

# Minimizing Virtual Channel Buffer for Routers in On-chip Communication Architectures

Mohammad Abdullah Al Faruque and Jörg Henkel  
 University of Karlsruhe, CES - Chair for Embedded Systems, Karlsruhe, Germany  
 {alfaruque,henkel} @ informatik.uni-karlsruhe.de

## Abstract

We present a novel methodology for design space exploration using a two-steps scheme to optimize the number of virtual channel buffers (buffers take the premier share of the router in a NoC [10]) used to implement logical channels multiplexed across the physical channel in a router output port for QoS supported on-chip communication. In the first step, the number of virtual channels is minimized during the mapping of tasks to the NoC at the design time of a System on Chip (SoC) for which we use a swarm intelligence-based Ant Colony Optimization (ACO) algorithm. In the second step, a probabilistic approach based on the traffic model of the application is used to further minimize the number of virtual channels. We achieve on average 90.2% reduction in the number of virtual channels compared to a fixed state-of-the-art (i.e. QNoC [1]) allocation for the E3S embedded application benchmark suit. The reduction depends on the designer and the QoS parameter; and it is dependent on the specific application driven traffic model. We demonstrate our design space exploration by means of a complete robot application and also extend our exploration by evaluating the E3S embedded application benchmark suit.

## 1 Introduction and Related Work

In one facet of the bi-directional embedded system's development market, the digital convergence of multiple complex applications as well as new critical applications (software side) in single terminal demand for higher computational power. On the other facet of the development, the semiconductor industries (i.e. Intel projects the availability of 100 billion transistors on a 300mm<sup>2</sup> die by 2015 [2]) allows to introduce thousands of processors or equivalent logic gates on a single chip (hardware side) to fulfill the computational demand. Typically, a Multi-Processor System on Chip (MPSoC) architecture is built by exploiting off-the-shelf standard components and uses multiple high performance hierarchical buses for establishing communication among components (bus-based system design). The bus-based system design has been seen incapable of meeting the design challenges for the next generation systems mainly for scalability and the imbalance between gate delays and wire delays on chip in the deep submicron (DSM) domain [4, 8]. Recently, the paradigm shift to communication-centric design has been emerged as an important step toward the MPSoC architecture.

Networks on Chips (NoCs) as an on-chip communication concept, its design methodology and key research problems have been discussed in [4, 8, 11, 14]. To handle the

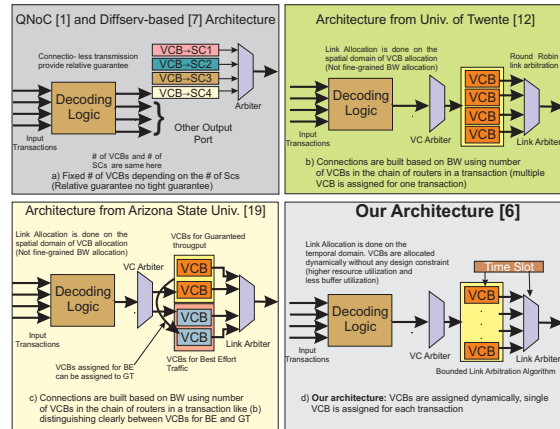


Figure 1: Architectural difference of state-of-the-art service-class-based architecture to our router architecture

real-time requirements of the applications Quality of Service (QoS) is considered to be one of the critical research issues [7, 16]. TDMA-based Aetheral-like [16] architectures provide tight guarantees but suffer lower resource utilization. On the other hand, service-class-based approaches (i.e. QNoC [1]) enjoy higher resource utilization but fail to ensure 100% tight guarantees.

Our QoS-supported scheme has taken the advantages of both service-class-based and connection-based approaches in [6] and it has some novel architecture level differences compared to the current service-class-related state-of-the-art NoC works. Our scheme uses wormhole switching scheme like [1, 12, 19] and utilizes virtual channel implementation. A *Virtual Channel* (VC) is a unidirectional logical or virtual connection between the tiles multiplexed with other VCs across the physical channel. Each VC is realized by an independently managed pair of message buffers referred to as *Virtual Channel Buffer* (VCB). Previous work related to service-class-based architectures have used a fixed number of VCBs in each output port and have not considered an application specific VCB allocation. Existing service-class-related works can be broadly classified into two parting directions. The first type of architecture presented in [1] has used a fixed number of service classes (see Fig. 1(a)). Each service class has a particular VCB in each output port. It does not provide 100% tight guarantee for each of the transaction and does not have a clear and fine-granular classification for each transaction. Therefore, the bandwidth is not allocated fairly to increase resource utilization.

