A Worst Case performance model for TDM Virtual Circuit in NoCs

Zhipeng Chen* and Axel Jantsch

Royal Institute of Technology(KTH), Sweden
{zhipeng, axel}@kth.se

Abstract. In Network-on-Chip (NoC), Time-Division-Mutiplexing(TDM) Virtual Circuit (VC) is well recognized as being capable to provide guaranteed services in both latency and bandwidth. We propose a method of modeling TDM based VC by using Network Calculus. We derive a tight upper bound of end-to-end delay and buffer requirement for indivdual VC. The performance analysis using Latency-Rate server is also presented in comparsion with our Performance model for TDM Virtual Circuit in NoCs (Pemvin). We conducted experiments on comparing Pemvin to the Latency-Rate server model. Our experiment results show the improvement of Pemvin on tightening the upper bound of end-to-end delay and buffer requirement.

1 Introduction

Development of modern submicron technology results in increasing of gate number and cores on one chip. Billion gates and over hundred cores on one die is now possible. Thus, current bus-based system-on-chip architecture is no longer adequate for multicores due to wire delay. Network-on-chip has emerged as novel paradigm which offers better on chip communication architecture. Network-on-chip has the potential to solve the scalabilty problem. However due to the contention of shared link and buffer, on chip network creates unpredictalbe performance. To overcome this nondeterminism, various approches are proposed to achieve Quality Of Service (QOS). Time-Division-Mutiplexing(TDM) Virtual Circuit (VC) is one approach among those which has been proposed in [1] [7].

TDM VC is a connection-oriented communitcation service in which two or more VC packet streams share buffers and link-bandwidth in turn. The time domain is divided into time slots, in every slot a fixed number of packets can be sent to the network. Each VC has its own dedicated slots to use shared resources. To dimension the worst case end-to-end delay and buffer requirement for TDM VC is an important probelm for three reasons. First, end-to-end delay is critical for implementing QOS, such as video stream or telephony service. Second, the buffer requirement is an important parameter while designing NoC implementations. Third, with the method that can dimension worst case end-to-end delay and buffer requirement, researchers are able to make high level evaluation of the NoC system without simulation.

In this paper we address the problem of performance analysis of individual TDM VC. Modeling of TDM VC makes it possible to have performance analysis of TDM VC in system synthesis. By knowing flow characteristics and slot allocation, researchers can obtain the end-to-end delay and buffer requirement directly without simulation. Currently this problem is not deep investigated. TDM VC is treated as Latency-Rate server in [6], which in some cases yields a tight bound. However, TDM VC has various slot distribution style within the time window, Latency-Rate server on the other hand uses avergage severvice rate to determin the behavior. Thus, a more accurate model is needed to give out precise performance analysis. We propose a formal approach by taking each slot and its corresponding interval within the time window as an individual session (see section 5.2).

The rest of the paper is structured as follows. Section 2 discusses the related work. Section 3 is an introduction of network calculus basic. We introduce the VC model of Latency-Rate server in Section 4. Section 5 introduces the formal model of TDM VC, Pemvin, for performance analysis. Section 6 shows the experimental results. Finally, Section 7 contains some conclusions and directions for future work.

2 Related work

End-to-end delay and buffer requirement dimensioning is a general problem of performance analysis of TDM VC after knowing the VC specification.

Network calculus [4] is a theoretical framework for analysing performance guarantees in computer networks. The foundation of network calculus lies in the mathematical theory of dioids, and in particular, the Min-Plus dioid. It offers us analytical mathmatical methods for buffer and delay dimensioning.

Latency-Rate server [6] is a general model for analysis of traffic scheduling algorithms. The behavior of a Latency-Rate scheduler is determined by two parameters the latency T and the allocated rate R. It can be used to derive bound of end-to-end delay and buffer requirements in a network of servers.

This paper considers modeling TDM VC into slots corresponding sessions. Each session can be modeled as Latency-Rate server as proposed in [6]. In [10], Lu and Jantsch proposed a simple model for evenly distributed time slots using network calculus. Pemvin analysis further on other occasions of VC slot allocation. Our results show that Pemvin has a significant improvement of the upper delay bound.

