

Manuscript Number: JSA-D-14-00078

Title: Delay Analysis and Buffer Sizing for Priority-Aware Networks-on-Chip

Article Type: Research Paper

Keywords: Networks-on-Chip (NoC); delay bound; buffer sizing

Corresponding Author: Mr. Baoliang Li,

Corresponding Author's Institution:

First Author: Baoliang Li

Order of Authors: Baoliang Li; Zeljko Zilic, Associate Professor; Wenhua Dou, Professor

**Abstract:** Priority-aware wormhole-switched Networks-on-Chip (NoC) is promising to meet both the worst-case and average-case performance requirements of on-chip communication. To deploy real-time applications on this platform, the worst-case delay and buffer requirement of each flow must be analyzed and guaranteed. In this paper, we first build a Real-Time Calculus (RTC) based performance model for the priority-aware wormhole-switched NoC. Based on this model, we then propose a delay analysis algorithm and a buffer sizing algorithm. The delay analysis algorithm can give tighter delay bound than the deterministic network calculus-based method, because our performance model takes the maximum service capability of routers and minimum arrival rate of each flow into consideration. The buffer sizing algorithm tries to reduce the buffer space required for each flow without violating the deadline constraint, which improves the result obtained by the link-level buffer-space analysis method. Both algorithms are topology-independent. Taking the architecture parameters and flow specifications as input, they gives the end-to-end delay bound and buffer requirement for each application. The tightness and correctness of our algorithms are verified by comparing with simulation results.

# Delay Analysis and Buffer Sizing for Priority-Aware Networks-on-Chip

Baoliang Li<sup>a,\*</sup>, Zeljko Zilic<sup>b</sup>, Wenhua Dou<sup>a</sup>

<sup>a</sup>*College of Computer, National University of Defense Technology, Changsha 410073, Hunan, China*  
<sup>b</sup>*Department of Electrical & Computer Engineering, McGill University, Montreal H3A-2A7, Quebec, Canada*

---

## Abstract

Priority-aware wormhole-switched Networks-on-Chip (NoC) is promising to meet both the worst-case and average-case performance requirements of on-chip communication. To deploy real-time applications on this platform, the worst-case delay and buffer requirement of each flow must be analyzed and guaranteed. In this paper, we first build a Real-Time Calculus (RTC) based performance model for the priority-aware wormhole-switched NoC. Based on this model, we then propose a delay analysis algorithm and a buffer sizing algorithm. The delay analysis algorithm can give tighter delay bound than the deterministic network calculus-based method, because our performance model takes the maximum service capability of routers and minimum arrival rate of each flow into consideration. The buffer sizing algorithm tries to reduce the buffer space required for each flow without violating the deadline constraint, which improves the result obtained by the link-level buffer-space analysis method. Both algorithms are topology-independent. Taking the architecture parameters and flow specifications as input, they gives the end-to-end delay bound and buffer requirement for each application. The tightness and correctness of our algorithms are verified by comparing with simulation results.

*Keywords:* Networks-on-Chip (NoC), delay bound, buffer sizing

---

## 1. Introduction

Networks-on-Chip (NoC) is proposed to meet the strict and complex communication requirements of modern large-scale Chip-MultiProcessor (CMP) and System-on-Chip (SoC). Although various NoC proposals have emerged, most of them are designed to improve the average-case performance of on-chip interconnects. However, there are a variety of on-chip applications, which are sensitive to the worst-case communication performance of NoC. Designing the on-chip real-time communication infrastructure for these applications and analyzing its feasibility is a major challenge for researchers.

---

\*Corresponding author. Tel: (+86)158-74875131  
Email address: libaoliang@nudt.edu.cn (Baoliang Li)

10 To meet the rigorous on-chip real-time communication requirement, some spe-  
cial hardware implementations, e.g. Time-Division Multiplexing-Access (TDMA) [1],  
circuit-switch [2] and time-triggered switch [3], have been proposed. However, the av-  
erage performance and resource utilization of these proposals are very poor when com-  
pared with the conventional wormhole-switched NoC. Thus, providing real-time com-  
munication support based on the wormhole-switched NoC becomes the most promising  
15 solution to meet both average-case and worst-case communication requirements.

20 A key step before the wormhole-switched NoC is adopted as the platform of real-  
time communication is to check whether the deadline constraints of all the real-time  
flows are met. In addition, an effective buffer analysis approach is also needed to op-  
timize the buffer allocation under real-time constraint, since the on-chip buffer usually  
contributes to a significant portion of the entire router's power and area [4, 5]. The  
tightness of worst-case delay and buffer requirement analysis is crucial for the appli-  
cation of wormhole switched NoCs in real-time communication, because an overoptimistic  
analysis will lead to the violation of deadline, while an overly pessimistic anal-  
ysis will make the utilization of on-chip resource very low.

25 The conventional simulation-based method is not appropriate for the worst-case  
analysis, because the worst-case scenarios are difficult to be captured by simulation.  
In contrast, analytical methods can directly establish the relationship between perfor-  
mance metrics and design parameters, and give the worst-case performance bound-  
30 s. Thus, most of the existing research focusing on the worst-case delay bound and  
buffer requirement of priority-aware wormhole-switched NoC are based on the analyt-  
ical method [6, 7, 8, 9, 10, 11, 12]. Among all these analytical methods, the Link-Level  
Analysis (LLA) method [7, 12] and the Deterministic Network Calculus (DNC)-based  
35 method [8] outperform the others when the tightness of delay bound and buffer require-  
ment are considered. The LLA method assumes that the buffer size is large enough to  
eliminate the influence of flow control on the delay bound. The DNC-based method  
[8] overcomes this limitation by utilizing the advanced operators and properties of the  
DNC theory. However, the delay bound obtained in [8] can be further improved if the  
maximum service capability of routers and the minimum arrival rate of input traffic are  
taken into consideration.

40 In this paper, we try to improve the delay bound and buffer requirement derived by  
the DNC [8] and LLA [7, 12] method. The theoretical background of our performance  
model is the Real-Time Calculus (RTC) [13], which is originally used for the schedul-  
ability analysis of real-time system. To the best of our knowledge, it is the first time  
45 that this theory is used to model and analyze the performance of NoCs. The reason we  
choose RTC lies in the fact that it is the extension of DNC theory, which considers the  
maximum service capability and minimum arrival rate while deriving the performance  
bounds. The main contribution of this paper is three-fold: (1) We construct a novel per-  
50 formance model for the priority-aware wormhole-switched NoC with credit-based flow  
control. (2) We propose an end-to-end delay analysis algorithm for the priority-aware  
wormhole-switched NoC based on our performance model. The delay bound obtained  
by our algorithm is much tighter than the DNC-based method [8]. (3) We propose  
a buffer sizing algorithm based on our performance model. This algorithm considers  
the impact of flow control on the delay bound, and allocates just enough buffer space  
at each router for the flows to meet their deadline. When applied to guide the buffer

55 allocation of priority-aware wormhole-switched NoCs, our algorithm can save more buffer space than the Link-Level Buffer-space Analysis (LLBA) method [12]. Our delay analysis and buffer sizing algorithms can be used for the design space exploration, IP core mapping, task mapping, routing selection, etc.

60 The rest of this paper is organized as follows: we present the existing performance analysis methods for the wormhole-switched NoC in Section 2. In Section 3, a brief introduction to the priority-aware wormhole-switched NoC and RTC theory are presented. The detailed modeling process is presented in Section 4, where we also propose the end-to-end delay analysis algorithm and buffer sizing algorithm. We present the comparison results with simulation and other analytical methods in Section 5. Finally, 65 we summarize our paper in Section 6.

## 2. Related Work

70 Network-on-Chip is designed to be either best-effort or guaranteed-service to meet different on-chip communication requirements. Best-effort NoC can make better use of the on-chip shared resource, but it does not necessarily provide any performance guarantee for the applications. To provide the guaranteed services for different applications, a simple and effective solution is to classify these applications into several service classes, each with different priorities, and the network provides services according to the priority of each class. Representative priority-aware implementations include QNoC [14], fixed-priority NoC [6] and *A*thereal [1] etc. The performance 75 evaluation methods for the priority-aware NoC include the average-case analysis and worst-case analysis. For the average-case analysis, simulation and analytical methods hold the dominant position. However, for the worst-case analysis, simulation is not sufficient due to the difficulty in covering all the corner cases.

80 The worst-case performance bound of the priority-aware wormhole-switched networks has been extensively studied in literature. In [9], the contention tree model was proposed to analyze the feasibility of real-time traffic delivered by priority-aware wormhole-switched NoC. It improves the previous results, e.g. lumped link model [10] and dependency graph model [11], by allowing the concurrent link usage. The Flow- 85 Level Analysis (FLA) proposed in [6] improves the results obtained by contention tree model [9], lumped link model [10] and dependency graph model [11]. A comprehensive comparison between FLA and the other method can be found in [15], in which several defects of the previous method are illustrated and the advantages of FLA are highlighted. The LLA [7] improves the FLA by analyzing each link segment separately. Two buffer sizing methods based on FLA and LLA, i.e. Flow-Level Buffer-space 90 Analysis (FLBA) and Link-Level Buffer-space Analysis (LLBA), are proposed in [12] to estimate the buffer size of priority-aware wormhole-switched NoC. Both FLA and LLA assume that the traffic arrives periodically or sporadically, and that routers have sufficiently large buffer size, which is a significant simplification to the realistic traffic pattern and router implementation. In addition, the FLBA and LLBA can only compute the minimum buffer size at each router to avoid the back-pressure caused by flow control. We can further reduce the buffer size until the deadline constraint is violated.

95 On the other hand, although the DNC based performance model for best-effort NoC proposed in [16] can also be applied to the analysis of priority-aware wormhole-

switched NoC, the obtained performance bounds are very conservative, especially for  
 100 the high-priority flows. This is because the model does not take the priority into consider-  
 eration. To overcome this limitation, a revised DNC performance model was proposed  
 105 to analyze the worst-case delay of priority-aware wormhole-switched NoC in [8]. We  
 found that the DNC method in [8] can be further improved if we take the maximum  
 110 service capability of each router and minimum arrival rate of each flow into consid-  
 eration. Motivated by this observation, we adopt the RTC theory [13] to build the  
 115 worst-case performance model for the priority-aware wormhole-switched NoC. Real-  
 time calculus extends the DNC theory [17] by integrating the minimum arrival curve  
 and maximum service curve to characterize more detailed information about the traffic  
 120 and service processes. Due to its ability of analyzing complex system, RTC has been  
 widely used in the modeling and analysis of Controller Area Networks [18], FlexRay  
 125 [19], etc. To ease the application of RTC, a toolbox has been implemented in [20] to  
 support the numerical calculation.