The second group of QoS-supported service-class-

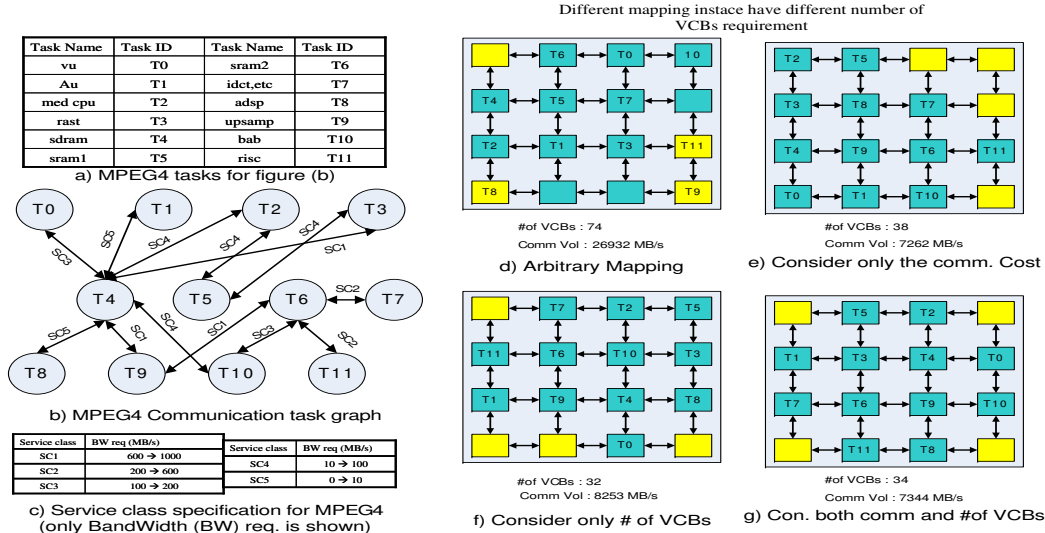


Figure 2: Motivating example MPEG4 video decoder mapped onto  $4 \times 4$  NoC architecture

associated NoCs (see Fig. 1(b,c)) is presented in [12, 19]. In this group the differentiation of transactions bandwidth requirement is done on the spatial domain. In a link of available bandwidth  $b$ , by allocating  $n$  number of *VCBs* (out of  $m$  *VCBs* in total) a total of  $(\frac{b}{m} \times n)$  bandwidth can be provided. The virtual channels are arbitrated in a round-robin fashion. It consumes a lot of buffer in terms of *VCB* implementation for fine granularity and suffers from starvation for some transactions. In [19] shown in Fig. 1(c) apart from [12] shown in Fig. 1(b), *VCBs* assigned for Best Effort (BE) traffic can be taken by Guaranteed Throughput (GT) at connection establishment time.

**To overcome all these problems**, we propose an architecture (Fig. 1(d)) that arbitrates on the temporal domain and has application-specific fine-granular service-class specification, thus higher resource utilization, and there is no particular static assignment of *VCBs*. For each transaction we need only one *VCB*, not multiple *VCBs* like [12, 19] and this motivates the reduction of *VCBs* for application-specific NoC design. In the scope of this paper, as a first step of the two-steps transaction specific *VCB* assignment strategy, the *VCB* assignment to each output port in the router is done during NoC mapping. There are several algorithms that can be used for mapping. In [9], a mapping using *Branch and Bound* algorithm has been proposed. *Genetic Algorithm* based mapping algorithms have been introduced in [13, 17]. All these previous works considered only the communication volume as the basis of their optimization. In [5] we are the first to address the importance of buffer minimization during NoC mapping and present a multi-objective optimization algorithm to optimize the amount of total communication volume and amount of *VCB* using a modified *ACO* algorithm. The performance of *ACO* algorithm and its suitability over other algorithms for optimization are given in [5, 18]. Within this paper, we adapt this *ACO* algorithm as the first step of our two-steps scheme to optimize the number of *VCBs*.

The rest of the paper is organized as follows. In Section 2, we summarize our novel contributions and show a motivational case study analysis using an MPEG4 video

decoder. In Section 3, our on-chip communication architecture using a service-class-based approach is outlined. In Section 4, we introduce our two-steps scheme to optimize the amount of buffers in a NoC. Experimental results are presented in Section 5 and finally we conclude in Section 6.

## 2 Our Novel Contribution and Motivation

Design space exploration to build an application specific QoS-supported NoC by parameterizing it, is an important research challenge. In the scope of this paper our novel contribution is as follows:

We provide a two-steps scheme to optimize the number of *VCBs* in a QoS-parameterized application specific on-chip communication architecture. The two steps are:

Step 1: We use a multi-objective mapping algorithm that considers both the communication volume and the number of *VCBs* during NoC mapping based on a modified *Ant Colony Optimization* (ACO) algorithm. This approach is orthogonal to any application driven traffic model.

