FAIR RATE PACKET ARBITRATION IN NETWORK-ON-CHIP

Falko Guderian, Erik Fischer, Markus Winter, Gerhard Fettweis

Vodafone Chair Mobile Communications Systems Technische Universität Dresden, 01062 Dresden, Germany E-mail: {falko.guderian, erik.fischer, winter, fettweis}@ifn.et.tu-dresden.de

ABSTRACT

This paper proposes two new arbitration techniques to enable fair link bandwidth allocation. One technique is a weighted round-robin scheme with weights based on the number of contending flows at the input port. The second technique is an age-based scheme with probabilistic arbitration where the traversed packet distance approximates age. Opposed to existing work, both schemes reach almost absolute fairness of link bandwidth allocation and simplify calculation of arbitration metrics.

I. INTRODUCTION

Network-on-chip (NoC) enables to easily scale from multi-processor system-on-chip (MPSoC) towards many-core systems [1][2][3][4][5]. As numerous processors (cores) will compete for the same network resources and congestion has to be avoided, arbitration becomes a critical router functionality. The arbiter grants an output port to one of multiple input port requests. Izu [6] observed the problem that applications with non-uniform traffic pattern, e.g., all cores share a single memory interface, show a NoC latency which is dependent on the location of the core where the application is executed. Hence, link bandwidth allocation becomes more unfair the longer the routes are. Lee et al. [7] define the issue of fair link bandwidth allocation as equality of service (EoS). EoS allows bandwidth allocation to become fair for each core and starvation is getting avoided. Especially applications which contain many parallel tasks sharing the same successor task will experience significantly shorter execution time. Unlike quality of service (QoS) [8][9], EoS does not provide services and guarantees for packet latency or bandwidth allocation. The goal is to develop fast and slight arbitration mechanisms which provide optimal EoS performance.

We observed that existing arbitration schemes have limitations regarding EoS. For example, traditional round-robin (RR) arbitration is providing insufficient EoS because it considers only local fairness for each router. Approximated age-based packet arbitration, introduced by Lee [7], provides better EoS but it has limitations regarding fairness of bandwidth allocation. This motivates us to introduce two fair rate packet arbitration schemes which allow for better EoS. The first technique is a weighted RR arbitration scheme with weights based on the number of contending flows at the input port. The second technique is an approximated age-based arbitration scheme, similar to Lee, with modified weight calculation. It belongs to the class of probabilistic arbitration and age is approximated only by the traversed distance of a flow. Both arbitration schemes are implemented into a flitlevel cycle-accurate NoC simulation environment and the performance is compared with traditional RR and Lee's approximated aged-based packet arbitration [7]. To measure the EoS performance, we look at two communication-intensive case studies: a synthetic (non-uniformly distributed) scenario applying different injection rates; a realistic H.264 decoder application scenario.

After introducing to global vs. local fairness in Section II, we review related work in Section III and explain both fair rate packet arbitration schemes in detail in Section IV. Then, Section V describes the system model and simulation environment. In Section VI, we demonstrate the experiments and corresponding results. Section VII concludes our work.

II. GLOBAL VS. LOCAL FAIRNESS

EoS needs to ensure fairness when link bandwidth is distributed amongst numerous cores in the NoC. Hence, link bandwidth is considered to be a limited resource. We distinguish between two different arbitration mechanisms to study fairness of link bandwidth allocation: port arbitration and packet arbitration.

Fair port arbitration ensures local fairness so that each port of the router (RT) is equally selected to forward available flits. This does not necessarily imply global fairness, meaning all packets are equally treated, as illustrated via the example in Figure 1. Therein, RR port arbitration has been applied and



Fig. 1: 5-ary 1D-mesh example to demonstrate global unfairness of round-robin port arbitration.

capacity is equal for all links. Packets are indicated by the source identifiers. If each of processing elements (PEs) 1-4 concurrently transfers packets to PE5, the problem occurs that PE4 already comes with 50% of the link bandwidth at the destination. PE3 is provided 25% and PE1-2 get 12.5% of the link bandwidth. In the example, global fairness is not achieved for all packets because the closer a source is located to the destination, the more bandwidth is available for packet transmission. In contrast, fair rate packet arbitration guarantees all currently transferring PEs to allocate an equal portion of the link bandwidth. Therefore, global fairness is ensured.