Our work focuses on end-to-end delay and buffer requirement dimensioning for TDM VC with given slot allocation. By comparing different schemes of VC slot allocation, we found that evenly distributed slots yield the best performance. However, with different number of VCs, it is not always possible to get evenly distributed slots. Thus, an approach to analysis performance of a given flow and slot allocation is necessary to get the allocation fit to the requirement of the flow.

3 Network calculus basic

3.1 Arrive curve and Tspec

In network calculus [4], the traffic sent by a source is bounded by an arrival curve. Given a wide-sense increasing function α defined for $t \geq 0$. We say that a flow F is constrained by α if and only if for all $s \leq t$: F(t) - F(s) $\leq \alpha$(t - s); F has α as an arrival curve.

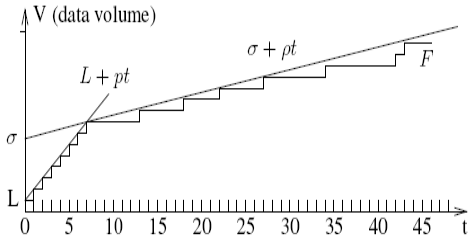

Fig. 1. TSPEC

A flow F(t) represents the accumulated number of bits transferred in the time interval [0, t]. We use TSPEC (Traffic SPECification) to represent flow characteristics. With TSPEC, F is characterized by an arrival curve $\alpha(t) = min(L + pt, \sigma + \rho t)$ in which L is the maximum transfer size, p the peak rate ($p \geq \rho$), σ the burstiness ($\sigma \geq L$), and ρ the average rate.

3.2 Service curve

Network calculus uses the concept of service curve to describe the minimum amount of service that is guaranteed to a flow. Consider a system S and a flow through S with input and output function F and F^*. We say that S offers to the flow a service curve β if and only if β is wide sense increasing, β (0) = 0 and $F^* \geq F \otimes \beta$, in which $\otimes$ is minplus convolution [4].

3.3 Analytical bound

With a flow's arrival curve and a network element's service curve, we can decide the delay and buffersize bound in the network element using network calculus. As illustrated in figure 2, the delay bound $\bar{D}$ is the maximum horizontal distance and the buffer size bound $\bar{B}$ the maximum vertical distance between the arrival curve and the service curve. In figure 2, $\bar{D}$ is either the distance marked as AA' or BB'; $\bar{B}$ is either the distance marked as AE or AE'. In network calculus, following equations are used to calculate delay bound and buffer size bound.

$$\bar{D} = inf\{D \geq 0 \; such \; that \; \alpha \oslash \beta(-d) \leq 0\} \tag{1}$$

$$\bar{B} = (\alpha \oslash \beta)(0) \tag{2}$$

where $\oslash$ is minplus deconvolution [4]. In network calculus, $\bar{D}$ and $\bar{B}$ are represented as maximum vertical distance and maximum horizontal distance of arrival curve and survice curve.

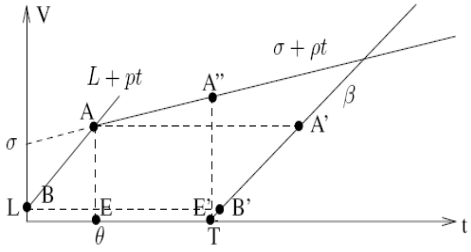

Fig. 2. Computation of delay and buffer size bound

4 VC modeling as simple Latency-Rate server

Function $\beta_{R,T} = R(t-T)^+$ describes the service model, where R is service rate and T is the maximum initial delay. Notation $x^+ = x$ if $x > 0$; $x^+ = 0$, otherwise.

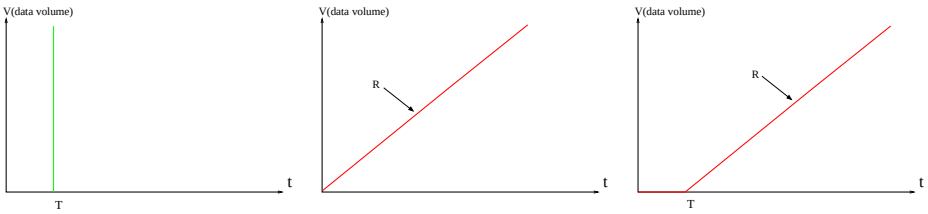

Fig. 3. latency rate server

Latency-Rate service curves as defined by function $\beta_{R,T} = R(t-T)^+$ is illustrated in figure 3. The service curve $\beta_{R,T} = \lambda_R \otimes \delta_T$, whereas $\otimes$ is defined as: $(\lambda_R \otimes \delta_T)(t) = inf\{\lambda_R(t-s) + \delta_T(s)\}$ [6]. For a given flow F(t) characterized by TSPEC arrival curve $\alpha(t) = min(L + pt, \alpha + \rho t)$, by knowing service rate R and initial Delay T, we have