Step 2: In the second step, we consider the traffic characteristics of the application. We observe that a wide range of digital media applications follow the *Poisson Distribution*. Therefore, we provide an analytical approach to optimize further the number of *VCBs* depending on QoS parameter  $q$  and the application specific traffic model. We present our methodology by means of a complete robot application case-study analysis. We also extend our exploration by evaluating the E3S embedded application benchmark suit.

Now let us motivate the need of an application specific virtual channel buffer assignment besides minimizing the total communication volume at design time by means of an MPEG4 video decoder case study analysis. This is the first step of our approach and it takes no assumptions on the traffic model and freeing/releasing buffer strategy. Fig. 2 (a,b,c) show the *Service Class associated Task Graph* (SCTG) of the MPEG4 video decoder. Fig. 2 (d) provides an arbitrary mapping of tasks in the  $4 \times 4$  mesh NoC. We find that the number of the *VCBs* needed is 74 and the total communication volume is 26,932 MB/s and therefore, higher area and energy consumption than the optimal value (with a given budget). The detailed explanation to calculate the

total number of *VCBs* and the total communication volume is given later in this paper in Section 4. The number of virtual channels is directly proportional to the area [5] and the leakage power [3]. The total communication volume is proportional to the communication related energy consumption [9, 13, 17]. Therefore, our goal for the application specific NoC design is to keep the amount of the *VCBs* and the total communication volume as low as possible and to consider them both at the mapping time.

Fig. 2 (e) shows the mapping of the tasks to the tiles considering only communication volume similar to the approaches [9, 13]. Here, the communication volume is 7,262 MB/s which is an optimal value and the number of the *VCBs* is 38. Now in Fig. 2 (f), if we optimize only for *VCBs*, then we have an optimal number of *VCBs* (32 *VCBs*) but the total communication volume increases to 8,252 MB/s. Therefore, it is clear that both these parameters need to be considered jointly during mapping. Using a very simple cost function presented later, we can achieve a trade off between the number of *VCBs* and the total communication volume. In Fig. 2 (g), we find 34 *VCBs* and the communication volume of 7,344 MB/s which are near optimal solution.

Let us now motivate the second step of our *VCB* reduction strategy. In Fig. 3 (a), different concurrent transactions between two tiles are shown. The traffic model and the parameters to fit into the *Poisson Distribution* is shown in Fig. 3 (b,c). The probability to use each *VC* is shown in Fig. 3 (d). It is assumed that the *VCBs* can be freed and be assigned at run-time, and thus depending on the probability, the designer can decide to put required amount of buffers in the corresponding output port. In this example, 2 *VCBs* can be used easily instead of 4 as *VCB3* and *VCB4* have lower probability (0.57% and 0.02% respectively) to be used during concurrent transmission. Detailed formulation and result analysis are given later in this paper. Therefore, if the traffic model is a priori known and the communication task graph is given then the *VCB* assignment can be carefully modelled at design time.

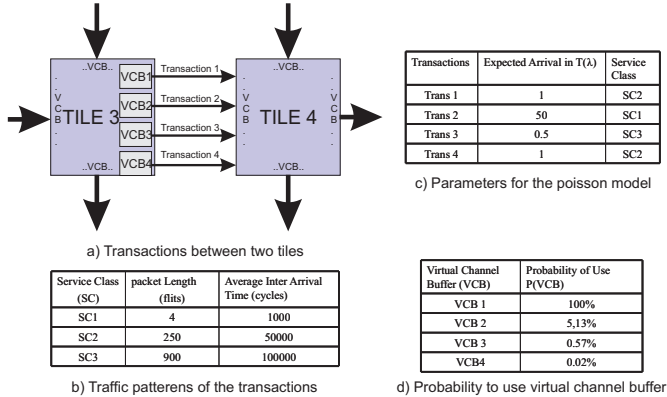


Figure 3: *VCBs* reduction using an analytical approach

### 3 Our On-chip Communication Scheme

Our on-chip communication scheme falls into the group of packet-based service-class-related NoC using wormhole routing. Virtual channels for the flow control mechanism are implemented using synchronous FIFOs with depth of  $(2x + y)$  described in [6] and width of the flit size for each element. Here  $x$  stands for the total amount of pipelined stages in the link (assuming 1 cycle for each stage) and  $y$  is the processing time in number of cycles in pair routers. The routing algorithm can be configured from any of the

