Flow Regulation for On-Chip Communication

Zhonghai Lu, Mikael Millberg, Axel Jantsch
Royal Institute of Technology (KTH), Sweden

Alistair Bruce
ARM

Pieter van der Wolf, Tomas Henriksson
NXP Semiconductors Research

***Abstract*—We propose (σ, ρ)-based flow regulation as a design instrument for System-on-Chip (SoC) architects to control quality-of-service and achieve cost-effective communication, where σ bounds the traffic burstiness and ρ the traffic rate. This regulation changes the burstiness and timing of traffic flows, and can be used to decrease delay and reduce buffer requirements in the SoC infrastructure. In this paper, we define and analyze the regulation spectrum, which bounds the upper and lower limits of regulation. Experiments on a Network-on-Chip (NoC) with guaranteed service demonstrate the benefits of regulation. We conclude that flow regulation may exert significant positive impact on communication performance and buffer requirements.**

I. Introduction

IPs for a SoC are typically developed concurrently using a standard interface, for example, AXI [1] or OCP. Despite the standard interfaces, integrating IPs to a SoC infrastructure presents challenges because (1) traffic flows from IPs are diverse and typically they have stringent performance constraints; (2) the impact of interferences among traffic flows is hard to analyze; (3) due to the cost constraint, buffers in the SoC infrastructure must not be over-dimensioned while still satisfying performance requirements even under worst case conditions.

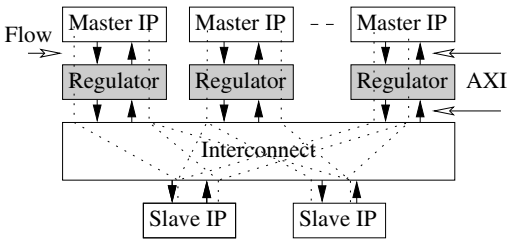

Fig. 1. IP Integration in SoCs

Figure 1 illustrates the approach that we propose and investigate in this paper for addressing the IP integration problem. Master IPs send requests to slave IPs which respond with read data and write acknowledgements. The admission of traffic flows from master IPs into the SoC infrastructure can be controlled by a regulator rather than injecting them as soon as possible. In this way, we aim to control Quality-of-Service (QoS) and achieve cost-effective communication. Informally, *flow regulation* means that traffic flows from IPs can be regulated according to a pre-configured agreement. By employing traffic regulation, we are able to trade one flow's performance with the others and reduce buffer requirements. To lay a solid foundation for our approach, our regulation is based on *network calculus* [2][3][4][5]. By importing the analytical methods from network calculus, we can obtain worst-case delay and backlog bounds. We also define *regulation spectrum*, which gives the theoretical effective range of regulation.

The aim of this paper is to show the benefits of flow regulation and quantify performance improvement and backlog reduction via analysis and simulation.

II. Related Work

Cruz [3] and Chang [5] have pioneered the *network calculus* [3], which is a mathematical framework to derive worst-case bounds on maximum latency, backlog and minimum throughput. In [4], a general latency-rate server model was proposed for analyzing traffic scheduling algorithms. Based on this model, they derived deterministic delay and backlog bounds. Le Boudec and Thiran [2] summarized the results of network calculus and their applications in Internet and ATM.

Real-time calculus [6], close to network calculus, was developed for platform-based embedded systems. It generalizes standard event models via upper and lower arrival curves, and processing-element models via upper and lower service curves. Based on these curves, it derives delay and backlog bounds.

III. Network Calculus Basics

A. Traffic Model: Arrival Curve and TSPEC

In network calculus [2], a flow $F(t)$ represents the accumulated number of bits transferred in the time interval $[0, t]$. To capture the average and peak characteristics of a flow, it uses TSPEC (Traffic SPECification). 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 (sustainable) rate. We denote this $F \sim (L, p, \sigma, \rho)$.

B. Service Curve and Analytical Bounds

Network calculus uses the abstraction of *service curve* to model a network element (node) processing traffic flows. A service curve reflects the processing latency and service capability of the node. A well-formulated service model is the latency-rate function $\beta_{R,T} = R(t-T)^+$, where R is the minimum service rate and T the maximum processing latency of the node [4]. Notation $x^+ = x$ if $x > 0$; $x^+ = 0$, otherwise.

