. The average value is desired to be $\ell_{\text{av}} = 0.01041\overline{6}$, but this value cannot be reached if the probabilities were to reduce for more than the maximum allowed reduction of 0.25. This value limits the reduction of Output $h_j = h_4 = 1$ resulting in $\ell(1) = 0.035$. Thus, the remaining outputs get a new desired average value of $\ell_{\text{av}} = 0.00\overline{2}$, which can be reached by reducing them by $0.08\overline{3}$ resulting in $\ell(2) = \ell(3) = \ell(0) = 0.00\overline{2}$.

The spatial distribution $\ell_2(h)$ for the multicast to $i = 2$ outputs results in $\ell_2(1) = 1$ and $\ell_2(2) = \ell_2(3) = \ell_2(0) = 0.\overline{3}$. Each multicast is destined to Output 2 and, with equal probability, one of the other outputs.

Finally, the unicasts ($i = 1$) are dealt with. Of course, all $b = 4$ remaining local probabilities must be considered and their resulting average value must drop to $\ell_{\text{av}} = 0$. This can be reached because unicast traffic can fulfill any spatial distribution, as already mentioned. The spatial distribution $\ell_1(h)$ for the unicasts must then be $\ell_1(1) = 0.84$ and $\ell_1(2) = \ell_1(3) = \ell_1(0) = 0.05\overline{3}$.

The results of the algorithm show that the spatial and the multicast distributions are compatible and can be used in a simulation. The simulation results will be reasonable.

A second example will demonstrate how incompatible distributions are discovered. The multicast distribution is identical to this in the first example and the spatial distribution is changed only slightly: $\ell(0) = 0.19$, $\ell(1) = 0.42$, $\ell(2) = 0.2$, $\ell(3) = 0.19$, $a(1) = 0.1$, $a(2) = 0.6$, $a(3) = 0.1$ and $a(4) = 0.2$. The same ordering of the local probabilities as in the previous example results: $\ell(h_1 = 0) = 0.19 \leq \ell(h_2 = 3) = 0.19 \leq \ell(h_3 = 2) = 0.2 \leq \ell(h_4 = 1) = 0.42$. The algorithm starts again with the multicast to all outputs, e.g. $i = N = 4$. The average value of the $b = N$ highest local probabilities is desired to be

$$\ell_{\text{av}} = \frac{(0.19 + 0.19 + 0.2 + 0.42) - \frac{0.2}{2.4} \cdot 4}{4} = 0.1\overline{6} \qquad (14)$$

as shown in Figure 12a by the dashed line.

Again, the maximum reduction is given by $0.08\overline{3}$ leading to $\ell(1) = 0.33\overline{6}$. A new $\ell_{\text{av}} = 0.11$ results for the remaining three local probabilities (dotted line in Figure 12a). Due to the maximum reduction, $\ell(2) = 0.11\overline{6}$ is obtained and $\ell_{\text{av}} = 0.10\overline{6}$ for the remaining two local probabilities which is reached by them: $\ell(3) = \ell(0) = 0.10\overline{6}$. The spatial distribution for the broadcasts (multicasts to $i = N = 4$ outputs) results in $\ell_4(1) = \ell_4(2) = \ell_4(3) = \ell_4(0) = 1$.

The multicasts to $i = 3$ outputs must also be distributed among all $b = 4$ (highest) local probabilities. Reducing them by this multicast is desired to lead to $\ell_{\text{av}} = 0.135417$ (dashed line in Figure 12b). The maximum allowed reduction is given by $0.041\overline{6}$ leading to $\ell(1) = 0.295$ and a new $\ell_{\text{av}} = 0.08\overline{2}$ (dotted line in Figure 12b). This value can be reached by the remaining outputs: $\ell(2) = \ell(3) = \ell(0) = 0.08\overline{2}$. The spatial distribution for the multicasts to $i = 3$ outputs results in $\ell_3(1) = 1$, $\ell_3(2) = 0.82\overline{6}$ and $\ell_3(3) = \ell_3(0) = 0.58\overline{6}$.