deadlock free deterministic routing algorithms. In the current implementation, XY routing is used. The topology is kept as a grid like 2D mesh structure. The number of *VCBs* is constrained in each output port by the number of allowed concurrent connections, some of them allowing flexible transactions. The connections are freed after each unit transaction (after every packet transmission the *VCB* is freed to be allocated by other concurrent transaction). The size of each unit transaction depends on the application behavior and is parameterizable at design time for application specific NoC. All these *VCBs* are arbitrated on the time domain and provide buffer minimization considering all transactions of an application at design time and fine-grained bandwidth assignment per each transaction. The arbitration on the bandwidth for competing transactions is done on the temporal domain using a modified Time Division Multiple Access (TDMA) approach [6]. For the current buffer analysis, we assume that the complete application behavior including the traffic characteristics are known a priori.

## 4 Virtual Channel Buffer Reduction

In our architecture we use one *VCB* per router output port for each transaction along the complete path. A connection is built using a chain of *VCBs* along the route of the transaction. In the next subsection we explain how these number of *VCBs* in each output port can be optimized using a two-steps methodology.

### 4.1 Minimizing *VCBs* during Mapping

Generally in a NoC scenario, the application task graph mapping to a set of processing elements, the communication volume [9, 13, 17] is considered as the optimization criteria. All these above mentioned works are not optimizing for multiple criteria and do not consider the effect of increased number of *VCBs*. In Eq. (1) only the minimization of the communication volume is shown. In [5], a multi-objective optimization criteria considering both the total communication volume as well as the number of *VCBs* is presented. We adapt this multi-objective NoC mapping scheme as the first step to optimize the number of *VCBs* in an application-specific NoC. Eq. (3) considering both the optimization objective shown in Eq. (1) and Eq. (2) gives the cost function of our optimization algorithm for the NoC mapping. In the following, the specification of the two optimization criteria is summarized.

- **Total Communication Volume** of a given mapping configuration is:

$$vol_{tot}(TG_v, map) = \sum_{i=0}^{N_{ts}-1} \sum_{j=0}^{N_{ts}-1} d(i, j) v_{i,j} \quad (1)$$

$$\forall i, j \in \{0, \dots, N_{ts}-1\} : d(i, j) = |x_i - x_j| + |y_i - y_j|$$

$$\forall i \in \{0, \dots, N_{ts}-1\} : x_i = ts_i \bmod N_{noc}$$

$$\forall i \in \{0, \dots, N_{ts}-1\} : y_i = ts_i \div N_{noc}$$

where  $TG_v$  is the task graph containing the communication volume for every task communication  $v_{i,j}$ ,  $\forall i, j \in \{0, \dots, N_{ts}-1\}$ . The *map* data object contains the representation of task mapping. The elements of the mapping data object are values  $ts_i$ , where  $i$  stands for the "task id" and the value stored in  $ts_i$  represents the "tile id" where the task is being mapped

to. The total communication volume is calculated by accumulating the communication volume of each flow,  $v_{i,j}$  between two different tasks  $ts_i$  and  $ts_j$  in the task graph.  $v_{i,j}$  is multiplied by the Manhattan Distance between these two tasks  $d(i, j)$ .

- **Total Number of Virtual Channels** of a given mapping configuration is:

$$vcb_{tot}(TG_v, map) = \sum_{i=0}^{N_{ts}-1} \sum_{j=0}^{N_{ts}-1} d(i, j) \sigma(v_{i,j}) \quad (2)$$

$$\forall x \in int, x \geq 0 : \sigma(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \end{cases}$$

where the  $\sigma()$  function is applied only on positive integer values.

The two optimization criteria are combined to the total *cost function*<sup>1</sup> for the NoC mapping:

$$c = \alpha \frac{vcb_{tot}(TG_v, map)}{vcb_{tot,unmap}(TG_v)} + (1-\alpha) \frac{v_{tot}(TG_v, map)}{v_{tot,unmap}(TG_v)} \quad (3)$$

Here,  $vcb_{tot,unmap}(TG_v)$  and  $v_{tot,unmap}(TG_v)$  are almost the same as the two objective functions defined above in Eq. 1 and Eq. 2. Only the distance between the tasks is not considered when calculating them. Thus, the resulting values can be used *as a value* to normalize the partial target which then can be better compared with the other target, as it is done in the cost function  $c$ .

NoC mapping is a *Quadratic Assignment Problem (QAP)* as it is an *NP-hard* optimization problem [18]. In our approach, we use the optimization algorithm that can consider both the communication volume as well as the number of *VCBs* to solve this NP-hard problem. Our exploration unveils *Ant Colony Optimization Algorithm (ACO)* [5, 18] in finding the best solution for NoC. Our scheme is not limited to a specific optimization algorithm (algorithms supporting multi-objective optimization can be replaced).