$$\bar{D} = \frac{L + \theta(p-R)^+}{R} + T, \ where\, \theta = \frac{(\sigma - L)}{(p - \rho)} \tag{3}$$

$$\bar{B} = \sigma + \rho T + (\theta - T)^+[(p-R)^+ - p + \rho], \ where\, \theta = \frac{(\sigma - L)}{(p - \rho)} \tag{4}$$

TDM VC has an average service rate denoted by $R = n/T_w$, where n is the number of slots with in the time window, and T_w is the length of time window. There are many possiblities of TDM VC for the initial delay T. The worst case is that a packet misses a slot and have to wait entire interval between two slots.

We present a Latency-Rate server model here. In a TDM VC with n unevenly allocated slots, there are n intervals. Assume that the lengths of these n intervals are $T_1, T_2, ..., T_n$. The model uses the longest inteval length $T_i = max(T_1, T_2, ..., T_n)$ as initial delay. For TDM VC that has even slot allocation, this model is a close approximation. But for uneven slot allocation, the difference between sessional service rate and average rate could be significant. The worst case performance can be too pessimistic. Thus, a more accurate method of modelling is needed for buffer and delay dimentioning.

5 Pemvin TDM VC modeling

5.1 TDM VC modeling example

An example is illustrated in figure 4. VC v goes through the network. Assume that there are 16 time slots in each time window. Four slots are assigned to v every time window. There are totally C_{16}^4 possiblities of the admission patterns. For example,there are 4 slots in the VC and all have different length. From the beginning is 5,3,2,6 consecutively. Considering a TSPEC flow (6.4,0.1,1,1), if we use average serving rate 4/16 and average initial delay 4, by applying equation 3 and 4, we can get worst case end-to-end delay and buffer size. But in reality, VC serves packet exactly at the time slot reserved, for each session, the service rate is 1/5,1/3,1/2 and 1/6 respectively. Every session's rate is different from the average rate. Thus, the model is not accurate. With different permutation of the reserved slots, the worst case behavior will be different for a given VC. As we can see from the figure 4, if the initial slot length is 5, the worst case delay is 19, and worst case buffer size is 5. When the initial slot is 6, the worst case delay is 24, and worst case buffer size is 6. The results from Latency-Rate server model, the worst case delay is 28, and worst case buffer size is 7.

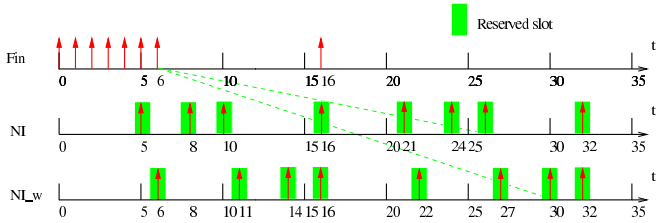

Fig. 4. Example of uneven VC

5.2 Pemvin TDM VC modeling

In a TDM VC, mutiple slots are allocated with a time window. Length of intervals between slots are different if allocation is uneven. While serving the flow, a finite length list of slots is repeated periodically. The intervals between slots may have same length(evenly distributed) or not(unevenly distributed). VC packets synchronously advance one hop per time slot. Pemvin models TDM VC with sessions.