As depicted in Figure 2a, a TSPEC flow $F \sim (L, p, \sigma, \rho)$ (denoted as $F : \alpha$) is served by a node guaranteeing a latency-rate service $\beta_{R,T}$. According to [2], the maximum delay for the flow is bounded by Eq. (1) and the buffer required for the flow is bounded by Eq. (2):

$$\bar{D} = \frac{L + \theta(p-R)^+}{R} + T \quad (1)$$

$$\bar{B} = \sigma + \rho T + (\theta - T)^+[(p-R)^+ - p + \rho] \quad (2)$$

978-3-9810801-5-5/DATE09 © 2009 EDAA

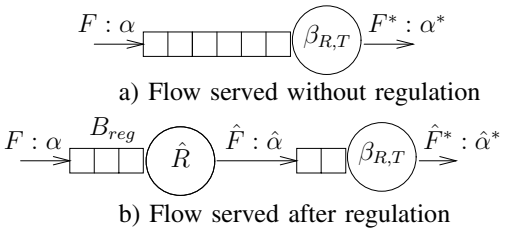

Fig. 2. Flow served by a latency-rate server

where $\theta = (\sigma - L)/(p - \rho)$. The output flow F^* is bounded by another affine arrival curve $\alpha^*(t) = (\sigma + \rho T) + \rho t$, $\theta \leq T$; $\alpha^*(t) = min\{(T+t)(min(p,R)) + L + \theta(p-R)^+, (\sigma+\rho T) + \rho t\}$, $\theta > T$.

IV. Flow Regulation and Periodic Flows

A. Regulation Spectrum

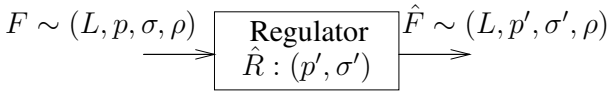

Fig. 3. Flow regulation.

TSPEC can be used to characterize flows. It can also be used to define a traffic regulator. Figure 3 shows that an input flow F is reshaped by a regulation component (regulator), resulting in an output flow $\hat{F}$. We assume the regulator has the same input and output data unit, *transfer*, and the same input and output capacity C transfers/cycle. We also assume that F's average bandwidth requirement must be preserved.

Given a flow F bounded by a TSPEC (L, p, σ, ρ) ($L \leq \sigma$, $\rho \leq p \leq C$), F can be losslessly reshaped by the regulator $\hat{R} : (p', \sigma')$ and the output flow $\hat{F}$ is characterized by the four parameters (L, p', σ', ρ), where $p' \in [\rho, p]$, $\sigma' \in [L, \sigma]$. "Lossless" means that $\hat{F}$ has the same L and average rate ρ as F. However, regulation may change F's peak rate and burstiness. The two intervals $p' \in [\rho, p]$ and $\sigma' \in [L, \sigma]$ are called the *regulation spectrum*, where the former is for the regulation of peak rate and the latter for the regulation of traffic burstiness.

Next, we use an example to show the effect of regulation and the regulation spectrum, which defines the limits of regulation.

B. Example

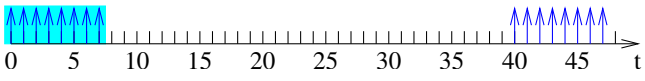

Fig. 4. Flow without regulation.

Let us consider a periodic data flow F that generates one transaction of 8 transfers every 40 cycles, as shown in Figure 4. Thus its average rate $\rho = 8/40 = 0.2$ transfer/cycle and burstiness $\sigma = 8 - 0.2 \cdot (8 - 1) = 6.6$ transfers. Its peak rate p is 1 transfer/cycle. Its TSPEC can therefore be written as $F \sim (1, 1, 6.6, 0.2)$. As can be seen, F has a maximal burstiness since all transfers in one period are launched one right after the other.

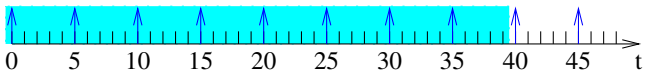

Fig. 5. Flow with strong regulation.