### 3. Preliminaries

#### 3.1. Basic Assumptions

115 In this paper, the priority-aware NoC is represented as a directional network topology graph  $G : V \times E$ , where  $V$  and  $E$  represent the set of routers and links respectively.  
 120 Each link  $e_{i,j} \in E$  corresponds to a physical channel connecting the two routers  $R_i$  and  
 $R_j$ . A flow is a sequence of packets with the same transmission path, source address  
 125 and destination address. Packet of different flows generated by a Intellectual Property  
(IP) core are buffered at different queues within the corresponding Network Interface  
 130 (NI). Each packet is comprised of one head flit, one tail flit and several body flits. The  
path of a flow  $f_i$  traversed is defined as a router chain starting from the injection router  
 135 (denoted as  $start_i$ ) and ending at the ejection router (denoted as  $end_i$ ). The set of all  
the flows in the network is denoted as  $\mathcal{F}$ , and each flow  $f_i \in \mathcal{F}$  has a fixed-priority  $P_i$   
 140 and deadline  $D_i$ . The set of routers along the path of  $f_i$  is denoted as  $\mathcal{R}_i$ , and the set  
of links a flow  $f_i$  traversed is denoted as  $\Gamma_i$ . There exists contention between flow  $f_i$   
 145 and  $f_j$ , if and only if  $\Gamma_i \cap \Gamma_j \neq \emptyset$ . For router  $R_j$  along the path of flow  $f_i$ , denote  
the set of contending flows at  $R_j$  sharing the same priority with  $f_i$  as  $\Theta_{R_j,f_i}$ , the set  
 150 of contending flows at  $R_j$  with lower priorities than  $f_i$  as  $\Omega_{R_j,f_i}$ , and the buffer size  
reserved at  $R_j$  for  $f_i$  as  $B_{R_j,f_i}$ .

The router we considered is the priority-aware wormhole-switched router proposed  
 155 in [6] and further discussed in [21][10][7]. Each router has the same number of input  
and output ports, and each input port have sufficient Virtual Channel (VC) to accom-  
modate all the incoming packets of different priority levels. The allocation of VC is  
 160 determined by the VC allocator. The buffer depth of each VC is finite, and the credit-  
based flow control [22] is adopted between adjacent routers to prevent buffer overflow.  
To ensure the predicable transmission delay, a deterministic routine computation mod-  
 165 ule is used to determine the output port of each packet. The crossbar is utilized to  
switch traffic from input ports to the output ports, and the switch operation is deter-  
mined by the switch allocator. The switch allocator is priority-aware. If multiple flits  
 170 from different input ports or different VCs of the same input port contend for the same

output port, it will only grant the flit with the highest priority. Flits from a lower priority can transmit a flit, if and only if there are no contending flits from higher priority or the higher-priority flits are self-blocked due to the insufficiency of VC buffer at the downstream router.

The micro-architecture of the priority-aware router considered in this paper has standard pipeline stages, i.e. Buffer-Write (BW), Route Computation (RC), VC Allocation (VA), Switch Allocation (SA), Switch Traversal (ST) and Link Traversal (LT). For the detailed description, please refer to [23]. Each head flit should go through all these stages to determine the path and reserve a VC for the following non-head flits. Non-head flits skip the RC and VA stages since the routine and VC have been determined by the head flit. Router resource and control information reserved for a packet will be released only after the tail flit of the packet has departed from the router. An additional priority field in the head flit is required for the routers to schedule multiple contending flows according to their priority. Although we focus on standard router, our method can be easily modified to support other router micro-architecture, e.g. single-cycle router [21][6][10][7] and speculation-based router [23]. We will demonstrate the adoption of our model in a single-cycle router in subsection 5.1. To simplify our analysis, we also assume that the entire chip is synchronous, with clock frequency  $f$  and period  $T$ . Our method can also be applied to analyze Global Asynchronous Local Synchronous (GALS) NoC with little modification, because the routers located in different voltage-frequency islands can be synchronized with a half cycle synchronizer [24], which can be abstracted as a fixed-latency element in DNC theory [17].

Our performance model is topology-independent, but to demonstrate the basic idea, we take the mesh topology shown in Fig. 1 as an example throughout this paper. Routers in the mesh topology have five input/output ports, corresponding to the four cardinal directions (West, East, North and South) and the Local IP core. Although there are only four flows, i.e.  $f_1, f_2, f_3$  and  $f_4$ , in the network, it is sufficient to demonstrate the idea of our method, and our method can handle more traffic flows efficiently. Our method extends the methods in [7][8] to allow multiple flows to share the same priority. Flits of different flows sharing the same priority are served in round-robin order when they compete the same output port. Since the minimum transmission unit in the priority-aware wormhole-switched NoC is flit and a high-priority flit can preempt the transmission of a low-priority flit, the NoC architecture considered in this paper is flit-level preemptive [25].

### 3.2. Introduction to Real-Time Calculus

Real-time calculus [13] is the extension of the DNC theory [17], by adding the upper service curve and lower arrival curve to describe the maximum service capability of a system and the minimum arrival rate of a event stream. Due to the space limitation, we only present the definitions of the RTC arrival curve and service curve in this subsection. For more details, please refer to [13].

**Definition 1 (Real-Time Arrival Curve [13]).** Denote by  $R[s, t)$  the number of events within the time interval  $[s, t)$ . The lower and upper bounds on  $R[s, t)$  are called the lower arrival curve  $\alpha^l$  and upper arrival curve  $\alpha^u$ , which satisfy

$$\alpha^l(t - s) \leq R[s, t) \leq \alpha^u(t - s), \forall s < t$$

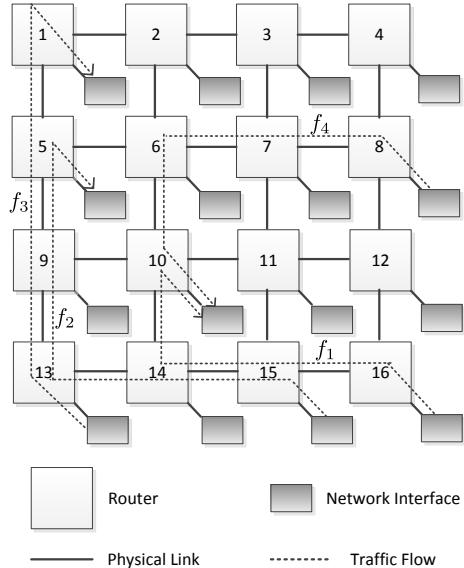


Figure 1: Mesh topology with four real-time traffic flows.

and  $\alpha^l(0) = \alpha^u(0) = 0$ . The RTC arrival curve for a event stream is denoted as  $\langle \alpha^l, \alpha^u \rangle$  for short.

**Definition 2 (Real-Time Service Curve [13]).** Denote by  $S[s, t]$  the number of events that can be processed by the system in time interval  $[s, t)$ . The lower and upper bounds on  $S[s, t]$  are called the lower service curve  $\beta^l$  and upper service curve  $\beta^u$ , which satisfy

$$\beta^l(t-s) \leq S[s,t] \leq \beta^u(t-s), \forall s < t$$

and  $\beta^l(0) = \beta^u(0) = 0$ . The RTC service curve for a system is denoted as  $\langle \beta^l, \beta^u \rangle$  for short.

From these two definitions, we find that the upper arrival curve and lower service curve correspond to the arrival curve and service curve of DNC theory [17]. Similarly, the upper service curve corresponds to the maximum service curve of DNC theory. Thus, the two concatenation theorems for service curve (see Theorem 1.46 in [17]) and maximum service curve (see Theorem 1.6.1 in [17]) together form the concatenation theorem for the RTC service curve. Assume an event stream traverses two systems  $S_1$  and  $S_2$  in sequence, and  $S_i$  offers an RTC service curve  $\langle \beta_i^l, \beta_i^u \rangle$  ( $i = 1, 2$ ) to this event stream. The concatenation theorem can give the equivalent RTC service curve offered by these two systems to the event stream, which is  $\langle \beta_1^l \otimes \beta_2^l, \beta_1^u \otimes \beta_2^u \rangle$ .

195 In this paper, we will utilize the discrete time RTC arrival curve and service curve to characterize the arrived traffic and service capability of the wormhole-switched NoC, since the minimum time unit of this system is the clock period  $T$ . Events in the definitions of arrival curve and service curve refer to the arrival and service of flits, respec-

tively. If we obtain the arrival curve  $\langle \alpha^l, \alpha^u \rangle$  of a specific flow at specific router  
200 and the service curve  $\langle \beta^l, \beta^u \rangle$  provided by this router, we can get the output arrival  
curve  $\langle \alpha'^l, \alpha'^u \rangle$  of this flow and leftover service curve  $\langle \beta'^l, \beta'^u \rangle$  of this router  
with the following equations [13]:

$$\alpha'^l = \min\{(\alpha^l \oslash \beta^u) \otimes \beta^l, \beta^l\} \quad (1)$$

$$\alpha'^u = \min\{(\alpha^u \otimes \beta^u) \oslash \beta^l, \beta^u\} \quad (2)$$

$$\beta'^l = (\beta^l - \alpha^u) \bar{\otimes} 0 \quad (3)$$

205

$$\beta'^u = \max\{(\beta^u - \alpha^l) \bar{\otimes} 0, 0\} \quad (4)$$

where  $\otimes, \oslash, \bar{\otimes}, \bar{\oslash}$  correspond to the min-plus convolution, min-plus de-convolution,  
max-plus convolution and max-plus de-convolution [17].

After obtaining the arrival curve  $\langle \alpha_f^l, \alpha_f^u \rangle$  of flow  $f$  and the equivalent service  
210 curve  $\langle \beta_f^l, \beta_f^u \rangle$  offered by the system to flow  $f$ , we can get the delay bound by the  
following equation [17]

$$Delay(f) = H(\alpha_f^u, \beta_f^l) \quad (5)$$

where operator  $H(\cdot, \cdot)$  computes the maximal horizontal deviation between its two  
operands.