III. RELATED WORK

A. Port Arbitration

Fairness is targeted in several port arbitration schemes. The aim is to allow the same fairness as RR port arbitration by simultaneously reducing arbitration processing delay and area. For example, Yang et al. [10] combine distributed ping pong arbitration with an input request masking approach. This allows fast and low-power arbitration and approximates RR fairness. As already mentioned in Section II, port arbitration is not able to support fair bandwidth allocation on packet level.

B. Packet Arbitration

Das et al. [11] follows a prioritized packet arbitration scheme which is based on application ranking using a "stall-time-criticality" heuristic. Monitoring stall times and assigning port priorities is relatively complex especially at run-time. Hence, it implies more computational effort and larger processing delay compared to our approach. For that reason, we did not consider the scheme for comparison.

Abts et al. [12] propose age-based packet arbitration in large-radix k-ary n-cubes which is able to provide EoS and simultaneously reduces packet latency. Because this approach would be quite complex for on-chip realization, Lee et al. [7] approximate agebased packet arbitration. Therein, EoS is achieved using probabilistic packet arbitration which is based on weights composed of distance and contention degree, i.e., number of flows contending for the same output port. We introduce a similar age-based packet arbitration scheme which considers only the traversed distance of a flow. In addition, our weighted RR arbitration scheme is a deterministic technique which uses weights based on the number of contending flows at each input port. Hence, we simplify calculation of arbitration metrics either with respect to deterministic weights or less input variables.

IV. FAIR RATE PACKET ARBITRATION SCHEMES A. Weighted Round-Robin Approach

We propose, fair rate packet arbitration, a simple modification of the RR-robin scheme that allows for global fairness by prioritizing the input ports. Our straightforward approach tracks the number of sources that share every input port. This number is further used as numerical weight to prioritize each input port. The more sources share an input port, the higher the priority of this port has to be chosen. Because this can be formulated as linear relationship the following slight adaptation of the RR mechanism, presented as pseudo-code (Algorithm 1), is sufficient to realize the prioritization.

Algorithm 1 Fair Rate Arbitration Algorithm	
1:	for each output port <i>o</i> do
2:	i = roundRobinArbit(RRP(o))
3:	newSource = sourceAddress($headFlit(i)$)
4:	updateTimeWindow(newSource)
5:	propagateFlit($i \rightarrow o$)
6:	// Update Round Robin Pointer
7:	W(i) = numSourcesInTimeWindow()
8:	increment($C(o)$)
9:	if $W(i) \leq C(o)$ then
10:	increment($RRP(o)$) mod $(N_i - 1)$
11:	C(o) = 0
12:	end if
13:	end for

Basically, the arbitration algorithm uses the RR mechanism that selects for every output port o an input port *i* according to the RR pointer RRP (line 2 of Algorithm 1). Usually, the pointer is incremented after every cycle, so that every input port is treated equally and is granted the highest priority once every N_i cycles (where N_i is the number of input ports). We slightly modified this part of the RR algorithm, so that RRP may stay assigned to the same input port for multiple cycles giving this input port a higher priority. The priority, i.e. the number of cycles that a specific input port *i* stays selected, is given by the weights W(i) (line 7). Thus, we need to store the number of cycles that the current selected input port was already granted the connection to an output port. For this purpose, we introduced a cycle counter C(o)for every output port that is incremented once a new flit has been propagated to the output port (line 8). If this counter exceeds the weight of the selected input port (line 9), the RR pointer RRP is updated (line 10)



Fig. 2: Example for fair rate packet arbitration concept.

and the cycle counter is reset to zero (line 11).

The weights W(i) are directly related to the number of sources that share the corresponding input port. Determining the weights is the main challenge of this algorithm, since the number of sources that share an input port cannot be assumed constant for a dynamic traffic scenario. For this purpose, we track the source addresses of the input flows, obtained from the head flit of the selected input port (line 3 of Algorithm 1), within a predefined time window (line 4). Afterwards, the flit is propagated from the selected input port to the corresponding output port (line 5). Finally, the weights W(i), on which the *RRP* update mechanism is based, are updated (line 7) based on the number of different source addresses that are observed within the time window.