Definition 1. *A session of a TDM VC is a reserved time slot and the inteval between this slot and next reserved time slot. For each session it has a session service rate R_i, which is given by function $R_i = 1/T_i$, where T_i is the length of interval.*

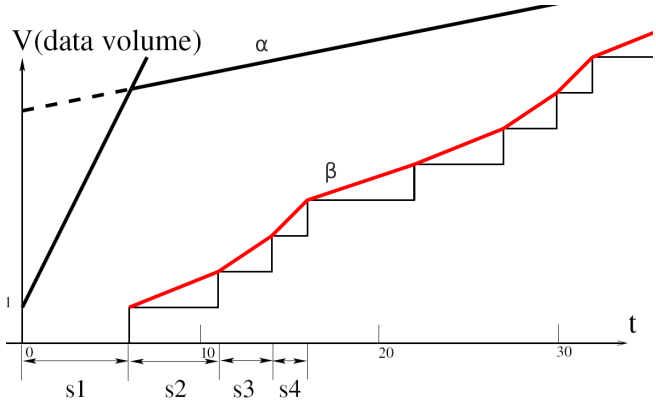

Fig. 5. Service curve of TDM VC (s_i stands for a session) α is the arrive curve and β is the service curve.

In Pemvin, each session considers to be an individual server. By concatenating the sessions together, we obtain:

$$\beta_{R_1,T_1} \wedge \beta_{R_2,T_2} \wedge \beta_{R_3,T_3} \wedge ... \wedge \beta_{R_I,T_I} = min_{1 \le i \le I}\{\beta_{R_i,T_i}\}$$
$$T_i = (T_j + \lfloor \frac{i}{n} \rfloor * (n - R_j T_w))$$
$$j = i - \lfloor \frac{i}{n} \rfloor * n, j \in [0, n-1]$$

in which β_{r_i,T_i} is session service curve. R_i is the service rate of the session service curve. T_i is the initial delay of the session service curve. j is the sequences number of the resevered slots. n is the number of the resevered slots of the time window. R_j is the average rate of j^{th} session. Together with flow characteristics and TDM VC slot allocation, we calculate R_i, T_i accordingly. Note that with different initial starting slots, there will be different service curves. With Pemvin, we obtain a new service curve, which taking into account of slot allocation of TDM VC. With network calculus, we use equation 1 and 2 to obtain upper bound of worst case end-to-end delay and buffer size.

5.3 Algorithm for Pemvin

To decide the starting slot of the TDM VC which yields worst case end-to-end delay and buffer size, Pemvin has following algorithm.

Algorithm 1 End-to-end Delay and buffer requirement dimensioning of TDM VC

Input: flow characteristics $F \sim (\sigma, \rho, L, p)$, VC slots $slots(s_1, s_2, ..., s_n)$, s_i is the time slot reserved.
output: Worst case end-to-end Delay D, Buffer size B.
Find if there is combinations of slots gives less service rate than ρ;
$\theta = (\sigma - L)/(p - \rho)$
If no combination found,check permutation for the first $\theta + 1$ packets;
If a combination exists, let this combination serves immediately after $\theta + 1$ packets.
Check if there is permuation gives lower service rate.
If so, change to new permutation and use the current permutation to get the worst case delay and buffer size.

6 Simulation and results

6.1 Simulation purpose and setup

In order to assess the proposed algorithm, we build up a simulator for TDM VC according to figure 6. The simulator was developed in C under Linux. A

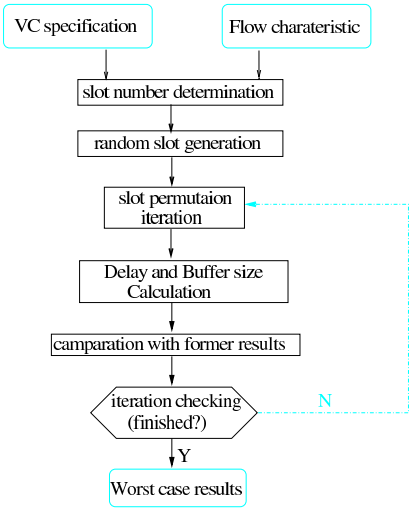

Fig. 6. Flow of Pemvin simulation

4 by 4 mesh network is constructed in the experiment. VCs are automatically generated. The bandwith of all links is 1 packet/cycle. We assume that all swithes have the same time window for TDM VC. The simulator will generate a TDM VC according to the characteristics of the TDM VC. Besides VC generator, the simulator also gives out the worst case delay and buffer size according to the flow characteristics and VC characteristics.