The multicast to $i = 2$ outputs again gives $b = 4$. The desired average value of the reduced local probabilities is then $\ell_{\text{av}} = 0.01041\overline{6}$ (dashed line in Figure 12c). Due to the maximum allowed reduction of 0.25, it cannot be reached by Output 1: $\ell(1) = 0.045$ is obtained. That means that Output 1 is always one of the destinations of the multicasts to $i = 2$ outputs: $\ell_2(1) = 1$. The second destination must be one of the three remaining outputs. The second destination is chosen with equal probability among them, and leads to an local probability of

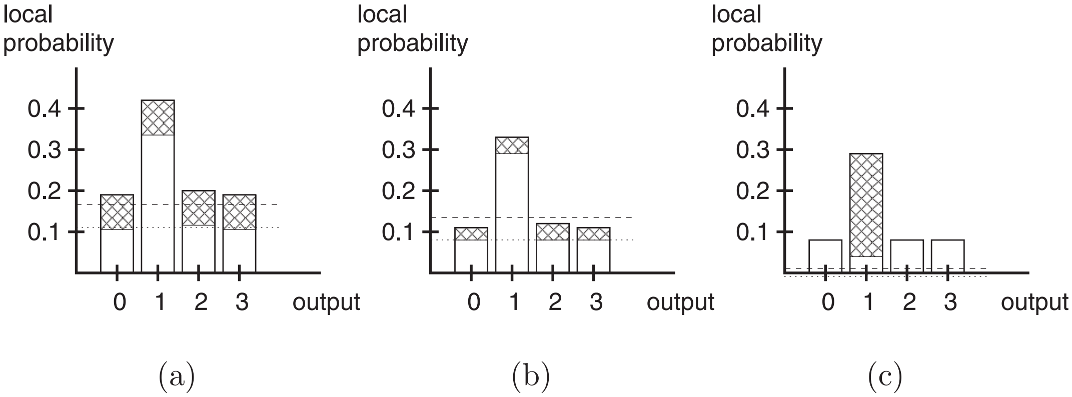


Figure 12. Desired average value and real reduction for multicasts to i outputs (Example 2): (a) $i = 4$; (b) $i = 3$; and (c) $i = 2$

$\ell(2) = \ell(3) = \ell(0) = \ell_{\mathrm{av}} = -0.00\overline{1}$ which must be reached. However, this is a negative value and is therefore not feasible as a local probability. In consequence, this means that the spatial and the multicast distributions are incompatible with each other. Starting simulations with these distributions would lead to results that are not reasonable and of no use.

4.4 Benefit

With the given algorithm, the modeler becomes able to check whether a particular desired spatial traffic distribution and multicast traffic distribution are compatible. Any inconsistent traffic parameters are prevented before the simulation or analysis of the model starts.

5. Conclusions

This paper addressed the modeling of chip multiprocessor traffic, which is necessary for performance evaluation of such systems. Chip multiprocessor traffic usually consists of some spatial distribution and some multicast distribution. Defining both distributions independently of each other for the model is the easiest way. Unfortunately, there are sometimes dependencies and some distributions are not compatible.

In this paper, a new algorithm for checking the compatibility of a given stochastic spatial traffic distribution and stochastic multicast traffic distribution is presented. This compatibility check avoids inconsistent traffic parameters while modeling.

The new algorithm can be performed before the simulation or analysis of the NoC in question is started. If the algorithm detects any incompatibility of the traffic distributions, simulation or analysis is aborted and the modeler is able to correct the traffic parameters.

Besides networks-on-chip, the algorithm can also be applied to off-chip networks with multicast traffic, e.g. to networks of parallel computer architectures.

In future work, our goal is to incorporate a more powerful stochastic traffic generator in the simulator *CINSim* [15]. The toolkit *CINSim* is a component-based interconnection network simulator for performance evaluation. A stochastic traffic generator is to be developed which fulfills a desired multicast and spatial traffic distribution at the network outputs. Currently, the traffic distributions at the network inputs can be chosen arbitrarily, but the distributions at the outputs differ from the inputs due to the dependencies. The new algorithm will help to characterize these dependencies and to draw conclusions for reverse calculation i.e. finding an input distribution to obtain a desired output distribution.

6. References

[1] Dally, W.J. and S. Lacy. 1999. VLSI Architecture: Past, Present, and Future. In *Proceedings of the 20th Anniversary Conference on Advanced Research in VLSI*, 232–241.

[2] Guerrier, P. and A. Grenier. 2000. A Generic Architecture for On-Chip Packet-Switched Interconnections. In *Proceedings of IEEE Design Automation and Test in Europe (DATE 2000)*, IEEE Press, 250–256.

[3] Alderighi, M., F. Casini, S. D'Angelo, D. Salvi, and G.R. Sechi. 2002. 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)*, 302–306.

[4] Pande, P.P., C.S. Grecu, A. Ivanov, and R.A. Saleh. 2003. Switch-Based Interconnect Architecture for Future Systems on Chip. In *Proceedings of the SPIE*, vol. 5117, 228–237.