#### 4. Delay Analysis and Buffer Sizing

In this section, we first build an RTC-based performance model for the priority-aware wormhole-switched NoC with credit-based flow control. Based on this performance model, we then propose an end-to-end delay analysis algorithm and a buffer sizing algorithm.  
215

The performance model comprises two parts, i.e. traffic model and service model. The traffic model utilizes the RTC arrival curve to describe the arrival process of each flow. We will introduce two methods to obtain the arrival curve in subsection 4.1. The service model characterizes the services offered by the priority-aware NoC to each flow. The construction of the service model is much more complicated than the traffic model, and the following three issues should be considered: (1) Only the head flit needs to traverse the RC and VA stage of a router, because the non-head flits of a packet follow the data-path built by the head flit. To simplify our service model, we need a special mechanism to characterize the service offered to head and non-head flits in a unified way. (2) Our performance model supports priority sharing among flows. Thus, the leftover service curve provided to the lower-priority flows can be derived only when all the service curves of high-priority flows have been computed. (3) The cyclic-dependence between the adjacent routers caused by flow control prevents us from deriving the end-to-end delay bound with Eq.(5), since the RTC theory is generally applicable to the feed-forward system. Thus, we should first break this cyclic-dependence before analyzing the performance bound. We will discuss the first two  
220  
225  
230

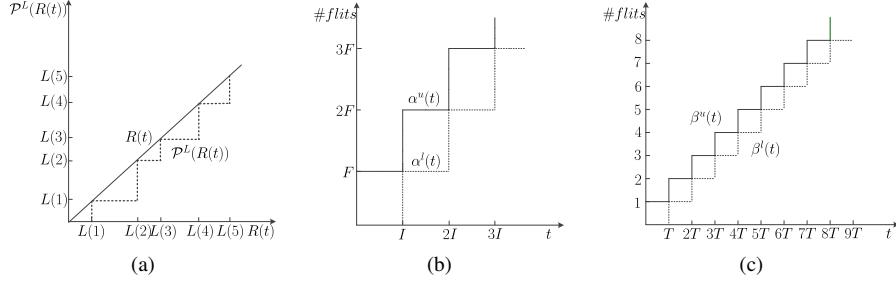


Figure 2: Traffic model and service model. (a) Definition of  $\mathcal{P}^L(R(t))$ . Cumulative arrival function  $R(t)$  and the  $L$ -packetized cumulative arrival function  $\mathcal{P}^L(R(t))$  are represented by the solid line and dotted line, respectively. (b) Real-time calculus arrival curve for periodically arrived traffic with period  $I$  and packet length  $F$ . The solid line and dotted line represent the upper arrival curve and lower arrival curve. (c) Service model for each pipeline stage. The solid lines and dotted lines represent the upper service curves and lower service curves, respectively.

issues in subsection 4.2, and the last issue is discussed in subsection 4.3. Finally, we  
235 present the delay analysis algorithm and buffer sizing algorithm in subsection 4.4 and  
subsection 4.5, respectively.

#### 4.1. Traffic Model

The communication in priority-aware wormhole-switched NoC is realized by transmitting packets, and each packet is divided into flits, which is the minimum transmission unit. Denote by  $\langle \alpha^l(\Delta), \alpha^u(\Delta) \rangle$  the flit arrival curve of a flow, namely, the minimum and maximum number of flits can be seen within any time window of length  $\Delta$ . We can extract the flit arrival curve from the synthetic traffic or communication trace with the sliding window method [13]. For each window length  $\Delta$ , this method tries to find the maximal and minimal number of arrived flits (corresponding to  $\alpha^l(\Delta)$  and  $\alpha^u(\Delta)$ ) by analyzing the time series of flits. However, the obtained flit arrival curve can only be applied to compute the worst-case performance bound at the flit level, because the service model we constructed in the following subsection characterizes the services provided by the system at the flit level. To obtain the packet level delay bound, this arrival curve must be  $L$ -packetized [17]. Denote by  $L(n)$  the cumulative packet length (in flits) of the first  $n$  packets in a flow,  $R(t)$  the cumulative arrived flits by time  $t$ . Then, the  $L$ -packetizer operator  $\mathcal{P}^l(\cdot)$  is defined as  $\mathcal{P}^L(R(t)) = \sup_{n \in \mathcal{N}} \{L(n)1_{L(n) \leq R(t)}\}$ <sup>1</sup>.  
240 Intuitively,  $\mathcal{P}^l(\cdot)$  can be interpreted as the largest cumulative packet length contained in  $R(t)$ , as shown in Fig. 2a. For any flit arrival curve  $\langle \alpha^l(\Delta), \alpha^u(\Delta) \rangle$ , the  $L$ -packetized arrival curve can be obtained by applying the following theorem.  
245

<sup>1</sup> $\mathcal{N}$  is the set of natural numbers and  $1_{\{val\}}$  is the indicator function,  $1_{\{val\}} = 1$  if and only if  $val$  is true.

255 **Theorem 1** (*L*-packetized arrival curve). Suppose a flow has a flit arrival curve  $\langle \alpha^l(\Delta), \alpha^u(\Delta) \rangle$ , and the maximum packet length (in flits) of this flow is  $l_{max}$ . Then, it has an *L*-packetized arrival curve  $\langle \alpha^l(\Delta) - l_{max}1_{\{\Delta>0\}}, \alpha^u(\Delta) + l_{max}1_{\{\Delta>0\}} \rangle$ .

PROOF. For  $\forall t \geq 0, \Delta \geq 0$ , according to the basic properties of *L*-packetizer [17], we have

$$R(t) - l_{max} < \mathcal{P}^L(R(t)) \leq R(t)$$

and

$$R(t + \Delta) - l_{max} < \mathcal{P}^L(R(t + \Delta)) \leq R(t + \Delta).$$

The inequalities above indicate

$$R(t + \Delta) - R(t) - l_{max} < \mathcal{P}^L(R(t + \Delta)) - \mathcal{P}^L(R(t))$$

and

$$\mathcal{P}^L(R(t + \Delta)) - \mathcal{P}^L(R(t)) < R(t + \Delta) - R(t) - l_{max}.$$

Based on the definition 1, the *L*-packetized flow has a packet arrival curve  $\langle \alpha^l(\Delta) - l_{max}1_{\{\Delta>0\}}, \alpha^u(\Delta) + l_{max}1_{\{\Delta>0\}} \rangle$ , which ends the proof. ■

260 For some special cases, we can also obtain the *L*-packetized arrival curve directly instead of transformation from flit arrival curve. For example, suppose all the packets in a flow have the same length  $F$  and arrived periodically with period  $I$ . By applying the sliding window method [13], we can obtain the flit arrival curve of this flow, which is a pair of staircase functions<sup>2</sup>  $\langle F \cdot u_{I,0}, F \cdot u_{I,I} \rangle$ . As shown in Fig. 2b, the obtained flit arrival curve which is equal to the *L*-packetized arrival curve  $\mathcal{P}^L(\alpha)$ , since  $\mathcal{P}^L(R(t)) = R(t)$ ,  $\mathcal{P}^L(\alpha^l(t)) = \alpha^l(t)$  and  $\mathcal{P}^L(\alpha^u(t)) = \alpha^u(t)$ .