6.2 Exploration of different flow and TDM VC characteristics

To observe the accuracy difference between our model and latency-rate server model, we made experiments over different flow charcteristics and different VC slot allocations. In the table 1,We show experiment results of 6 flows. They are $F_1 \sim (1, 1, 16.16, 0.015)$,$F_2 \sim (1, 1, 14.11, 0.023)$,$F_3 \sim (1, 1, 12.06, 0.031)$, $F_4 \sim (1, 1, 6.16, 0.015)$, $F_5 \sim (1, 1, 4.11, 0.023)$, $F_6 \sim (1, 1, 2.06, 0.031)$ respectively. The first 3 flows have 20 packets injected into network for the first 256 cycles, and the last 3 flows have 10 packets injected into network within the 256 cycles. And for each flow has average rate ρ, VCs with average rate ρ, 2ρ are assigned.

Table 1. Results from Pemvin and Latency-Rate server model

σ	ρ	N	R	D_{lrs}	D_{Pemvin}	$diff_D$	B_{lrs}	B_{Pemvin}	$diff_B$
16.16	0.015	4	0.015	1125	1061	6.03%	17.53	17	3.12%
16.16	0.015	8	0.031	596	544	9.56%	17.46	17	2.71%
14.11	0.023	6	0.023	730	666	8.77%	17.07	17	0.41%
14.11	0.023	12	0.047	356	325	8.71%	15.53	15	3.41%
12.06	0.031	8	0.031	480	437	8.96%	15	15	0%
12.06	0.031	16	0.063	250	216	13.6%	14	14	0%
6.16	0.015	4	0.016	529	458	13.42%	8.18	8	2.2%
6.16	0.015	8	0.031	242	216	10.74%	6.88	6	14.67%
4.11	0.023	6	0.023	231	197	17.26%	5.4	5	8%
4.11	0.023	12	0.047	140	127	10.24%	5.35	5	7%
2.06	0.031	8	0.031	134	114	14.93%	4.42	4	10.5%
2.06	0.031	16	0.063	79	68	16.18%	3.52	3	17.33 %

All TDM VC slots from the table 1 are randomly generated. In the table, σ is the burstness of the flow, ρ is the average service rate of the flow, N is the number of the slots assigned to VC for this flow, D_{lrs} and B_{lrs} are the end-to-end delay and buffer requirement of the Latency-Rate server model, while D_{Pemvin} and B_{Pemvin} are the end-to-end delay and buffer requirement of Pemvin. $diff_D$ and $diff_B$ show the percentage differences of end-to-end delay and buffer requirement between Pemvin and Latency-Rate server model.

We simulated with evenly distributed TDM VCs and consecutively allocated TDM VCs too. We can see from the figure 7, for the same flow and same slots

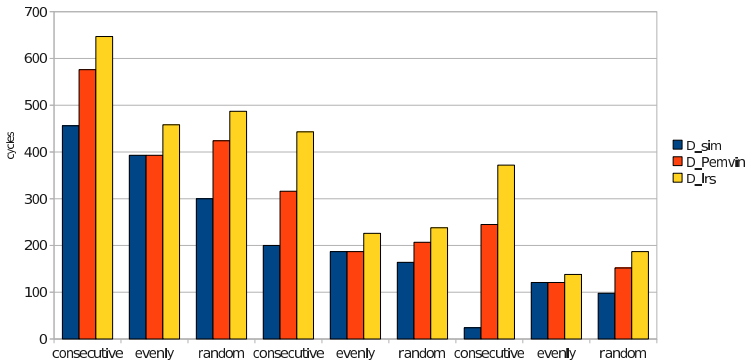

Fig. 7. Simulation results:end-to-end delay

number, the evenly distributed TDM VCs yeild best performance. On the contrary the consectively allocated TDM VCs yield worst performance. Each bar in the chart is the average values of 1000 simulations. The blue column is the value that starts from a random slot, the red column is the value from Pemvin and the yellow column is the value from Latency-rate server model. Labels under the columns mark the slots allocating style. We use flow $F \sim (1, 1, 10, 0.015)$ for simulation, 4 slots, 8 slots and 12 slots are assigned to the VC every 256 cycles.