[5] Lahiri, K., S. Dey, and A. Raghunathan. 2001. Evaluation of the Traffic-Performance Characteristics of System-on-Chip Communication Architectures. In *Proceedings of the 14th International Conference on VLSI Design (VLSID '01)*, IEEE Press, 29–35.

[6] Wingard, D. 2001. MicroNetwork-Based Integration for SoCs. In *Proceedings of the Design Automation Conference (DAC 2001)*, ACM, 673–677.

[7] Wiklund, D. and D. Liu. 2003. SoCBUS: Switched Network on Chip for Hard Real Time Embedded Systems. In *Proceedings of the 17th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2003)*, IEEE Press, 78–85.

[8] Kumar, S., A. Jantsch, J.-P. Soininen, M. Forsell, M. Millberg, J. Öberg, K. Tiensyrjä, A. Hemani. 2002. A Network on Chip Ar-

chitecture and Design Methodology. In *Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI'02)*, 105–112.

[9] Bononi, L. and N. Concer. 2006. 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)*, ACM, New York, 154–159.

[10] Azimi, M., N. Cherukuri, D. Jayasimha, A. Kumar, P. Kundu, S. Park, I. Schoinas, and A.S. Vaidya. 2007. Integration Challenges and Tradeoffs for Tera-scale Architectures. *Intel Technology Journal* 11(3), 173–184.

[11] Lee, S.-J., S.-J. Song, K. Lee, J.-H. Woo, S.-E. Kim, B.-G. Nam, and H.-J. Yoo. 2003. An 800MHz Star-Connected On-Chip Network for Application to Systems on a Chip. In *Proceedings of 2003 IEEE International Solid-State Circuits Conference (ISSCC 2003)*, IEEE Press, 468–475.

[12] Sánchez, J.L. and J.M. García. 2000. Dynamic Reconfiguration of Node Location in Wormhole Networks. *Journal of Systems Architecture* 46(10), 873–888.

[13] Bertozzi, D., A. Jalabert, S. Murali, R. Tamhankar, S. Stergiou, L. Benini, and G. De Micheli. 2005. NoC Synthesis Flow for Customized Domain Specific Multiprocessor Systems-on-Chip. *IEEE Transactions on Parallel and Distributed Systems* 16(2), 113–129.

[14] Ching, D., P. Schaumont, and I. Verbauwhede. 2004. Integrated Modeling and Generation of a Reconfigurable Network-On-Chip. In *Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS 2004)*, 139–145.

[15] Tutsch, D., D. Lüdtke, A. Walter, and M. Kühm. 2005. CINSim – A Component-Based Interconnection Network Simulator for Modeling Dynamic Reconfiguration. In *Proceedings of the 12th International Conference on Analytical and Stochastic Modelling Techniques and Applications (ASMTA 2005); Riga*, IEEE/SCS, 132–137.

[16] Tutsch, D. and D. Lüdtke. 2007. Spatial Distributions versus Multicast Distributions: Traffic Modeling of Chip Multiprocessor Networks. In *Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2007); San Diego*, SCS, Erlangen, 512–520.

[17] Tutsch, D. 2006. *Performance Analysis of Network Architectures*. Springer Verlag, Berlin.

[18] Tutsch, D. and G. Hommel. 2006. High Performance Low Cost Multicore NoC Architectures for Embedded Systems. In *Proceedings of the International Workshop on Embedded Systems – Modeling, Technology and Applications*, Springer-Verlag, Berlin, 53–62.

[19] Lüdtke, D., D. Tutsch, A. Walter, and G. Hommel. 2005. Improved Performance of Bidirectional Multistage Interconnection Networks by Reconfiguration. In *Proceedings of 2005 Design, Analysis, and Simulation of Distributed Systems (DASD 2005); San Diego*, SCS, Erlangen, 21–27.

[20] Lüdtke, D. and D. Tutsch. 2006. Combining Spatial and Multicast Traffic Distributions for Stochastic Modeling and Simulation of Networks-on-Chip. In M. Becker, H. Szczerbicka, eds., *Proceedings of the 19th Symposium on Simulation Technique (ASIM 2006)*, SCS, Erlangen, 241–246.

***Dietmar Tutsch** received the diploma degree Dipl.-Ing. in Electrical Engineering from the University of Saarbrücken (Germany) in 1993. In 1998 and in 2005, he received his Ph.D. and his Habilitation degree, respectively, in computer science from TU Berlin. There, he now holds a professorship in architecture of embedded systems.*

***Daniel Lüdtke** received the degree Dipl.-Ing. in Computer Engineering from the TU Berlin in 2003. He is currently a Ph.D. student and research assistant at the Institute of Computer Engineering and Microelectronics, TU Berlin.*