## 4.2 Probabilistic Analysis

In order for further optimization the number of *VCBs* after the first step, we have carefully analyzed the application-driven traffic model. We have found a wide range of embedded system applications follow the *Poisson Distribution* for their corresponding traffic modeling.

### 4.2.1 Traffic Modeling

Multimedia applications, for instance, exhibit much more predictable traffic behavior than those found in traditional networks. The *Poisson Distributed Model* is a mathematical model commonly used in network analysis. It correctly models events which occur independently at random intervals. In a NoC, traffic patterns are generally more regular for a wide variety of applications. In wide range of digital media applications such as a motion recognition application of a robot called Image Processing Line (IPL) [15], for instance, packets are sent from one task to another in fairly regular intervals making it easy to determine well-defined

<sup>1</sup>Optimization criteria, cost function and goal function are used interchangeably

average intervals. In cases where a task does not send anything over one interval and does in another, we simply consider the sending interval. The *Poisson Distribution* is then calculated for that interval. Packet arrival in our model is assumed to be *Poisson Distributed*. Thus, the probability of  $X$  packets arriving in a given time interval  $T$  can be expressed by Eq. 4. The value of  $\lambda_{TR}$  is the number of arrivals of a particular transaction expected in the interval.

$$P_{\lambda_{TR}}(X = k) = \frac{e^{-\lambda_{TR}} \lambda_{TR}^k}{k!} \quad (4)$$

where:

$$\begin{aligned} \lambda_{TR} &= \frac{T}{\text{Average Arrival Time}} \\ k &= \# \text{of arrivals occurring in } T \end{aligned}$$

The *Poisson Distribution* only provides information on the number of packets expected in an interval. For some analysis, such as to predict *VCB* use, it is necessary to make a prediction on expected arrival times. Packet arrival behavior can be analyzed using the *Poisson Distribution's* respective *Poisson Process* shown in Eq. 5. Here  $P((N(t + \tau) - N(t)) = k)$  denotes the probability for  $N = k$  events to occur in the arbitrary time interval of length  $\tau$ . This allows us to make predictions on an event's expected time.

$$P_{\lambda_{TR}}((N(t + \tau) - N(t)) = k) = \frac{e^{-\lambda_{TR}\tau} (\lambda_{TR}\tau)^k}{k!} \quad (5)$$

### 4.2.2 VCB reduction due to QoS

We can now analyze the required number of virtual channel buffers to satisfy a QoS parameter  $q$  which defines the probability for successful transmission in terms of meeting the provided QoS bounds on throughput and latency in our architecture. This translates to the QoS-supported packet transmission. Here, the *VCB* reduction is done after the first step in the design space exploration to optimize the number of *VCBs*, where the number of *VCB* for concurrent connections and also designer defined flexible provision for transmission is already decided.

In the scope of this work, the QoS parameter  $q$  is defined to be a probabilistic value of the time-related QoS parameter latency and the throughput  $Q(l, t)$ . If a router architecture  $\mathfrak{R}(B, DL)$ , ( $B$  represents buffer elements and  $DL$  represents decoding logic) can provide bounded latency and throughput, means 100% of its required QoS value then  $q$  is assumed to be 100%. The value of  $q$  is parameterised by the number of *VCBs*, as if, there are rooms for every transaction considering the worst case: concurrent transactions. Every unit transaction will get a *VCB* in each output port if it is demanded after the first step of the transaction specific *VCB* assignment. The availability of the *VCB* is guaranteed because in the first step the buffer assignment is done depending on the total number of worst-case concurrent transactions. The value of  $q$  less than 100% influences the packet blocking time in the buffer assignment for concurrent transactions and thus can not meet the 100% guarantee.

Depending on all these properties of the router architecture and the application traffic model (*Poisson Distribution*), we now calculate the required number of *VCBs* using the following probabilistic analysis. We use the expected