Figure 2 picks up the small 5-ary chain example from Section II to demonstrate the improved behavior of the modified RR scheme. In router RT2, two input ports contend for the same output port: one port served from PE1 and the other port served from PE2. Hence, both ports are treated equally with the same priority. The resulting output sequence of RT2 is as follows: 2,1,2,1,.... In RT3, two input ports are contending for the same output port. In this case, the second input port is served by two PEs (PE1 and PE2), while the first input port is still supplied by a single PE (PE3). Correspondingly, the weights are set to 1x and 2x respectively. The resulting output sequence of RT3 is: 3,2,1,3,2,1,.... From the final sequence at the output of RT4 it is easy to conclude that every flow gets assigned an equal bandwidth of 25%, independent of its origin which was definitely not the case for the example presented in Section II.

B. Modified Age-Based Packet Arbitration

In contrast to the algorithm presented in the previous subsection, the arbitration scheme presented by Lee [7] is not based on the RR mechanism. Rather, a probabilistic decision is made for selecting an input port for every output port. For providing EoS, a non-uniform distribution is used for the probabilistic decision. The distribution is given by weights W(i)that are determined for each input port, similar to our approach in Section IV-A. The heuristic formula for the computation of the weights is given by $W(i) = C^{h(i)}$ (method of Variably Increasing Weights [7]) where Crepresents the number of contending flows that have to be propagated to the same output port and h(i)is the already traversed distance of the flow that is pending at the corresponding input port i.

We will show that this prioritization significantly improves EoS but does not yield an optimal solution for fair bandwidth allocation. In fact, the algorithm would only be optimal, if the total number of contending flows that share the output port tends to infinity. This number depends on the topology and traffic pattern and is usually rather small. By analyzing this algorithm, we found an even more simple solution that is able to provide a nearly optimal solution to the problem. We modified the equation for the weights computation as follows: $W(i) = h(i)^N$ where N is an arbitrary and unique constant. We found that by using this weighting, EoS will be improved, the higher we choose N. In contrast to the total number of contending flows, we are free to choose the constant parameter N at our own discretion. Consequently, the computation of weights can be simplified by making it independent of C. We can go even further and think of $N \to \infty$. In this case, the arbitration mechanism would always just select the input port where the flow is pending with the longest traversed distance. This equals an agedbased arbitration scheme, where the age would simply be defined by the traversed distance. According to the observed trend for $N \to \infty$, this simple arbitration mechanism should provide an optimal solution for the fair bandwidth allocation problem (we have not yet implemented the $N \to \infty$ solution and cannot give a definitive statement on this). In Section VI, we demonstrate the performance of the proposed modification of Lee's algorithm for N = 10.

V. SYSTEM MODEL AND SIMULATION ENVIRONMENT

The simulation environment is based on a flit-level, cycle-accurate C++ simulator framework, provided by [13], which was extended to support fair rate packet arbitration. The system model, illustrated in Figure 3, is the basis of the simulations.

Due to the common usage, our work focuses on deterministic XY routing. The arbitration logic within the routers has to resolve contention situations where multiple input ports simultaneously requesting a connection to the same output port. The RR mechanism that has been implemented in the router model of the simulation framework is a very well known technique [14][15]. As discussed before, RR is not able to provide global fairness. Referring to Section IV, we propose two packet arbiters that enable global fairness. The packet switching technique also influences the arbitration mechanism. Wormhole switching has emerged to the most favored switching technique for best effort services in network-on-chips [16], due to its



Fig. 3: System model overview.

low buffer requirements and its low end-to-end delay. Though the wormhole switching method is simple to realize, it has a strong influence on the arbitration. Hence, we want to introduce a second more "objective" technique that we call flit-based switching. In contrast to wormhole switching that only allows to toggle the input port at packet borders, flit-based switching enables the arbiter to toggle the input port at flit-level, i.e., even within packets. We consider the flitbased switching only for evaluation purpose and use it as reference for the performance measurements.