265 270 275 280 285 290 295 300 305 310 315 320 325 330 335 340 345 350 355 360 365 370 375 380 385 390 395 400 405 410 415 420 425 430 435 440 445 450 455 460 465 470 475 480 485 490 495 500 505 510 515 520 525 530 535 540 545 550 555 560 565 570 575 580 585 590 595 600 605 610 615 620 625 630 635 640 645 650 655 660 665 670 675 680 685 690 695 700 705 710 715 720 725 730 735 740 745 750 755 760 765 770 775 780 785 790 795 800 805 810 815 820 825 830 835 840 845 850 855 860 865 870 875 880 885 890 895 900 905 910 915 920 925 930 935 940 945 950 955 960 965 970 975 980 985 990 995 1000 1005 1010 1015 1020 1025 1030 1035 1040 1045 1050 1055 1060 1065 1070 1075 1080 1085 1090 1095 1100 1105 1110 1115 1120 1125 1130 1135 1140 1145 1150 1155 1160 1165 1170 1175 1180 1185 1190 1195 1200 1205 1210 1215 1220 1225 1230 1235 1240 1245 1250 1255 1260 1265 1270 1275 1280 1285 1290 1295 1300 1305 1310 1315 1320 1325 1330 1335 1340 1345 1350 1355 1360 1365 1370 1375 1380 1385 1390 1395 1400 1405 1410 1415 1420 1425 1430 1435 1440 1445 1450 1455 1460 1465 1470 1475 1480 1485 1490 1495 1500 1505 1510 1515 1520 1525 1530 1535 1540 1545 1550 1555 1560 1565 1570 1575 1580 1585 1590 1595 1600 1605 1610 1615 1620 1625 1630 1635 1640 1645 1650 1655 1660 1665 1670 1675 1680 1685 1690 1695 1700 1705 1710 1715 1720 1725 1730 1735 1740 1745 1750 1755 1760 1765 1770 1775 1780 1785 1790 1795 1800 1805 1810 1815 1820 1825 1830 1835 1840 1845 1850 1855 1860 1865 1870 1875 1880 1885 1890 1895 1900 1905 1910 1915 1920 1925 1930 1935 1940 1945 1950 1955 1960 1965 1970 1975 1980 1985 1990 1995 2000 2005 2010 2015 2020 2025 2030 2035 2040 2045 2050 2055 2060 2065 2070 2075 2080 2085 2090 2095 2100 2105 2110 2115 2120 2125 2130 2135 2140 2145 2150 2155 2160 2165 2170 2175 2180 2185 2190 2195 2200 2205 2210 2215 2220 2225 2230 2235 2240 2245 2250 2255 2260 2265 2270 2275 2280 2285 2290 2295 2300 2305 2310 2315 2320 2325 2330 2335 2340 2345 2350 2355 2360 2365 2370 2375 2380 2385 2390 2395 2400 2405 2410 2415 2420 2425 2430 2435 2440 2445 2450 2455 2460 2465 2470 2475 2480 2485 2490 2495 2500 2505 2510 2515 2520 2525 2530 2535 2540 2545 2550 2555 2560 2565 2570 2575 2580 2585 2590 2595 2600 2605 2610 2615 2620 2625 2630 2635 2640 2645 2650 2655 2660 2665 2670 2675 2680 2685 2690 2695 2700 2705 2710 2715 2720 2725 2730 2735 2740 2745 2750 2755 2760 2765 2770 2775 2780 2785 2790 2795 2800 2805 2810 2815 2820 2825 2830 2835 2840 2845 2850 2855 2860 2865 2870 2875 2880 2885 2890 2895 2900 2905 2910 2915 2920 2925 2930 2935 2940 2945 2950 2955 2960 2965 2970 2975 2980 2985 2990 2995 3000 3005 3010 3015 3020 3025 3030 3035 3040 3045 3050 3055 3060 3065 3070 3075 3080 3085 3090 3095 3100 3105 3110 3115 3120 3125 3130 3135 3140 3145 3150 3155 3160 3165 3170 3175 3180 3185 3190 3195 3200 3205 3210 3215 3220 3225 3230 3235 3240 3245 3250 3255 3260 3265 3270 3275 3280 3285 3290 3295 3300 3305 3310 3315 3320 3325 3330 3335 3340 3345 3350 3355 3360 3365 3370 3375 3380 3385 3390 3395 3400 3405 3410 3415 3420 3425 3430 3435 3440 3445 3450 3455 3460 3465 3470 3475 3480 3485 3490 3495 3500 3505 3510 3515 3520 3525 3530 3535 3540 3545 3550 3555 3560 3565 3570 3575 3580 3585 3590 3595 3600 3605 3610 3615 3620 3625 3630 3635 3640 3645 3650 3655 3660 3665 3670 3675 3680 3685 3690 3695 3700 3705 3710 3715 3720 3725 3730 3735 3740 3745 3750 3755 3760 3765 3770 3775 3780 3785 3790 3795 3800 3805 3810 3815 3820 3825 3830 3835 3840 3845 3850 3855 3860 3865 3870 3875 3880 3885 3890 3895 3900 3905 3910 3915 3920 3925 3930 3935 3940 3945 3950 3955 3960 3965 3970 3975 3980 3985 3990 3995 4000 4005 4010 4015 4020 4025 4030 4035 4040 4045 4050 4055 4060 4065 4070 4075 4080 4085 4090 4095 4100 4105 4110 4115 4120 4125 4130 4135 4140 4145 4150 4155 4160 4165 4170 4175 4180 4185 4190 4195 4200 4205 4210 4215 4220 4225 4230 4235 4240 4245 4250 4255 4260 4265 4270 4275 4280 4285 4290 4295 4300 4305 4310 4315 4320 4325 4330 4335 4340 4345 4350 4355 4360 4365 4370 4375 4380 4385 4390 4395 4400 4405 4410 4415 4420 4425 4430 4435 4440 4445 4450 4455 4460 4465 4470 4475 4480 4485 4490 4495 4500 4505 4510 4515 4520 4525 4530 4535 4540 4545 4550 4555 4560 4565 4570 4575 4580 4585 4590 4595 4600 4605 4610 4615 4620 4625 4630 4635 4640 4645 4650 4655 4660 4665 4670 4675 4680 4685 4690 4695 4700 4705 4710 4715 4720 4725 4730 4735 4740 4745 4750 4755 4760 4765 4770 4775 4780 4785 4790 4795 4800 4805 4810 4815 4820 4825 4830 4835 4840 4845 4850 4855 4860 4865 4870 4875 4880 4885 4890 4895 4900 4905 4910 4915 4920 4925 4930 4935 4940 4945 4950 4955 4960 4965 4970 4975 4980 4985 4990 4995 5000 5005 5010 5015 5020 5025 5030 5035 5040 5045 5050 5055 5060 5065 5070 5075 5080 5085 5090 5095 5100 5105 5110 5115 5120 5125 5130 5135 5140 5145 5150 5155 5160 5165 5170 5175 5180 5185 5190 5195 5200 5205 5210 5215 5220 5225 5230 5235 5240 5245 5250 5255 5260 5265 5270 5275 5280 5285 5290 5295 5300 5305 5310 5315 5320 5325 5330 5335 5340 5345 5350 5355 5360 5365 5370 5375 5380 5385 5390 5395 5400 5405 5410 5415 5420 5425 5430 5435 5440 5445 5450 5455 5460 5465 5470 5475 5480 5485 5490 5495 5500 5505 5510 5515 5520 5525 5530 5535 5540 5545 5550 5555 5560 5565 5570 5575 5580 5585 5590 5595 5600 5605 5610 5615 5620 5625 5630 5635 5640 5645 5650 5655 5660 5665 5670 5675 5680 5685 5690 5695 5700 5705 5710 5715 5720 5725 5730 5735 5740 5745 5750 5755 5760 5765 5770 5775 5780 5785 5790 5795 5800 5805 5810 5815 5820 5825 5830 5835 5840 5845 5850 5855 5860 5865 5870 5875 5880 5885 5890 5895 5900 5905 5910 5915 5920 5925 5930 5935 5940 5945 5950 5955 5960 5965 5970 5975 5980 5985 5990 5995 6000 6005 6010 6015 6020 6025 6030 6035 6040 6045 6050 6055 6060 6065 6070 6075 6080 6085 6090 6095 6100 6105 6110 6115 6120 6125 6130 6135 6140 6145 6150 6155 6160 6165 6170 6175 6180 6185 6190 6195 6200 6205 6210 6215 6220 6225 6230 6235 6240 6245 6250 6255 6260 6265 6270 6275 6280 6285 6290 6295 6300 6305 6310 6315 6320 6325 6330 6335 6340 6345 6350 6355 6360 6365 6370 6375 6380 6385 6390 6395 6400 6405 6410 6415 6420 6425 6430 6435 6440 6445 6450 6455 6460 6465 6470 6475 6480 6485 6490 6495 6500 6505 6510 6515 6520 6525 6530 6535 6540 6545 6550 6555 6560 6565 6570 6575 6580 6585 6590 6595 6600 6605 6610 6615 6620 6625 6630 6635 6640 6645 6650 6655 6660 6665 6670 6675 6680 6685 6690 6695 6700 6705 6710 6715 6720 6725 6730 6735 6740 6745 6750 6755 6760 6765 6770 6775 6780 6785 6790 6795 6800 6805 6810 6815 6820 6825 6830 6835 6840 6845 6850 6855 6860 6865 6870 6875 6880 6885 6890 6895 6900 6905 6910 6915 6920 6925 6930 6935 6940 6945 6950 6955 6960 6965 6970 6975 6980 6985 6990 6995 7000 7005 7010 7015 7020 7025 7030 7035 7040 7045 7050 7055 7060 7065 7070 7075 7080 7085 7090 7095 7100 7105 7110 7115 7120 7125 7130 7135 7140 7145 7150 7155 7160 7165 7170 7175 7180 7185 7190 7195 7200 7205 7210 7215 7220 7225 7230 7235 7240 7245 7250 7255 7260 7265 7270 7275 7280 7285 7290 7295 7300 7305 7310 7315 7320 7325 7330 7335 7340 7345 7350 7355 7360 7365 7370 7375 7380 7385 7390 7395 7400 7405 7410 7415 7420 7425 7430 7435 7440 7445 7450 7455 7460 7465 7470 7475 7480 7485 7490 7495 7500 7505 7510 7515 7520 7525 7530 7535 7540 7545 7550 7555 7560 7565 7570 7575 7580 7585 7590 7595 7600 7605 7610 7615 7620 7625 7630 7635 7640 7645 7650 7655 7660 7665 7670 7675 7680 7685 7690 7695 7700 7705 7710 7715 7720 7725 7730 7735 7740 7745 7750 7755 7760 7765 7770 7775 7780 7785 7790 7795 7800 7805 7810 7815 7820 7825 7830 7835 7840 7845 7850 7855 7860 7865 7870 7875 7880 7885 7890 7895 7900 7905 7910 7915 7920 7925 7930 7935 7940 7945 7950 7955 7960 7965 7970 7975 7980 7985 7990 7995 8000 8005 8010 8015 8020 8025 8030 8035 8040 8045 8050 8055 8060 8065 8070 8075 8080 8085 8090 8095 8100 8105 8110 8115 8120 8125 8130 8135 8140 8145 8150 8155 8160 8165 8170 8175 8180 8185 8190 8195 8200 8205 8210 8215 8220 8225 8230 8235 8240 8245 8250 8255 8260 8265 8270 8275 8280 8285 8290 8295 8300 8305 8310 8315 8320 8325 8330 8335 8340 8345 8350 8355 8360 8365 8370 8375 8380 8385 8390 8395 8400 8405 8410 8415 8420 8425 8430 8435 8440 8445 8450 8455 8460 8465 8470 8475 8480 8485 8490 8495 8500 8505 8510 8515 8520 8525 8530 8535 8540 8545 8550 8555 8560 8565 8570 8575 8580 8585 8590 8595 8600 8605 8610 8615 8620 8625 8630 8635 8640 8645 8650 8655 8660 8665 8670 8675 8680 8685 8690 8695 8700 8705 8710 8715 8720 8725 8730 8735 8740 8745 8750 8755 8760 8765 8770 8775 8780 8785 8790 8795 8800 8805 8810 8815 8820 8825 8830 8835 8840 8845 8850 8855 8860 8865 8870 8875 8880 8885 8890 8895 8900 8905 8910 8915 8920 8925 8930 8935 8940 8945 8950 8955 8960 8965 8970 8975 8980 8985 8990 8995 9000 9005 9010 9015 9020 9025 9030 9035 9040 9045 9050 9055 9060 9065 9070 9075 9080 9085 9090 9095 9100 9105 9110 9115 9120 9125 9130 9135 9140 9145 9150 9155 9160 9165 9170 9175 9180 9185 9190 9195 9200 9205 9210 9215 9220 9225 9230 9235 9240 9245 9250 9255 9260 9265 9270 9275 9280 9285 9290 9295 9300 9305 9310 9315 9320 9325 9330 9335 9340 9345 9350 9355 9360 9365 9370 9375 9380 9385 9390 9395 9400 9405 9410 9415 9420 9425 9430 9435 9440 9445 9450 9455 9460 9465 9470 9475 9480 9485 9490 9495 9500 9505 9510 9515 9520 9525 9530 9535 9540 9545 9550 9555 9560 9565 9570 9575 9580 9585 9590 9595 9600 9605 9610 9615 9620 9625 9630 9635 9640 9645 9650 9655 9660 9665 9670 9675 9680 9685 9690 9695 9700 9705 9710 9715 9720 9725 9730 9735 9740 9745 9750 9755 9760 9765 9770 9775 9780 9785 9790 9795 9800 9805 9810 9815 9820 9825 9830 9835 9840 9845 9850 9855 9860 9865 9870 9875 9880 9885 9890 9895 9900 9905 9910 9915 9920 9925 9930 9935 9940 9945 9950 9955 9960 9965 9970 9975 9980 9985 9990 9995 10000 10005 10010 10015 10020 10025 10030 10035 10040 10045 10050 10055 10060 10065 10070 10075 10080 10085 10090 10095 10100 10105 10110 10115 10120 10125 10130 10135 10140 10145 10150 10155 10160 10165 10170 10175 10180 10185 10190 10195 10200 10205 10210 10215 10220 10225 10230 10235 10240 10245 10250 10255 10260 10265 10270 10275 10280 10285 10290 10295 10300 10305 10310 10315 10320 10325 10330 10335 10340 10345 10350 10355 10360 10365 10370 10375 10380 10385 10390 10395 10400 10405 10410 10415 10420 10425 10430 10435 10440 10445 10450 10455 10460 10465 10470 10475 10480 10485 10490 10495 10500 10505 10510 10515 10520 10525 10530 10535 10540 10545 10550 10555 10560 10565 10570 10575 10580 10585