### Algorithm 1 Transaction-specific VCB assignment

```

1: initialize(pher_tab)
   // initial values of the entries are  $\frac{1}{N_{tiles}}$ 
2: while !exit_condition do
   // ACO iterations can be limited by time or by
   // maximum number of iterations
3:   initialize(ant_pop)
   // generates a new ant population
4:   construct_solutions(ant_pop, TG_vol, pher_tab)
   // assign ants' tasks to tiles
5:   calculate_fitness(ant_pop)
   // cost_function:  $c = \alpha \frac{\#VCB}{\#VCB_n} + (1-\alpha) \frac{Comm_{vol}}{Comm_{vol,n}}$ 
6:   local_search(ant_pop)
7:   evaporate(pher_tab)
8:   drop_pheromone(pher_tab, ant_pop)
9:   if fitness(ant_best_pop) < fitness(ant_best_glob) then
10:    ant_best_glob  $\leftarrow$  ant_best_pop
11:   end if
12: end while
13: Use the probabilistic approach after the NoC mapping using ACO
14: for all tiles  $t_i$  do
   // All the tiles in the NoC
15:   if there are any transmission with the East tile then
   // connections in the East
16:     Use the probabilistic approach to opt. VCB depending on  $q$ 
17:   end if
18:   if there are any transmission with the West tile then
   // connections in the West
19:     Use the probabilistic approach to opt. VCB depending on  $q$ 
20:   end if
21:   if there are any transmission with the North tile then
   // connections in the North
22:     Use the probabilistic approach to opt. VCB depending on  $q$ 
23:   end if
24:   if there are any transmission with the South tile then
   // connections in the South
25:     Use the probabilistic approach to opt. VCB depending on  $q$ 
26:   end if
27: end for

```

values of the *Poisson Distributions*  $E(X) = \lambda$  to calculate the probability of  $TR_a$  claiming the first VCB:

$$P(VCB_1 = TR_a) = \frac{\lambda_{TR_a}}{\sum_i \lambda_{TR_i}} \quad (6)$$

The assignment of subsequent VCBs is then the conditional probability of the allocation of previous VCBs. That is, we take the probability for the first VCB and examine the *Poisson Processes* of each transaction for the time it is occupied. Since we are "filling up" the VCBs, we assume them to initially be empty. Thus  $\tau$  is only proportional to the  $TR$  in the first VC.

$$\tau = \frac{(p_a)(2m+n)}{T}$$

With a given unit time interval  $T$  and the packet size in flits of  $TR_a = p_a \times (2m+n)$  is the number of cycles that a flit remains inside the buffer. The probability of  $TR_b$  occupying the second VCB is then:

$$P(VCB_2 = TR_b | VCB_1 = TR_a) = (1 - e^{-\lambda_{TR_b} \tau}) \times P(VCB_1 = TR_a)$$

where:

$$\tau = \frac{\min(p_b, p_a)(2m+n)}{T}$$

Here,  $P(VCB_2 = TR_b)$  is calculated from Eq. 5. Now for the  $n$ th VC:

$$P(VCB_n = TR_a | VCB_{n-1} = TR_b) = (1 - e^{-\lambda_{TR_a} \tau}) \times P(VCB_{n-1} = TR_b | VCB_{n-2} = TR_c)$$

In order to meet the QoS requirements, the usage of a VCB must be guaranteed for a given percentage  $q$  contention-less packet transmission. To accomplish this, its usage may not exceed  $(1 - q)$ . That is, a virtual channel buffer is not required if:

$$\bigcup_{a,b} (P(VC_n = TR_a | VC_{n-1} = TR_b)) < (1 - q) \quad (7)$$

Therefore, to optimize the number of VCBs in the first step, we use a multi-objective mapping function during NoC mapping considering the worst case concurrent transactions and it provides 100% QoS. To further optimize the number of VCBs, we have analyzed the application specific traffic model. In the second step, depending on the QoS value, the designer can find an optimal solution to assign VCBs. The pseudo code of our VCB reduction scheme is presented in Algorithm 1. In this algorithm Line 1 to Line 12 present the pseudo code for the ACO-based mapping algorithm presented in [18]. A global pheromone table, *pher\_tab* is initialized by an uniform value considering the number of tiles,  $N_{tiles}$ . Ants in the algorithm can update the pheromone table and pheromone evaporation is also used to overcome the local minima in the optimization algorithm. Line 13 to Line 27 presents the probabilistic approach based on the traffic model and QoS parameter, which is applied after an optimum mapping is achieved using our mapping algorithm.

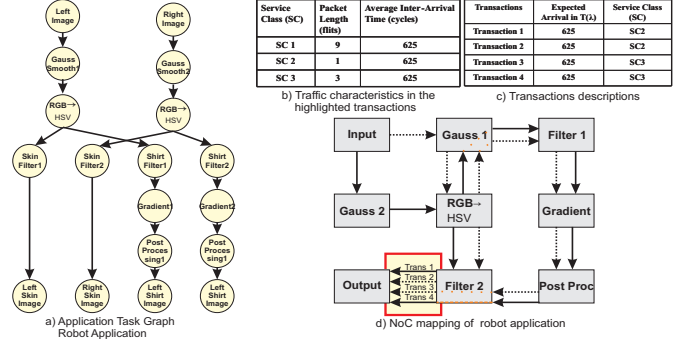


Figure 4: Application scenario for the robot application [15]