Concerning the router throughput our system model assumes that one flit of every input port can be routed per clock cycle as long as no contention occurs. In the system model, illustrated in Figure 3, PEs have no processing functionality. They are implemented as pure data sources that are able to generate packets internally according to traffic commands that contain information on the traffic scenarios (e.g. uniform, limited distance, etc.) and absolute packet injection times. The traffic commands are provided via externally generated traffic command files (TCFs, refer to Figure 3). This allows the simulation environment to analyze specific application scenarios provided by external tools.

In heavy network congestion, flit injection can fail at the requested injection time. In this case the flit is stored in the NoC interface (NoCIF, see Figure 3) and sent as soon as possible.

VI. RESULTS

In the following, we present experimental results on the comparison of our two new packet arbitration schemes with RR arbitration and Lee's probabilistic arbitration using a synthetic and a realistic H.264 decoder application scenario. Since Lee proposed different methods for determining the weights, we selected the best performing method (Variably Increasing Weights [7]) for the comparison.

A. Experimental Setup

To consider larger networks containing also long routes, we choose the 8x8 mesh topology in both sce-

narios. For the sake of clarity and briefness, we introduce following abbreviations for the analyzed arbitration schemes: round-robin (RR), fair-rate round-robin (Fair/RR), Lee's probabilistic arbitration (Ch/Prob), Lee's algorithm with modified weights (hN/Prob).

Synthetic application scenario

In this scenario, PEs are injecting packets with exponentially distributed interarrival times. Packets are send to the same destination, corresponding to a single shared memory interface. This type of traffic model emphasizes the unfair bandwidth allocation problem of the RR scheduler guite good, since it contains many chain-like traffic sub scenarios, refer to Figure 1. We observed that the more uniform the traffic distribution gets, the less fair-rate arbiters have impact on EoS. This is because we are not able to reach the network throughput limit even at an injection rate of (near) 1 for real uniform traffic. Only in situations where more bandwidth is required than available, the advantages of EoS become obvious. We use XY routing with wormhole and flit-based switching, respectively, and compare Fair/RR against RR. The Fair/RR window size for tracking packet sources was heuristically set to 50. Furthermore, we analyzed and compared it against Ch/Prob and the modified version hN/Prob for N=10. Per simulation run, each PE injects 1000 packets with a length of 10 flits each. Injection rates from 0.001 to 1 have been considered. We analyzed such low injection rates, since the network throughput, respectively acceptance rate, already saturates at 0.015 (1/64). This is because all PEs have the same target and thus contend at the target (bottleneck) router

H.264 application scenario

H.264, a popular multimedia standard requires high processor performance. Because enough parallelism is available on frame-, slice- and macro-block level, execution performance can be accelerated using NoC-based MPSoC. In this work, we use a H.264 (QCIF) decoder application benchmark provided as task graph model by [17]. It includes many parallel tasks sharing the same successor task. Packet injection times are derived via design space exploration, provided by [18], by mapping the application onto the NoC architecture and not considering the communication network. The benchmark uses only macro block parallelism to reduce data size and processor performance requirements. Multiple, concurrent running H.264 applications are simulated. Earliest deadline first policy is used to schedule each decoder and competing applications are prioritized via least laxity policy. The NoC architecture contains a single DMA to which a processor is able to read/write data packets.

B. Results for Synthetic Scenario

In the following, we present performance results for the four arbitration schemes using the synthetic scenario described above. Since we target a fair band-



Fig. 4: Distance-specific acceptance rates for injection rate 1.0.

width allocation independent of the distance, we classified all received packets according to the distance that they have passed and computed the acceptance rate (as a measure for the provided bandwidth) for each of those classes. The acceptance rate A is computed as follows. Let T_{SIM} be the total simulation time interval (in clock cycles). The acceptance rate A can be determined by the following equation:

$$A = \frac{N_F}{T_{SIM} \cdot N_{PE}}.$$
 (1)

The parameter N_F and N_{PE} , representing the total number of flits sent within the simulation interval and the number of PEs, respectively, and are known from the simulation setup. In Figure 4, we present the results for the arbitration schemes RR, Fair/RR, Ch/Prob and hN/Prob for flit-based and wormhole routing respectively at an injection rate of 1.0.