If the admission of flow F is controlled by regulator $\hat{R} : (0.2, 1)$, F is re-shaped as shown in Figure 5 and $\hat{F}$ has a TSPEC, $\hat{F} \sim (1, 0.2, 1, 0.2)$. This is the other extreme case where the burstiness ($\sigma = L = 1$) is minimal and its peak rate equals average rate ($p = \rho = 0.2$). As a result, the 8 transfers are evenly distributed over the period.

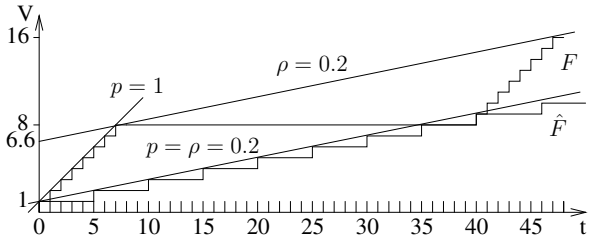

Fig. 6. Regulation spectrum.

Since the above two cases cover the two possible extremes of regulation, we can depict the flow's regulation spectrum in Figure 6. The region bounded by the two arrival curves $\alpha_{upper}(t) = min(1 + t,\ 6.6 + 0.2t)$ and $\alpha_{lower}(t) = 1 + 0.2t$ defines the regulation spectrum, where the former gives the upper bound and the latter the lower bound. Above this spectrum ($\sigma \geq 6.6,\ \rho = 0.2$), regulation has no re-shaping effect, i.e., $\hat{F} = F$. Below this spectrum ($\rho < 0.2$), the rate ρ of 0.2 cannot be sustained, thus it is not acceptable.

C. Revised Output Arrival Curve for Periodic Flows

The output arrival curve $\alpha^*(t)$ of a latency-rate server bounds the worst-case output for the input arrival curve $\alpha(t)$. However, the $\alpha(t)$-bounded worst-case input may not always occur. For example, for the periodic traffic in Section IV-B, the arrival curve $\alpha(t) = 6.6 + 0.2t$ allows injecting 14 transfers in 40 cycles. But, the periodic traffic which has an average rate of 0.2 transfer/cycle generates only 8 transfers per 40 cycles. Because of this, the output burstiness according to $\alpha^*(t)$ may be loose. We revise $\alpha^*(t)$ for the periodic traffic under consideration as follows.

Let σ^* be the burstiness of the output arrival curve. Let N be the number of bursty transfers to be delivered at rate $min(p, R)$ in each period. N can be calculated from the input arrival curve using the relation $N = \sigma + \rho\theta = L + p\theta$. Suppose the service time for the N transfers is θ^*, we have $N = \sigma^* + \rho\theta^*$ and $\theta^* = (\sigma^* - L)/(min(p, R) - \rho)$. Then we obtain $\sigma^* = (N(min(p, R) - \rho) + \rho L)/min(p, R)$. Therefore the output arrival curve for the periodic flow can be revised as

$$\alpha^*(t) = min\{L + (min(p, R))t,\ \sigma^* + \rho t\} \qquad (3)$$

D. Regulation Limits and Impact on IPs

Figure 2b shows that the flow is served after being regulated. From Eq. (1) and (2), we can see that reducing σ decreases both delay and backlog bounds, $\bar{D}$ and $\bar{B}$. The strongest regulation

occurs when $\sigma = L$, $p = \rho$. Then we obtain the minimal bounds on delay and backlog as $\bar{D}_{min} = \frac{L}{R} + T$ and $\bar{B}_{min} = L + \rho T$, respectively.

There are two ways that regulation may affect the behavior of IPs. One is that IPs are *stalled* and there is no queuing buffer at the regulator. The other is that IPs are *not stalled* but the regulators use buffers to store transactions. This can reduce back-pressure at the expense of buffering cost. Let D_{reg} and B_{reg} be the delay and backlog due to the regulation, respectively. In the first case, $B_{reg} = 0$ and $D_{reg} = 0$. In the second case as shown in Figure 2b, we have $B_{reg} = \Delta\sigma$, which is the difference between the input and output burstiness, and $D_{reg} = \Delta\sigma/\rho$. If taking into account of D_{reg} and B_{reg}, the regulation spectrum gives the range of delay bound variance, $\Delta\bar{D} = \bar{D} - (\bar{D}_{min} + D_{reg})$, and of backlog bound variance, $\Delta\bar{B} = \bar{B} - (\bar{B}_{min} + B_{reg})$.