Transactions	$VC_1 = TR_1$	$VC_1 = TR_2$	$VC_1 = TR_3$	$VC_1 = TR_4$
$P(VC_2 = TR_1)$	0.0%	0.4%	1.17%	1.17%
$P(VC_2 = TR_2)$	0.4%	0.0%	1.17%	1.17%
$P(VC_2 = TR_3)$	0.4%	0.4%	0.0%	1.17%
$P(VC_2 = TR_4)$	0.4%	0.4%	1.17%	0.0%

Table 1: Likelihood of transaction  $i$  occupying  $VC_2$

Virtual Channel Buffer (VCB)	Prob. of use $P(VCB)$
$VCB_1$	100%
$VCB_2$	9.07%
$VCB_3$	0.44%
$VCB_4$	0.01%

Table 2: Probability to use each Virtual Channel Buffer

## 5 Results and Case Study Analysis

We have used a robot application [15] for the case study analysis but our scheme can be used for any other application. Fig. 4(a) gives the communication task graph and the application scenario for the image processing application used in a robot eye. The figure shows that the application has a pre-processing stage and a post-processing stage. The application is mapped onto a  $3 \times 3$  NoC architecture

shown in Fig. 4(d) to gain performance increase in terms of more frames decoding per second (from 5 fps in uniprocessor system to 25 fps in a MPSoC). For the experimental setup, some assumptions are considered: The image size is  $320 \times 200$  pixels, it decodes 25 frames per second what means 1 pixel per 625 ns, each flit payload size is 24 bits (1 RGB pixel), and robot left eye and the right eye process image parallelly. In a fixed *VCB* assignment approach the NoC needs 96 *VCBs*. After using the first step of our *VCB* reduction scheme the number of *VCBs* can be optimized to 20 for this presented optimized mapping. The second step of our scheme depends on the analysis of two neighboring tiles. For this application we have only shown the *VCB* reduction for the tiles *Filter 2* and *Output*. From *Filter 2* to *Output* there are 4 *VCBs*. The traffic characteristics for all these connections are also shown in the Fig. 4(b,c). Two transactions are of service class 2 and the other two are of service class 3. Depending on the probabilistic calculations, the number of *VCBs* can be optimized to 2 from 4 for this physical link as *VCB3* and *VCB4* have the probability of use of 0.44% and 0.01% respectively and these can be neglected in terms of provided QoS (see Table 2). Other connections can be also calculated using the same rules. Using the Eq. 6 the probability to have at least 1 *VCB* is found to be (100%). The expected probability to use a second *VCB* is 9.07% and the other values are shown in Table 1 using Eq. 7.

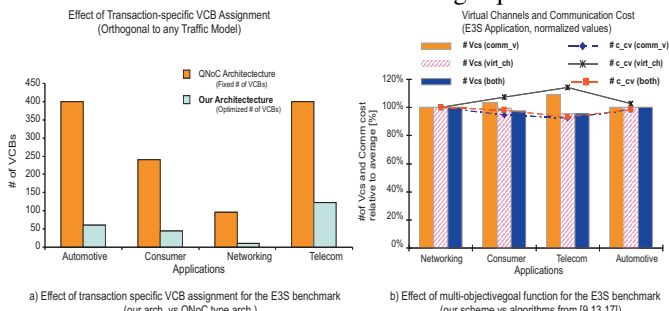


Figure 5: Comparing our architecture to the other state-of-the-art schemes

In Fig. 5(a), *VCB* savings, taking only the *VCBs* as the optimization criteria for the E3S benchmarks, compared to the fixed number of the *VCB* assignment approach is shown. The saving can be on average 90.2% for the applications. The number of *VCBs* in each output port for a fixed approach is taken 4 which is comparable to the QNoC [1]. In Fig. 5(b) the effect of multiple goal functions in NoC mapping for the E3S benchmark is shown. The bar chart shows the normalized number of *VCBs* and the lines show the normalized amount of total communication volume for three different optimization criteria. The results show different optimized mappings considering only the communication volume, only the number of *VCBs*, and both respectively.

## 6 Conclusion

A methodology for design space exploration using a two-steps schemes to optimize the number of virtual channel buffers in each output port of a router is presented in this paper. In our on-chip communication scheme, the routers are built to assign single *VCB* per router output port for each transaction and there is no *dedicated VCB* for a specific service class. This increases the chance to optimize the amount of buffer space, if the application-provided traffic characteristics is given. We achieve on an average 90.2% reduction in the number of virtual channels compared to state-of-the-art

QNoC, a fixed allocation for the E3S embedded application benchmark suit using the first step. In the second step, the reduction depends on the designer, the QoS parameter and the application-specific traffic model. We demonstrate our complete two-steps virtual channel buffer reduction scheme by means of a robot application case-study analysis.