The high injection rate makes the effect of the arbitration schemes clearly visible, since the required bandwidth is much higher than the available bandwidth. This is where the EoS aware arbiters can prove their strength. The classical RR (blue curves with circle markers) is clearly unfair as Figure 4 shows. While short range flows get assigned a very high bandwidth of 6% at 4 hops and even 33% at 2 hops (due to scaling not shown in the figure), long distance flows get assigned a very low link bandwidth (0.06% at 10 hops) or even suffer of starvation (above 12 hops). The Fair/RR arbitration (green curves with cross markers) shows a very fair behavior. Every PE gets assigned the same bandwidth in this case independent of its distance to the target PE: 1.56% (1/64). The Ch/Prob arbiter (red curves with triangle markers) in turn shows a very interesting behavior. In contrast to RR that penalizes long distance flows, Ch/Prob now even favors the long range flows, granting them a higher bandwidth to the disadvantage



Fig. 5: Jain's fairness index for link bandwidth distribution at multiple injection rates.

of the short range flows. However, the unfairness is significantly reduced compared to RR. Thus, a link bandwidth of 3% is assigned to the long range flows of 16 hops while the short range flows of 2 hops get an acceptance rate of 0.7%. Referring to Figure 4, our modified probabilistic arbitration scheme hN/Prob (black curves with square markers) finally offers nearly optimal EoS that is comparable to the Fair/RR arbiter. We can also conclude from the figure that the fairness behavior is independent of the applied packet switching method (i.e. flit-based or wormhole).

The previous analysis only covers the behavior of the arbitration schemes at an injection rate of 1.0. Certainly, we must also check how the algorithms perform at different lower injection rates. For a clear presentation of these results, we introduce Jain's fairness index [19], a commonly used measure to evaluate fairness in networks. For this purpose, we classify the received packets according to the distance (in # hops) that they have traversed. For each of these "hop classes" *d* we determine the acceptance rate A_d according to Equation (1). Now, we apply Equation (2) to aggregate these values into a single value: Jain's fairness index.

$$\mathcal{J}_{acc} = \frac{(\sum_{i=1}^{i=N} A_i)^2}{N \cdot \sum_{i=1}^{i=N} A_i^2}.$$
 (2)

Jain's index gives a clear impression of the fairness that an arbitration scheme can provide. It yields a value of 1 for perfect EoS. The minimum value is 1/N (where N is the number of different hop classes which is 15 in our 8x8 2D-mesh case) and means that only a single hop class is served while all other hop classes starve. The result of the analysis for multiple injection rates from 0.001 up to 1 is shown in Figure 5.

We can see that all four arbitration schemes provide a fair bandwidth allocation up to the point of network saturation which is at an injection rate of around 1.5% for our architecture and traffic scenario.



Fig. 6: Distance-specific acceptance rates for H.264 application scenario

From this point on, the fairness of the RR scheme starts decreasing very fast and drops down to around 0.16 which, intuitively spoken means, that only one sixth of the hop classes is still served fair while the rest completely starves. The Ch/Prob arbitration also starts decreasing after the network saturation point is exceeded. However, it decreases only down to 0.9 and thus provides a much better fairness compared to RR. The arbiter Fair/RR as well as hN/Prob feature an absolute fair behavior concerning bandwidth distribution that is independent of the injection rate and hence show the best performance results. We can conclude that the behavior that we observed in detail for an injection rate of 1 is also confirmed for multiple lower injection rates.

C. Results for H.264 Application Scenario

To demonstrate the applicability of our approach for real application scenarios, we simulated the network traffic based on an H.264 decoder application that is executed on multiple PEs in parallel. The injection rate has been measured to be approximately 2.5%. The simulation results for the scenario are given in Figure 6 in form of the distance-specific acceptance rates.

We can confirm the same behavior that we already observed at Figure 4, even though the trends are less pronounced due to the lower injection rate. RR yields the worst EoS and favors short distance flows ($\mathcal{J}_{acc} =$ 0.65). In contrast, Ch/Prob provides a significantly better, but not perfect EoS, since it slightly penalizes the short distance flows ($\mathcal{J}_{acc} = 0.98$). Fair/RR as well as hN/Prob once again prove nearly perfect EoS $(\mathcal{J}_{acc} = 1.00)$. Hence, our fair rate arbitration schemes provide EoS under realistic application conditions.