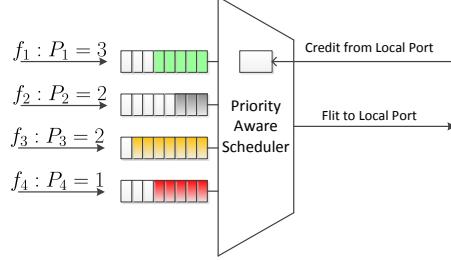


Figure 3: Structure of priority-aware network interface. Each flow has its dedicated buffer, and the scheduler schedules the flit with the highest-priority to go through the output link at each cycle.

Given the set of flow specifications, the service curve obtained by each flow at the source NI can be derived by applying Algorithm 1. The flow specification of  $f_i$  is a quadruple  $(\langle \alpha^l, \alpha^u \rangle, \mathcal{R}_i, D_i, P_i)$  that specifies the arrival curve, routine, deadline and priority of a flow.

---

**Algorithm 1** Compute the service curve at source NI

---

**Require:**  $(\langle \alpha^l, \alpha^u \rangle, \mathcal{R}_i, D_i, P_i) //$  the set of flow specifications  
**Ensure:**  $\langle \beta_{NI,f_j}^l, \beta_{NI,f_j}^u \rangle //$  the service curve obtained by each flow

- 1: Group the flows with priority  $P_i$  into subset  $\mathcal{F}_i$ .
- 2:  $\beta_{NI}^{l'} = \lfloor \frac{t}{T} \rfloor; \beta_{NI}^{u'} = \lceil \frac{t}{T} \rceil$ .
- 3: **for** each  $\mathcal{F}_i$  from highest priority to lowest priority **do**
- 4:   **for** each flow  $f_j \in \mathcal{F}_i$  **do**
- 5:      $\beta_{NI,f_j}^l = \lfloor \frac{\beta_{NI}^{l'}}{|\mathcal{F}_i|} \rfloor; // |\mathcal{F}_i|$  denotes the cardinality of  $\mathcal{F}_i$
- 6:      $\beta_{NI,f_j}^u = \lceil \frac{\beta_{NI}^{u'}}{|\mathcal{F}_i|} \rceil$ .
- 7:   **end for**
- 8:    $\alpha^l = \sum_{f_j \in \mathcal{F}_i} \alpha_{f_j}^l; \alpha^u = \sum_{f_j \in \mathcal{F}_i} \alpha_{f_j}^u;$
- 9:    $\beta_{NI}^{l'} = (\beta_{NI}^{l'} - \alpha^u) \bar{\otimes} 0; \beta_{NI}^{u'} = \max\{(\beta_{NI}^{u'} - \alpha^l) \bar{\otimes} 0, 0\}$ .
- 10: **end for**

---

285

#### 4.2.2. Service Curve Provided by Routers

While modeling the service capability of a router with RTC, we can analyze the data-path of a flow in the router stage-by-stage. Once we obtain the service curves offered by each stage, the service curve provided by the router to a flow can be derived by concatenating all these service curves. This is significantly different from the existing DNC based model [16, 8], where they treat the entire router as a whole and designate a Latency-Rate (LR) service curve [17] to simplify the performance bounds derivation. Whereas, our model uses the staircase functions to characterize the detailed behavior of this discrete time system. The advantage of our method is that, it can be easily modified to characterize the non-standard router micro-architectures, by simply letting the

290

295

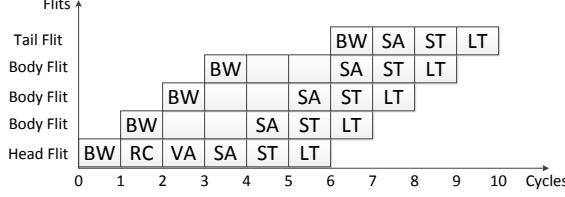


Figure 4: Time-line graph of a packet going through the standard router pipeline. The delayed tail flit enters the ST stage immediately after it is written into the dedicated buffer.

service curve of non-existed stages to be a burst delay function  $\delta_0(t)$ <sup>3</sup>. Next, we try to derive the service curves of all these stages:

(1) BW stage, SA stage and LT stage: all the flits within a traffic flow will go through these three stages, and experience a fixed delay  $T$  at each stage. The service curves provided by these stages, i.e.  $\langle \beta_{BW}^l, \beta_{BW}^u \rangle$ ,  $\langle \beta_{SA}^l, \beta_{SA}^u \rangle$  and  $\langle \beta_{LT}^l, \beta_{LT}^u \rangle$ , can be derived by applying the sliding window method [13], which are the same as the service curve of source NI, as shown in Fig. 2c.

(2) RC stage and VA stage: the latency of head flit experienced at these two stages is  $T$ . Although the non-head flits do not go through these two stages, they have to wait for at most two cycles before entering the SA stage, e.g. the first three body flits of a packet shown in Fig. 4. Thus, a sophisticated solution to construct a unified lower service curve for head flit and non-head flits at these two stages comes from viewing each of these two stages impose an additional delay  $T$  for all the flits. Thus, the equivalent lower service curve of these two stages, i.e.  $\beta_{RC}^l$  and  $\beta_{VA}^l$ , can be obtained by applying the sliding window method [13], which are equal to the staircase function  $u_{T,0}(t)$ . To derive the upper service curve of these two stages, let us consider the most ‘lucky’ flits of a flow, e.g. the tail flit shown in Fig. 4. This flit can enter the SA stage immediately after it was written into the dedicated VC buffer. For this case, the RC and VA stages impose a zero latency to it. Thus, we can utilize the burst delay function  $\delta_0(t)$  to represent the upper service curve of these two stages.

(3) ST stage: each output port of the wormhole-switched NoC has a switch allocator to schedule the switch traversal among all the contending flows at each clock cycle. The notation  $\langle \beta_{ST,R_i^p}^l, \beta_{ST,R_i^p}^u \rangle$  is used to identify the service curve obtained by all the contending flows injected into switch port  $p$ . For the mesh topology, let the port indicator  $p$  be  $W$  (West port),  $E$  (East port),  $S$  (South port),  $N$  (North port) or  $L$  (Local port). Thus, Following the same procedure as BW stage, we can get the service curves provided to all the contending flows, which is  $\langle u_{0,T}(t), u_{T,T}(t) \rangle$ , as shown in Fig. 2c.

It can be found that the contention of different flows within a router only might occur at ST stage. For the fixed-priority scheduling policy, switch allocators schedule

<sup>3</sup> $\delta_{val}(t) = +\infty$  if  $t > val$ , and 0 otherwise.

the flow with the highest priority first, flows with the same priority will be served with round-robin order. All the unscheduled flows will impose an additional latency  $T$  due to the failure of switch arbitration.

Denote by  $\langle \beta_{ST,R_i^p}^l, \beta_{ST,R_i^p}^u \rangle$  the total service curve provided by the ST stage,  
330     $\langle \beta_{ST,R_i,f_j}^l, \beta_{ST,R_i,f_j}^u \rangle$  the service curve provided to flow  $f_j$  by ST stage of router  
 $R_i$ , and  $\langle \beta_{ST,R_i^p}^{l'}, \beta_{ST,R_i^p}^{u'} \rangle$  the leftover service curve after serving the flows with  
higher priority than  $f_j$ . In order to obtain the service curve  $\langle \beta_{ST,R_i,f_j}^l, \beta_{ST,R_i,f_j}^u \rangle$ ,  
we should consider the following two cases:

(a) All the flows contending with  $f_j$  at  $R_i$  have lower priorities than  $f_j$ . For the synchronized router architecture, flow  $f_j$  gets the total service curve  $\langle \beta_{ST,R_i^p}^{l'}, \beta_{ST,R_i^p}^{u'} \rangle$ .  
335

(b) There exist some contention flows with the same priority as  $f_j$ . Since all the flows in  $\Theta_{R_i,f_j}$  got serviced in round-robin order, the service curve provided to  $f_j$  is  
 $\langle \lfloor \beta_{ST,R_i^p}^{l'}/(|\Theta_{R_i,f_j}| + 1) \rfloor, \lceil \beta_{ST,R_i^p}^{u'}/(|\Theta_{R_i,f_j}| + 1) \rceil \rangle$ . After serving all the flows  
in  $\Theta_{R_i,f_j}$ , the leftover service curve for low-priority flows can be derived by applying  
340 Eq.(3) and Eq.(4).