V. Experiments

A. The Experimental Setup

The purpose of the experiments is to investigate the impact of flow regulation on communication performance and buffering cost. To show that our regulation is meaningful on emerging multicore SoC infrastructures, we performed experiments, both simulation and analysis, on the Nostrum NoC [7]. To help understand the regulation benefits, we designed simple but illustrative experiments.

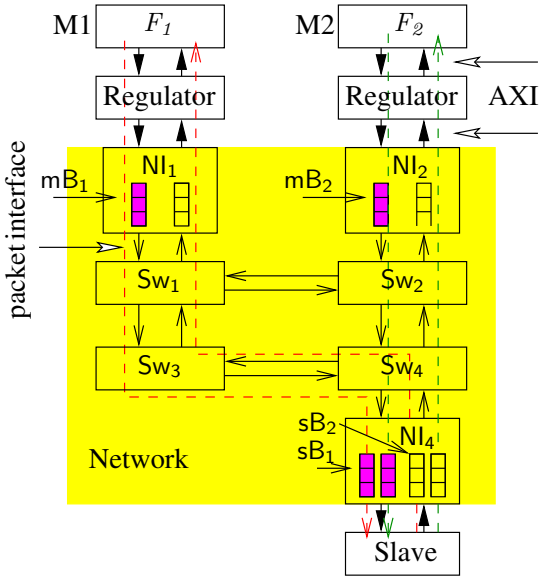

Fig. 7. The experiment architecture.

As depicted in Figure 7, the basic experimental architecture consists of 2 masters (M1 and M2), 1 slave, and a 2-by-2 mesh network. The network contains 4 switches (Sws) and 3 network interfaces (NIs). The interconnect interface is ARM's AXI [1]. Masters initiate read/write transactions over AXI. One transfer is encapsulated into one packet in its source-side NI, delivered in the network, and de-encapsulated into a transfer in its destination-side NI.

NIs provide an interface between IPs and the network. Note that the presence of NIs is the consequence of using a network not regulators. In NIs, there are buffer queues to buffer transfers before they enter the network. NI_4 multiplexes transactions from M1 and M2 to the slave. The slave is a simple memory model (with controller). There is no queuing and arbitration inside. The assumptions for this model are (1) it responds to a request in one cycle, as a typical SRAM does; (2) it can process multiple transactions concurrently. This means transfers of multiple transactions may be interleaved to the slave. Regulators are inserted between the masters and NIs. In the case of stalling IPs, the AXI AWREADY, WREADY and ARREADY signals are used by the regulators to signify whether the master is allowed to transfer or not on a clock cycle basis.

In the experiments, we have used two write flows, F_1 from M1 and F_2 from M2, as illustrated in Figure 7. Each flow allows one outstanding transaction. M1 and M2 generate one write transaction every 160 cycles. Each transaction contains 16 data transfers. Though the network supports both Virtual Circuit (VC) and Best Effort (BE) services, we have chosen to use the Virtual Circuit (VC) service for both flows. The reasons are (1) using the VC service can avoid stochastic behavior of the BE service; (2) as the VC service itself has stronger regulation effect on traffic than the BE service, we show that regulation makes sense even for strongly-regulated services. Each flow reserves one slot every 4 cycles, i.e., 1/4 transfer/cycle bandwidth. They are contention-free in the network. In our experiments, we set the slave processing capacity to one transfer every 4 cycles, i.e., 1/4 transfer/cycle. In this way, although both flows are free from contention in the network, they contend in the slave NI for the shared input bandwidth to the slave. The arbitration policy is round robin.

B. Analysis and Simulation: Theory Meets Practice

We built analytical models for the experiments in order to obtain and compare calculated results with simulated results. We concentrate on the uni-directional write data transfers W, since they are the most significant (write address (AW) and response (B) take only one transfer each).