VII. CONCLUSION AND FUTURE WORK

In this paper, we have presented two new packet arbitration schemes that allow all cores to fairly allocate the available link bandwidth in a network-onchip, also referred to as EoS. We have analyzed two communication-intensive case studies: a synthetic scenario applying different injection rates and a realistic H.264 decoder application scenario. From the experiments, we conclude that our two arbitration techniques are able to prevent starvation significantly better than existing work. The main contribution of this paper is the introduction of two fair rate packet arbitration schemes which simplify calculation of arbitration metrics either with respect to deterministic weights or less input variables. They additionally provide better EoS compared to existing arbitration techniques reaching almost absolute fairness of link bandwidth allocation. In future work, we will cover more diverse applications. In addition, we want to analyze hardware implementations and provide according synthesis results.

References

- [1] P. Guerrier and A. Greiner, "A generic architecture for on-chip packet-switched interconnections," in Proc. of DATE, 2000.
- A. Hemani, A. Jantsch, S. Kumar, A. Postula, J. Öberg, M. Milberg, and D. Lindquist, "Network on a chip: An architecture for billion transistor era," in *Proc. of NorChip*, 2000.
 W. Dally and B. Towles, "Route packets, not wires: on-chip interconnection networks," in *Proc. of DAC*, 2001.
 R. Ho, K. Mai, and M. Horowitz, "The future of wires," *Proceedings of the IEEE*, vol. 89, no. 4, p. 490504, April 2001.
- [3]
- [4] *ceedings of the IEEE*, vol. 89, no. 4, p. 490504, April 2001. L. Benini and G. De Micheli, "Networks on chips: a new soc
- [5] paradigm," *Computer*, vol. 35, no. 1, pp. 70–78, January 2002. C. Izu, "Throughput fairness in k-ary n-cube networks," in *Proc.*
- C. Izu, "Through of ACSC, 2006. [6]
- M. M. Lee, J. Kim, D. Abts, M. Marty, and J. W. Lee, "Probabilis-[7] tic distance-based arbitration: Providing equality of service for many-core cmps," in Proc. of MICRO, 2010.
- [8] K. Goossens, J. van Meerbergen, A. Peeters, and R. Wielage, 'Networks on silicon: combining best-effort and guaranteed services," in Proc. of DATE, 2002.
- [9] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "QNoC: QoS architecture and design process for network on chip," J. Systems Architecture, vol. 50, no. 2-3, pp. 105–128, February 2004
- [10] S.-W. Yang, M.-H. Sheu, C.-K. Yeh, C.-Y. Wen, C.-C. Lin, and W.-K. Tsai, "Fast fair crossbar scheduler for on-chip router," in *Proc. of ISCAS*, 2007. [11] R. Das, O. Mutlu, T. Moscibroda, and C. Das, "Application-
- aware prioritization mechanisms for on-chip networks," in Proc. of MICRO, 2009.
- [12] D. Abts and D. Weisser, "Age-based packet arbitration in large-radix k-ary n-cubes," in *Proc. of SC*, ser. SC '07, 2007.
 [13] M. Winter and G. P. Fettweis, "A network-on-chip channel allo-
- cator for run-time task scheduling in multi-processor systemon-chips," Digital Systems Design, Euromicro Symposium on, vol. 0, pp. 133–140, 2008.
- P. Gupta and N. McKeown, "Designing and implementing a [14] fast crossbar scheduler," Micro, IEEE, vol. 19, no. 1, pp. 20 -28, 1999.
- [15] Y.-L. Lee, J. M. Jou, and Y.-Y. Chen, "A high-speed and decentralized arbiter design for noc," in Proc. of AICCSA, 2009
- [16] R. Marculescu, J. Hu, and U. Ogras, "Key research problems in noc design: a holistic perspective," in Proc. of CODES+/SSS, 2005.
- Www.vodafone-chair.com/research/embedded benchmarks.
- [18] B. Ristau, T. Limberg, and G. Fettweis, "A mapping framework based on packing for design space exploration of heterogeneous mpsocs," J. Signal Process. Syst., vol. 57, pp. 45-56, Oct 2009.
- [19] D. C. R. Jain and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," in DEC Research Report TR-301, September 1984.