After we obtained the service curve provided by ST stage to flow  $f_j$ , we can get the service curve of the router directly. The equivalent feed-forward service curve of router  $R_i$  provided to  $f_j$ , i.e.  $\langle \beta_{R_i,f_j}^l, \beta_{R_i,f_j}^u \rangle$ , can be obtained by concatenating the service curves of all these stages together:

$$\begin{aligned}\beta_{R_i,f_j}^l &= \beta_{BW}^l \otimes \beta_{RC}^l \otimes \beta_{VA}^l \otimes \beta_{SA}^l \otimes \beta_{ST,R_i,f_j}^l, \\ \beta_{R_i,f_j}^u &= \beta_{BW}^u \otimes \beta_{RC}^u \otimes \beta_{VA}^u \otimes \beta_{SA}^u \otimes \beta_{ST,R_i,f_j}^u.\end{aligned}$$

#### 4.3. Feedback Service Model

To this end, we have constructed the traffic model and the basic feed-forward service model. However, these two models can only be applied to analyze the performance of wormhole-switched NoC without credit-based flow control. Flow control introduces  
345 cyclic-dependence between the adjacent routers, and blocks a flow when there is no buffer space available at the downstream router. The cyclic-dependence between the adjacent routers prevents us from deriving the performance bound directly even after we have obtained the service curve reserved at each router for the target flow. In existing literature, this problem is addressed by fixed-point iteration [26] or transformation  
350 from marked dataflow graph [27]. In this subsection, we will try to tackle the same problem with another solution.

Our solution is inspired by [16], where the authors abstract the flow control as a network element (called flow controller) providing a service curve (corresponding to the lower service curve of RTC theory), and this service curve can be obtained by applying  
355 some basic properties of DNC theory [17]. To make the discussion concrete, we take flow  $f_2$  in Fig. 1 as an example. We also utilize the scheduling network model [13] in RTC theory to visualize the credit-based flow control and complex relationship among  $f_2$  and the other flows, as shown in Fig. 5. We ignore flow  $f_4$  and the flow control of the other flows for brevity and clarity. We also assume that, all the destination IP cores can consume the arrived flits immediately, thus there is no flow control between the  
360

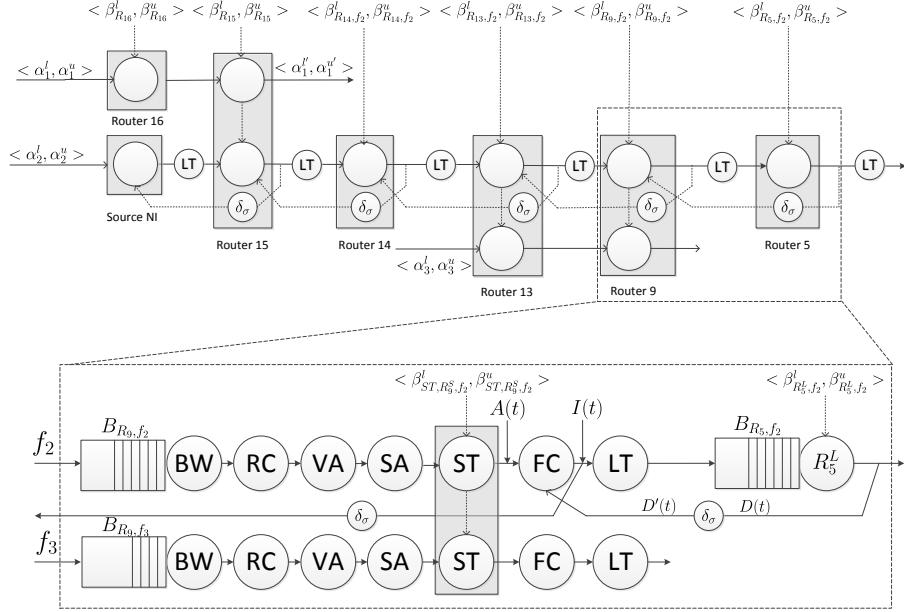


Figure 5: Scheduling network model for flow  $f_2$

ejection router and destination NI. However, to prevent the buffer overflow, the flow control between source NI and injection router is necessary.

A flow in the wormhole-switched router with credit-based flow control will be blocked if there is no credit available. We can treat the blocking of a flow caused by credit-insufficiency as traversing a virtual pipeline stage, called Flow Control (FC) stage, as shown in Fig. 5. The equivalent service curve for this virtual stage can be obtained by the following theorem, which enables us to break the cyclic-dependence caused by flow control and build a comprehensive performance model for the wormhole-switched router.

**Theorem 2.** Suppose that a router provides a feed-forward service curve  $\langle \beta^l, \beta^u \rangle$ , and the physical link provides service curve  $\langle \beta_{LT}^l, \beta_{LT}^u \rangle$ , the buffer size and credit feedback delay are denoted as  $B$  and  $\sigma$ , respectively. Then, the corresponding flow controller provides an service curve  $\langle \beta^l \otimes \beta_{LT}^l \otimes \delta_\sigma + B, \beta^u \otimes \beta_{LT}^u \otimes \delta_\sigma + B \rangle$ , where  $\bar{f}$  is the sub-additive closure<sup>4</sup> of  $f$ .

PROOF. We will take the flow controller between  $R_9$  and  $R_5$  in Fig. 5 as an example to derive the service curve of flow controller. Denote the amount of injected and departed flits at  $R_5$  by time  $t$  as  $I(t)$  and  $D(t)$ , and the amount of flits served by  $R_9$  by time  $t$  as  $A(t)$ . The feedback link can be represented as a network element providing upper

<sup>4</sup> $\bar{f} = \inf_{n \geq 0} \{f^{(n)}\}$ , where  $f^{(n)}$  represents the  $n$ th-fold convolution of  $f$  and  $f^{(0)}(t) = \delta_0(t)$ .

service curve  $\delta_\sigma(t)$ . The DNC service curve (i.e. lower service curve of RTC) has been derived in [16], which is  $\overline{\beta^l \otimes \beta_{LT}^l \otimes \delta_\sigma + B_{R_5, f_2}}$ .

In the rest of this proof, we will derive the upper service curve for the flow controller. For the flow control between router  $R_9$  and  $R_5$ , we have  $I(t) \leq A(t)$  for causality and  $I(t) \leq D'(t) + B_{R_5, f_2}$  due to the effect of flow control, where  $D'(t) \leq D \otimes \delta_\sigma(t)$ . Thus,

$$I(t) \leq \min\{A(t), D'(t) + B_{R_5, f_2}\}.$$

Since the upper service curve corresponds to the maximum service curve in DNC theory [17], the following inequality holds by the definition of maximum service curve (see definition 1.6.1 in [17])

$$D(t) \leq I \otimes \beta_{LT}^u(t) \otimes \beta_{R_5, f_2}^u.$$

Bring  $I(t)$  and  $D'(t)$  into the above inequality, we get

$$\begin{aligned} D(t) &\leq I \otimes \beta_{R_5, f_2}^u \otimes \beta_{LT}^u(t) \\ &\leq \min\{A \otimes \beta_{R_5, f_2}^u \otimes \beta_{LT}^u(t), D \otimes \delta_\sigma \otimes \beta_{R_5, f_2}^u \otimes \beta_{LT}^u(t) + B\}. \end{aligned}$$

By applying Theorem 4.31 in [17], we have

$$D \leq A \otimes \beta_{R_5, f_2}^u \otimes \beta_{LT}^u \otimes \overline{\beta_{R_5, f_2}^u \otimes \beta_{LT}^u \otimes \delta_\sigma + B_{R_5, f_2}}.$$

Thus,

$$\begin{aligned} I &\leq \min\{A, D' + B_{R_5, f_2}\} \\ &\leq \min\{A, D \otimes \delta_\sigma + B_{R_5, f_2}\} \\ &\leq \min\{A, A \otimes \beta_{R_5, f_2}^u \otimes \beta_{LT}^u \otimes \overline{\beta_{R_5, f_2}^u \otimes \beta_{LT}^u \otimes \delta_\sigma + B_{R_5, f_2}} \otimes \delta_\sigma + B_{R_5, f_2}\} \\ &= \min\{A \otimes \delta_\sigma, A \otimes \overline{\delta_\sigma \otimes \beta_{R_5, f_2}^u \otimes \beta_{LT}^u + B_{R_5, f_2}}\} \\ &= A \otimes \min\{\delta_\sigma, \overline{\beta_{R_5, f_2}^u \otimes \beta_{LT}^u \otimes \delta_\sigma + B_{R_5, f_2}}\} \\ &= A \otimes \overline{\beta_{R_5, f_2}^u \otimes \beta_{LT}^u \otimes \delta_\sigma + B_{R_5, f_2}} \end{aligned}$$

where the steps from the third line to the fifth line hold due to the general properties of  $\otimes$  operator (see Rule 6 and Rule 7 of Theorem 3.1.5 in [17]), and the last step follows from the definition of sub-additive closure.

The inequality  $I \leq A \otimes \overline{\beta_{R_5, f_2}^u \otimes \beta_{LT}^u \otimes \delta_\sigma + B_{R_5, f_2}}$  implies that the flow controller at  $R_9$  provides an upper service curve  $\overline{\beta_{R_5, f_2}^u \otimes \beta_{LT}^u \otimes \delta_\sigma + B_{R_5, f_2}}$ . Thus, we can conclude that for any router providing an upper service curve  $\beta^u$ , the corresponding flow controller offers an upper service curve  $\beta^u \otimes \beta_{LT}^u \otimes \delta_\sigma(t) + B$ , which ends the proof. ■

On obtaining the equivalent service curve of flow controller for  $f_i$  at router  $R_j$  (denoted as  $< \beta_{FC, R_j, f_i}^l, \beta_{FC, R_j, f_i}^u >$ ), we get the equivalent service curve of router  $R_j$  after breaking the cyclic-dependence loop:

$$\beta_{R_j, f_i}^l = \beta_{BW}^l \otimes \beta_{RC}^l \otimes \beta_{VA}^l \otimes \beta_{SA}^l \otimes \beta_{ST, R_j, f_i}^l \otimes \beta_{FC, R_j, f_i}^l,$$

$$\beta_{R_j, f_i}^u = \beta_{BW}^u \otimes \beta_{RC}^u \otimes \beta_{VA}^u \otimes \beta_{SA}^u \otimes \beta_{ST, R_j, f_i}^u \otimes \beta_{FC, R_j, f_i}^u.$$