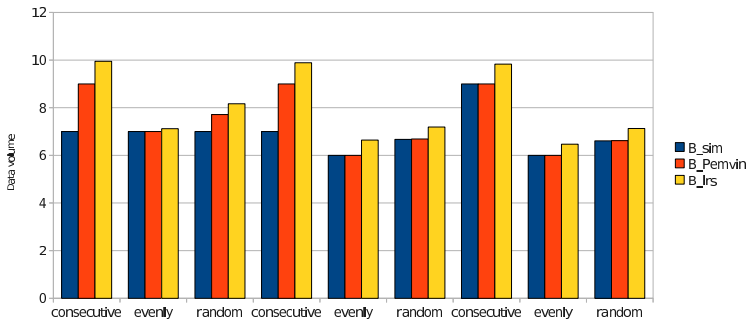

Fig. 8. Simulation results:buffer size

Also we can see from the chart in the difference in delay between Latency-Rate server model and Pemvin can go up to more than 30 percent. Thus, Pemvin significantly increase the accuracy of the delay bound. However the increase of accuracy on buffer size is not as good as on delay bound. Yet it is still tighter than Latency-Rate server model.

7 Conclusion

We proposed a TDM VC performce model Pemvin for end-to-end delay and buffer requirement dimensioning in this paper. A TDM VC is divided and modeled as sessions. By exploring different slot allocation schemes using Pemvin, we found that evenly distributed slots give out best worst case end-to-end delay and buffer requirement. In comparison with Latency-Rate server model, Pemvin yields significantly tighter bounds on worst case delay and tighter bounds on worst case buffer requirement.

Our future work will develop performance models for other VC schemes other than TDM VC. We will use results of Pemvin as constraints for TDM VC allocation. Our invetigestion will also extend to other switching schemes such as deflection flow control.

References

1. Goossens, K., Dielissen, J., Radulescu., A.: The Æthereal network on chip: Concepts, architectures, and implementations. In: IEEE Design and Test of Computers, vol. 22, no. 5, pp. 21–31 (2005)
2. Wang,Y., Zhou, K., Lu, Z., Yang, H.: Dynamic TDM Virtual Circuit Implementation for NoCs. In: Proceedings of Asia-Pacific Conference on Circuits and Systems (APCCAS'08), IEEE Conferences, China (2008)
3. Lu, Z., Jantsch, A.: Slot Allocation for TDM Virtual-Circuit Configuration for Network-on-Chip. In: Proceedings of the 2007 International Conference on Computer-Aided Design (ICCAD'07), IEEE Conferences, USA (2007)
4. Boudec, J.Y., Thiran, P.: Network Calculus A Theory of Deterministic Queuing Systems for the Internet. Online Version of the Book Springer Verlag - LNCS 2050 (2004)
5. Tong, L., Lu, Z., Zhang, H.: Exploration of Slot Allocation for On-Chip TDM Virtual Circuits. The 12th EUROMICRO Conference on Digital System Design (DSD'09), IEEE Conferences, Greece (2009)
6. Stiliadis, D., Varma, A.: Latency-Rate Servers: A General Model for Analysis of Traffic Scheduling Algorithms. Computer Engineering Department University of California Santa Cruz (1995)
7. Millberg, M., Nilsson, E., Thid, R., Jantsch, A.: Guaranteed bandwidth using looped containers in temporally disjoint networks within the Nostrum network on chip. In: Proceedings of the Design Automation and Test in Europe Conference, vol. 2, pp. 890–895 , IEEE Conferences, France (2004)
8. Coenen, M., Murali, S., Radulescu, A., Goossens, K., Micheli, G.: A buffer-sizing algorithm for networks on chip using TDMA and creditbased end-to-end flow control. In: Proceedings of International Conference on Hardware/Software Codesign and System Synthesis, Korea (2006)
9. Lu, Z., Jantsch, A.: TDM virtual-circuit configuration for Network-on-chip. In: Proceedings of the 2007 International Conference on Computer-Aided Design (ICCAD'07), IEEE Conferences, USA (2007)
10. Lu, Z., Brachos D., Jantsch, A.: A Flow Regulator for On-Chip Communication, The 22nd IEEE International SoC Conference (SoCC'09), IEEE Conferences, UK (2009)