Service Model for VC and Multiplexer: A VC can be modeled as a latency-rate server $\beta_{Bw,T}$, where Bw is the bandwidth of the VC and T the maximum waiting time for accessing the VC. In our case, a VC is modeled as $\beta_{0.25,3}$ since it provides $1/4 = 0.25$ packet/cycle bandwidth and the maximum waiting time for VC access is 3 cycles.

Using the VC service, write transfers reach the multiplexer in order, with an AW transfer leading a number of W transfers. The round-robin multiplexer interleaves transfers on its two input ports to the slave. The multiplexer offers 0.125 transfer/cycle (1 transfer every 8 cycles) to each flow and the maximum delay for each flow is 7 cycles. Note that we set the slave processing capacity to be 0.25 transfer/cycle. Therefore, the multiplexer guarantees a latency-rate service curve $\beta_{0.125,7}$ for each flow.

The Communication Analysis Model: With the service curves for VC and multiplexer, we built a network calculus analysis model for the experiments, as shown in Figure 8. To capture back-pressure due to regulation, we consider the case of not-stalling IPs (Section IV-D). Flow F_1/F_2 passes regulator

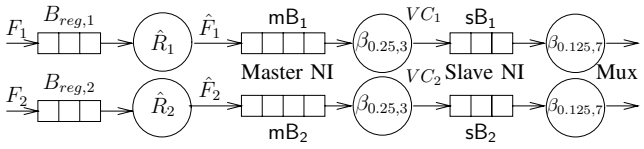

Fig. 8. The analysis model for experiments

$\hat{R}_1$/$\hat{R}_2$, and may be then buffered in NI_1/NI_2 before being delivered by VC_1/VC_2. Afterwards, F_1 and F_2 enter the slave NI (NI_4), which arbitrates and sends both flows to the slave.

The traffic model for F_1 and F_2 is TSPEC. Without regulation, $F_1 = F_2 \sim (1, 1, 14.5, 0.1)$. To examine the influence of regulating one flow on the other flow, we treat F_1 and F_2 differently. For F_2, we set regulator $\hat{R}_2$ as $\hat{R}_2 : (1, 14.5)$, resulting in no regulation effect on F_2, i.e., $\hat{F}_2 = F_2 \sim (1, 1, 14.5, 0.1)$. While un-regulating F_2, we apply a different regulation strength on F_1 by configuring regulator $\hat{R}_1$ for three cases : $\hat{R}_1 : (1, 14.5)$ (unregulated), $\hat{R}_1 : (1, 3)$, and $\hat{R}_1 : (0.1, 0.1)$ (strongest regulation). For the latter two cases, F_1 is reshaped as $\hat{F}_1 \sim (1, 1, 3, 0.1)$ and $\hat{F}_1 \sim (1, 0.1, 1, 0.1)$, respectively.

After obtaining service and arrival curves, we use Eq. (1) and (2) to compute the delay and backlog bounds, respectively, and use Eq. (3) to derive the output arrival curve of VCs.

$D_{transfer}$ (cycles)	**UnReg.** $(1, 1, 14.5, 0.1)$		**Reg.** F_1 $(1, 1, 3, 0.1)$		**Reg.** F_1 $(1, 0.1, 1, 0.1)$	
Flow	F_1	F_2	F_1	F_2	F_1	F_2
NC	128	126	38	126	23	126
SM	122	118	31	110	20	98
D_{reg} (NC)	0	0	115	0	135	0
D_{total} (NC)	128	126	153	126	158	126

TABLE I
ANALYZED VS. SIMULATED TRANSFER DELAY

Delay Results: Table I compares the calculated worst-case delay bounds and measured maximum delays for transfers, $D_{transfer}$. Regulation delays, D_{reg}, are also listed. The network calculus analysis and simulation results are annotated as "NC" and "SM", respectively. To calculate the worst-case delay bounds, we have adopted the concatenated node model to tighten the bounds using the so-called "pay bursts once" technique [2]. By this model, the two servers $\beta_{0.25,3}$ and $\beta_{0.125,7}$ are merged into one server $\beta_{0.125,10}$ ($\beta_{min(0.125,0.25),3+7}$). In addition, the transfer propagation delay through the regulator (1 cycle) and through the network (for F_1, 4 cycles; for F_2, 2 cycles) are added. As can be observed, the calculated bounds are fairly tight. As F_1 is regulated, the measured F_2's delays are also decreased. This is because holding back F_1's transfers facilitates F_2's traversal through the multiplexer. As a consequence, compared with the simulated results (from 118 to 110 to 98 cycles), the calculated delay bound (126 cycles) is getting looser since the simulations become farther away from capturing the worst case.