Theorem 2 can be utilized to derive the RTC service curve of a single flow controller, and we can get the service curves of all the flow controllers along the router chain of any flow by applying Theorem 2 iteratively. As shown in Fig. 5, the service curve of a flow controller is determined by the service curves of the downstream flow controllers and routers. Hence, for each flow, we should compute the service curves of flow controllers from the ejection router to the source NI. Take flow  $f_2$  as an example, we have  $\beta_{FC, R_5, f_2}^l(t) = \delta_0(t)$  and  $\beta_{FC, R_5, f_2}^u(t) = \delta_0(t)$  since there is no flow control between  $R_5$  and destination NI. Then, we can compute the service curve of flow controller at  $R_9$ , which is  $\langle \beta_{R_5, f_2}^l \otimes \beta_{LT}^l \otimes \delta_\sigma + B_{R_5, f_2}, \beta_{R_5, f_2}^u \otimes \beta_{LT}^u \otimes \delta_\sigma + B_{R_5, f_2} \rangle$ . By applying the concatenation theorem, we can obtain the equivalent service curve provided to  $f_2$  by router  $R_9$ , which can be utilized to derive  $\langle \beta_{FC, R_{13}, f_2}^l, \beta_{FC, R_{13}, f_2}^u \rangle$  further. Following the same procedure, the service curve  $\langle \beta_{FC, R_{14}, f_2}^l, \beta_{FC, R_{14}, f_2}^u \rangle$ ,  $\langle \beta_{FC, R_{15}, f_2}^l, \beta_{FC, R_{15}, f_2}^u \rangle$  and  $\langle \beta_{FC, NI, f_2}^l, \beta_{FC, NI, f_2}^u \rangle$ <sup>5</sup> can be derived.

#### 4.4. End-to-End Delay Analysis

In this subsection, we present the delay analysis algorithm, as shown in Algorithm 2. This algorithm takes the architecture parameters and flow specifications as input, and gives the worst-case end-to-end delay for all the flows. The architecture parameters specify the network topology graph, buffer size of each VC and the service curve of each pipeline stage. The flow specifications describes the arrival curve, routine, deadline and priority of each flow. The arrival curve of flow  $f_i$  at the source NI and ST stage of router  $R_j$  are denoted as  $\langle \alpha_{f_i}^l, \alpha_{f_i}^u \rangle$  and  $\langle \alpha_{R_j, f_i}^l, \alpha_{R_j, f_i}^u \rangle$ , respectively. The leftover service curve of ST stage at output port  $p$  is represented as  $\langle \beta_{ST, R_j^p}^{l'}, \beta_{ST, R_j^p}^{u'} \rangle$  (Initially, let  $\beta_{ST, R_j^p}^{l'} = \beta_{ST, R_j^p}^l$  and  $\beta_{ST, R_j^p}^{u'} = \beta_{ST, R_j^p}^u$ ).

In the fixed-priority flit-level preemptive NoC, only the leftover service curve can be used by the low-priority flows. Thus, our algorithm compute the leftover service curve and delay bound from high-priority flows to low-priority flows. For each iteration, it performs the following four steps in sequence: (1) Calculate the service curves provided by the routers (lines 6-7) and flow controllers (lines 8-9) along the path. (2) Compute the worst-case end-to-end delay of the flow (lines 11-15), where the service curve provided by the source NI to  $f_i$ , i.e.  $\langle \beta_{NI, f_i}^l, \beta_{NI, f_i}^u \rangle$  has been computed with Algorithm 1. (3) The highlight of performance model when compared with the LLA method [7] and DNC method [8] is that our algorithm supports the priority-sharing. Thus, the leftover service curve at each router for low-priority flows can only be calculated when all the flows sharing the same priority have been calculated. To calculate the leftover service curve at ST stage, we have to first derive the equivalent service curve from source NI to  $R_j$  (lines 21-22) and the arrival curve at  $R_j$  (lines 24-25). Then, derive the leftover service curve with the aggregate arrival curve of the same priority-level (lines 26-31). The overall algorithm has two-level embedded loops, and

---

<sup>5</sup> $\langle \beta_{FC, NI, f_2}^l, \beta_{FC, NI, f_2}^u \rangle$  denotes the service curve of flow controller between source NI and injection router.

---

**Algorithm 2** End-to-end delay analysis algorithm

---

**Require:** Architecture parameters and flow specifications  
**Ensure:** Worst-case end-to-end delay for all the flows

- 1: **for** each flow  $f_i \in \mathcal{F}$  with priority order **do**
- 2:   // Compute the service curve at each router for  $f_i$
- 3:    $\beta_\tau^l = \delta_0(t);$
- 4:    $\beta_\tau^u = \delta_0(t);$
- 5:   **for** each router  $R_j \in \mathcal{R}_i$  from  $end_i$  to  $start_i$  **do**
- 6:      $\beta_{R_j, f_i}^l = \beta_{BW}^l \otimes \beta_{RC}^l \otimes \beta_{VA}^l \otimes \beta_{SA}^l \otimes \lfloor \frac{\beta_{ST, R_j^p}^{l'}}{|\Theta_{R_j, f_i}|+1} \rfloor \otimes \beta_\tau^l;$
- 7:      $\beta_{R_j, f_i}^u = \beta_{BW}^u \otimes \beta_{RC}^u \otimes \beta_{VA}^u \otimes \beta_{SA}^u \otimes \lceil \frac{\beta_{ST, R_j^p}^{u'}}{|\Theta_{R_j, f_i}|+1} \rceil \otimes \beta_\tau^u;$
- 8:      $\beta_\tau^l = \overline{\beta_{R_j, f_i}^l \otimes \beta_{LT}^l \otimes \delta_\sigma(t) + B_{R_j, f_i}};$
- 9:      $\beta_\tau^u = \overline{\beta_{R_j, f_i}^u \otimes \beta_{LT}^u \otimes \delta_\sigma(t) + B_{R_j, f_i}};$
- 10:   **end for**
- 11:    $\beta_{FC, NI, f_i}^l = \beta_\tau^l;$
- 12:    $\beta_{FC, NI, f_i}^u = \beta_\tau^u;$
- 13:   // Compute the end-to-end delay for  $f_i$
- 14:    $\beta_{f_i} = \beta_{NI, f_i}^l \otimes \beta_{FC, NI, f_i}^l \otimes (\bigotimes_{R_k \in \mathcal{R}_i} (\beta_{R_k, f_i}^l \otimes \beta_{LT}^l));$
- 15:    $Delay(f_i) = H(\alpha_{f_i}^u, \beta_{f_i});$
- 16:   // Compute the leftover service curve for low-priority flows
- 17:    $\beta_{f_i}^l = \beta_{NI, f_i}^l \otimes \beta_{FC, NI, f_i}^l;$
- 18:    $\beta_{f_i}^u = \beta_{NI, f_i}^u \otimes \beta_{FC, NI, f_i}^u;$
- 19:   **for**  $\forall R_j \in \mathcal{R}_i$  from  $start_i$  to  $end_i$  **do**
- 20:     **if**  $\Omega_{R_j, f_i} \neq \emptyset$  **then**
- 21:        $\beta^l = \beta_{f_i}^l \otimes \beta_{BW}^l \otimes \beta_{RC}^l \otimes \beta_{VA}^l \otimes \beta_{SA}^l;$
- 22:        $\beta^u = \beta_{f_i}^u \otimes \beta_{BW}^u \otimes \beta_{RC}^u \otimes \beta_{VA}^u \otimes \beta_{SA}^u;$
- 23:       //Compute the arrival curve of  $f_i$  at  $R_j$
- 24:        $\alpha_{R_j, f_i}^l = \min\{(\alpha_{f_i}^l \otimes \beta^u) \otimes \beta^l, \beta^l\};$
- 25:        $\alpha_{R_j, f_i}^u = \min\{(\alpha_{f_i}^u \otimes \beta^u) \otimes \beta^l, \beta^u\};$
- 26:       **if**  $Delay(f_k)$  has been calculated for  $\forall f_k \in \Theta_{R_j, f_i}$  **then**
- 27:          $\alpha_{R_j, f_i}^l = \alpha_{R_j, f_i}^l + \sum_{f_k \in \Theta_{R_j, f_i}} \alpha_{R_j, f_k}^l;$
- 28:          $\alpha_{R_j, f_i}^u = \alpha_{R_j, f_i}^u + \sum_{f_k \in \Theta_{R_j, f_i}} \alpha_{R_j, f_k}^u;$
- 29:          $\beta_{ST, R_j^p}^{l'} = (\beta_{ST, R_j^p}^{l'} - \alpha_{R_j, f_i}^u) \bar{\otimes} 0;$
- 30:          $\beta_{ST, R_j^p}^{u'} = \max\{(\beta_{ST, R_j^p}^{u'} - \alpha_{R_j, f_i}^l) \bar{\otimes} 0, 0\};$
- 31:       **end if**
- 32:       **end if**
- 33:        $\beta_{f_i}^l = \beta_{f_i}^l \otimes \beta_{R_j, f_i}^l \otimes \beta_{LT}^l;$
- 34:        $\beta_{f_i}^u = \beta_{f_i}^u \otimes \beta_{R_j, f_i}^u \otimes \beta_{LT}^u;$
- 35:     **end for**
- 36: **end for**

---

the computation complexity for this algorithm is  $O(HN)$ , where  $N$  and  $H$  is the number of flows and the hop count of each flow. This algorithm can be easily implemented in the RTC toolbox [20] to compute the end-to-end delay bound automatically. Since our algorithm takes the upper service curve and lower arrival curve into consideration, the delay bound obtained by our algorithm is much tighter than that of DNC-based delay analysis algorithm proposed in [8].

#### 435 4.5. Buffer Sizing

The priority-aware wormhole-switched NoC [6, 11, 21] requires the same amount of VCs as the priorities to prevent priority inversion, which refers to the blocking of high-priority flows when the low priority flows occupy all the VCs [10]. To reduce the buffer area and power consumption of priority-aware wormhole-switched NoC, priority sharing [28] and buffer optimization [12] techniques have been proposed. However, the backlog bound of each router derived in [12] is the minimum buffer size that does not trigger the flow control. Reducing this buffer size further will cause the back-pressure between adjacent routers and lead to a larger end-to-end delay. However, it is allowed to do so as long as the deadline constraint of each flow is not violated. Thus, our buffer sizing algorithm tries to reduce the initial buffer size of each router iteratively until the end-to-end delay of a flow violates its deadline or the buffer size is equal to one. The initial buffer size to avoid flow control can be obtained by applying the following theorem.