## References

- [1] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny. "QNoC: QoS architecture and design process for network on chip". *Journal of Syst. Archit.* 50(2-3), pages 105–128, 2004.
- [2] S. Borkar. "Thousand core chips –A technology perspective". *DAC'07: Proc. of the 44th annual conf. on Design automation*, pages 746–749, 2007.
- [3] X. Chen and L.-S. Peh. "Leakage power modeling and optimization in interconnection networks". *ISLPED'03: Proc. of the 2003 int. symposium on Low power electronics and design*, pages 90–95, 2003.
- [4] W. J. Dally and B. Towles. "Route packets, not wires: on-chip interconnection networks". *DAC'01: Proc. of the 38th conf. on Design automation*, pages 684–689, 2001.
- [5] M. A. A. Faruque and J. Henkel. "Transaction specific virtual channel allocation in QoS supported on-chip communication". *ASAP'07: Proc. of the 18th int. conf. on Application-specific Systems, Architectures and Processors*, pages 76–81, 2007.
- [6] M. A. A. Faruque, G. Weiss, and J. Henkel. "Bounded arbitration algorithm for QoS-supported on-chip communication". *CODES+ISSS'06: Proc. of the 4th int. conf. on Hardware/software codesign and system synthesis*, pages 76–81, 2006.
- [7] J. Henkel, W. Wolf, and S. Chakradhar. "On-chip networks: A scalable, communication-centric embedded system design paradigm". *VLSID'04: Proc. of the 17th int. conf. on VLSI Design*, pages 845–851, 2004.
- [8] R. Ho, K. Mai, and M. Horowitz. "The future of wires". *Proc. of the IEEE*, 89(4): 490–504, April. 2001.
- [9] J. Hu and R. Marculescu. "Exploiting the routing flexibility for energy/performance aware mapping of regular NoC architectures". *DATE'03: Proc. of the conf. on Design, Automation and Test in Europe*, pages 10688–10693, 2003.
- [10] J. Hu and R. Marculescu. "Application-specific buffer space allocation for networks-on-chip router design". *ICCAD'04: Proc. of the 2004 IEEE/ACM int. conf. on Computer-aided design*, pages 354–361, 2004.
- [11] A. Jantsch and H. Tenhunen. "Networks on chip". *Kluwer Academic Publishers*, 2003.
- [12] N. Kavaldjiev, G. J. M. Smit, P. G. Jansen, and P. T. Wolkotte. "A Virtual Channel Network-on-Chip for GT and BE traffic". *ISVLSI '06: Proc. of the IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures*, pages 211–216, 2006.
- [13] T. Lei and S. Kumar. "A two-step genetic algorithm for mapping task graphs to a network on chip architecture". *DSD'03: Proc. of the Euromicro Symposium on Digital Systems Design*, pages 180–189, 2003.
- [14] U. Y. Ogras, J. Hu, and R. Marculescu. "Key research problems in NoC design: a holistic perspective". *CODES+ISSS'05: Proc. of the 3rd IEEE/ACM/IFIP int. conf. on Hardware/software codesign and system synthesis*, pages 69–74, 2005.
- [15] P. Azad et. al. "Image-based Markerless 3D Human Motion Capture using Multiple Cues". *Proc. of the int. workshop on vision based human-robot interaction*, 2006.
- [16] E. Rijpkema, K. G. W. Goossens, A. Radulescu, J. Dielissen, J. van Meerbergen, P. Wielage, and E. Waterlander. "Trade offs in the design of a router with both guaranteed and best-effort services for networks on chip". *DATE'03: Proc. of the conf. on Design, Automation and Test in Europe*, pages 10350–10355, 2003.
- [17] D. Shin and J. Kim. "Power-aware communication optimization for networks-on-chips with voltage scalable links". *CODES+ISSS'04: Proc. of the 2nd IEEE/ACM/IFIP int. conf. on Hardware/software codesign and system synthesis*, pages 170–175, 2004.
- [18] T. Stutzle and M. Dorigo. "ACO algorithms for the quadratic assignment problem". *New ideas in opt.*, pages 33–50, 1999.
- [19] P. Vellanki, N. Banerjee, and K. S. Chatha. "Quality-of-service and error control techniques for mesh-based network-on-chip architectures". *Integr. VLSI J.*, 38(3):353–382, 2005.