Backlog Results: We compare the analysis and simulation results for backlog in Table II. B_{reg} is the buffer before the regulator, if an input queue is used to buffer stalled transfers. Strengthening regulation results in more transfers stalled by the regulator before being allowed to be sent.

Backlog (flits)		**UnReg.** $(1, 1, 14.5, 0.1)$		**Reg.** F_1 $(1, 1, 3, 0.1)$		**Reg.** F_1 $(1, 0.1, 1, 0.1)$	
Flow		F_1	F_2	F_1	F_2	F_1	F_2
B_{mB}	NC	13	13	3.3	13	1.3	13
	SM	12	12	3	12	1	12
B_{sB}	NC	10.6	10.6	3	10.6	1.7	10.6
	SM	9	9	2	9	1	7
B_{reg} (NC)		0	0	11.5	0	13.5	0
B_{total}(NC)		23.6	23.6	17.8	23.6	16.5	23.6

TABLE II
ANALYZED AND SIMULATED BACKLOG

We can observe from Table II: (1) The simulated backlogs all fall into the calculated bounds, and all bounds are very tight. For example, for $\hat{F}_1 \sim (1, 1, 3, 0.1)$, the calculated bound for mB_1 in NI_1 is 3.3, and the observed maximum backlog is 3; (2) Due to regulation, F_1's buffer requirements for mB_1 in NI_1 and sB_1 in NI_4 are greatly decreased. Specifically, buffer bound for mB_1 is decreased from 13 to 1.3, bound for sB_1 from 10.6 to 1.7, and hence from 23.6 to 3 in total. Even if the buffers due to regulation without stalling the master IP (B_{reg} in the table) are taken into account, F_1's total buffer bound is decreased from 23.6 to 16.5.

VI. CONCLUSION

IP integration requires provision of performance guarantees for traffic flows and efficient buffer dimensioning techniques. In this paper, based on the (σ, ρ)-calculus, we have formally defined traffic regulation and regulation spectrum. We have also demonstrated, both theoretically and experimentally, that flow regulation may exert significant impact on communication performance and buffer requirements. Flow regulation slows down the traffic injection into the SoC infrastructure. This can potentially reduce contention for shared resources. Thus it can be used to control system performance. It also decreases traffic burstiness, reducing buffer requirements. On the other hand, regulation stalls traffic admission. It has to be used with care so that the IP performance constraints can be met.

ACKNOWLEDGMENTS

This work is supported in part by the SPRINT project funded under the EU IST IP6 programme contract IST-2004-027580.

REFERENCES

[1] ARM, "AMBA Advanced eXtensible Interface (AXI) protocol specification, Version 1.0," http://www.amba.com.

[2] J.-Y. L. Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. Number 2050 in LNCS, 2004.

[3] R. L. Cruz, "A calculus for network delay, part I: Network elements in isolation; part II: Network analysis," *IEEE Transactions on Information Theory*, vol. 37, no. 1, January 1991.

[4] D. Stiliadis and A. Varma, "Latency-rate servers: A general model for analysis of traffic scheduling algorithms," *IEEE/ACM Transactions on Networking*, vol. 6, no. 5, pp. 611–624, October 1998.

[5] C. Chang, *Performance Guarantees in Communication Networks*. Springer-Verlag, 2000.

[6] S. Chakraborty, S. Kunzli, and L. Thiele, "A general framework for analysing system properties in platform-based embedded system designs," in *Proc. of Design, Automation and Test in Europe Conference*, 2003, pp. 190– 195.

[7] Z. Lu and A. Jantsch, "TDM virtual-circuit configuration for network-on-chip," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 8, pp. 1021–1034, 2008.