**Theorem 3.** Denote by  $\beta_{R_j, f_i}^l$  the feed-forward lower service curve obtained at  $R_j \in \mathcal{R}_i$  by flow  $f_i$ ,  $\beta_{LT}^l$  the lower service curve provided by the physical link between two adjacent routers,  $B_{R_j, f_i}$  the VC buffer size reserved for  $f_i$  at  $R_j$  and  $\sigma$  the credit feedback delay. Then, for  $\forall R_j \in \mathcal{R}_i$ , the buffer size

$$B_{R_j, f_i} = \lceil \inf\{B | \beta_{R_j, f_i}^l \otimes \beta_{LT}^l \otimes \overline{\beta_{R_j, f_i}^l \otimes \beta_{LT}^l \otimes \delta_\sigma(t) + B} \geq \beta_{R_j, f_i}^l \otimes \beta_{LT}^l\} \rceil$$

is large enough to avoid flow control.

PROOF. We take the flow  $f_2$  in Fig. 5 as an example to verify this theorem. As stated by Theorem 2, we have

$$\beta_{FC, R_9, f_2}^l = \overline{\beta_{LT}^l \otimes \beta_{R_5, f_2}^l \otimes \delta_\sigma + B_{R_5, f_2}}$$

450 and

$$\begin{aligned} \beta_{FC, R_{13}, f_2}^l &= \overline{\beta_{LT}^l \otimes \beta_{R_9, f_2}^l \otimes \overline{\beta_{LT}^l \otimes \beta_{R_5, f_2}^l \otimes \delta_\sigma + B_{R_5, f_2}} \otimes \delta_\sigma + B_{R_9, f_2}} \\ &= \overline{(\beta_{LT}^l \otimes \beta_{R_9, f_2}^l \otimes \delta_\sigma + B_{R_9, f_2}) \otimes \overline{\beta_{LT}^l \otimes \beta_{R_5, f_2}^l \otimes \delta_\sigma + B_{R_5, f_2}}} \\ &= \overline{\beta_{LT}^l \otimes \beta_{R_9, f_2}^l \otimes \delta_\sigma + B_{R_9, f_2}} \otimes \overline{\beta_{LT}^l \otimes \beta_{R_5, f_2}^l \otimes \delta_\sigma + B_{R_5, f_2}} \end{aligned}$$

where the step from the first line to the second line holds due to the basic property of min-plus convolution (see Rule 7 of Theorem 3.1.5 in [17]). By applying Theorem 3.1.11 in [17], the last line holds.

Similarly, we can prove that

$$\begin{aligned}\beta_{FC,R_{14},f_2}^l &= \overline{\beta_{LT}^l \otimes \beta_{R_{13},f_2}^l \otimes \delta_\sigma + B_{R_{13},f_2}} \otimes \beta_{FC,R_{13},f_2}^l, \\ \beta_{FC,R_{15},f_2}^l &= \overline{\beta_{LT}^l \otimes \beta_{R_{14},f_2}^l \otimes \delta_\sigma + B_{R_{13},f_2}} \otimes \beta_{FC,R_{14},f_2}^l, \\ \beta_{FC,NI,f_2}^l &= \overline{\beta_{LT}^l \otimes \beta_{R_{15},f_2}^l \otimes \delta_\sigma + B_{R_{15},f_2}} \otimes \beta_{FC,R_{15},f_2}^l.\end{aligned}$$

Then, the equivalent feedback service curve obtained by  $f_2$  is

$$\begin{aligned}\beta_{f_2}^l &= \beta_{NI,f_2}^l \otimes \beta_{FC,NI,f_2}^l \otimes \left( \bigotimes_{R_j \in \mathcal{R}_2} (\beta_{R_j,f_2}^l \otimes \beta_{FC,R_j,f_2}^l \otimes \beta_{LT}^l) \right) \\ &= \beta_{NI,f_2}^l \otimes (\beta_{LT}^l \otimes \left( \bigotimes_{R_j \in \mathcal{R}_2} \beta_{R_j,f_2}^l \right)) \otimes (\beta_{FC,NI,f_2}^l \otimes \left( \bigotimes_{R_j \in \mathcal{R}_2} \beta_{FC,R_j,f_2}^l \right)) \\ &= \beta_{NI,f_2}^l \otimes (\beta_{LT}^l \otimes \left( \bigotimes_{R_j \in \mathcal{R}_2} \beta_{R_j,f_2}^l \right)) \otimes \left( \bigotimes_{R_j \in \mathcal{R}_2} \overline{\beta_{R_j,f_2}^l \otimes \beta_{LT}^l \otimes \delta_\sigma + B_{R_j,f_2}^l} \right) \\ &= \beta_{NI,f_2}^l \otimes \left( \bigotimes_{R_j \in \mathcal{R}_2} (\beta_{R_j,f_2}^l \otimes \beta_{LT}^l \otimes \overline{\beta_{LT}^l \otimes \beta_{R_j,f_2}^l \otimes \delta_\sigma + B_{R_j,f_2}^l}) \right)\end{aligned}$$

<sup>455</sup> where the second line and the fourth line hold due to the commutativity of min-plus convolution, and the step from the second line to the third line holds due to the basic property of sub-additive closure (see Corollary 3.1.1 in [17]).

According to the isotonicity of min-plus convolution (see Theorem 3.1.7 in [17]), we know that

$$\beta_{R_j,f_2}^l \otimes \beta_{LT}^l \otimes \beta_{R_j,f_2}^l \otimes \beta_{LT}^l \otimes \delta_\sigma(t) + B_{R_j,f_2}^l \geq \beta_{R_j,f_2}^l \otimes \beta_{LT}^l, \forall R_j \in \mathcal{R}_2 \quad (6)$$

is a sufficient condition for

$$\beta_{f_2}^l \geq \beta_{NI,f_2}^l \otimes \left( \bigotimes_{R_j \in \mathcal{R}_2} (\beta_{R_j,f_2}^l \otimes \beta_{LT}^l) \right)$$

<sup>460</sup> where the right side is the equivalent end-to-end feed-forward service curve provided to  $f_2$ . Thus, to avoid flow control, the buffer size at each router reserved for flow  $f_2$  can be assigned to the minimum integer value to make Eq.(6) hold, which verifies the correctness of this theorem. ■

<sup>465</sup> Suppose that the applications have been mapped onto the NoC, and each flow  $f_i$  has been assigned to their corresponding priority  $P_i$  and deadline  $D_i$ . Following the same notation as Algorithm 2, we propose the buffer sizing algorithm to allocate just enough buffer for each flow to meet their deadline constraint, as shown in Algorithm 3. We assume that the service curve provided by the source NI of  $f_i$ , i.e.  $\langle \beta_{NI,f_i}^l, \beta_{NI,f_i}^u \rangle$ , has been computed with Algorithm 1.

<sup>470</sup> Our algorithm tries to reduce the buffer size for each flow from high-priority to low-priority iteratively. For each iteration, it performs the following four steps: (1) Calculate the equivalent feed-forward service curves provided by the routers (lines 3-5). (2) Calculate the initial buffer size to avoid flow control for each router (line 6). (3)

---

**Algorithm 3** Buffer sizing algorithm

---

**Require:** Architecture parameters and flow specifications  
**Ensure:** Buffer required for each flow to meet their deadline

- 1: **for** each flow  $f_i \in \mathcal{F}$  with priority order **do**
- 2:   **for** each router  $R_j \in \mathcal{R}_i$  **do**
- 3:      $\beta_{FC,R_j,f_i}^l = \delta_0(t); \beta_{FC,R_j,f_i}^u = \delta_0(t);$
- 4:      $\beta_{R_j,f_i}^l = \beta_{BW}^l \otimes \beta_{RC}^l \otimes \beta_{VA}^l \otimes \beta_{SA}^l \otimes \lfloor \frac{\beta_{ST,R_j}^{l'}}{|\Theta_{R_j,f_i}|+1} \rfloor \otimes \beta_{FC,R_j,f_i}^l;$
- 5:      $\beta_{R_j,f_i}^u = \beta_{BW}^u \otimes \beta_{RC}^u \otimes \beta_{VA}^u \otimes \beta_{SA}^u \otimes \lceil \frac{\beta_{ST,R_j}^{u'}}{|\Theta_{R_j,f_i}|+1} \rceil \otimes \beta_{FC,R_j,f_i}^u;$
- 6:     compute the initial buffer size  $B_{R_j,f_i}$  according to Theorem 3;
- 7:   **end for**
- 8:      $\beta_{FC,NI,f_i}^l = \delta_0(t); \beta_{FC,NI,f_i}^u = \delta_0(t);$
- 9:     Buffer\_Reduction( $\alpha_{f_i}^u, \mathcal{R}_i, D_i$ );                      ▷ Call the Buffer\_Reduction procedure
- 10:      $\beta_{f_i}^l = \beta_{NI,f_i}^l \otimes \beta_{FC,NI,f_i}^l; \beta_{f_i}^u = \beta_{NI,f_i}^u \otimes \beta_{FC,NI,f_i}^u;$
- 11:     **for**  $\forall R_j \in \mathcal{R}_i$  from  $start_i$  to  $end_i$  **do**
- 12:       **if**  $\Omega_{R_j,f_i} \neq \emptyset$  **then**
- 13:          $\beta^l = \beta_{f_i}^l \otimes \beta_{BW}^l \otimes \beta_{RC}^l \otimes \beta_{VA}^l \otimes \beta_{SA}^l;$
- 14:          $\beta^u = \beta_{f_i}^u \otimes \beta_{BW}^u \otimes \beta_{RC}^u \otimes \beta_{VA}^u \otimes \beta_{SA}^u;$
- 15:          $\alpha_{R_j,f_i}^l = \min\{(\alpha_{f_i}^l \otimes \beta^u) \otimes \beta^l, \beta^l\};$
- 16:          $\alpha_{R_j,f_i}^u = \min\{(\alpha_{f_i}^u \otimes \beta^u) \otimes \beta^l, \beta^u\};$
- 17:         **if**  $\forall f_k \in \Theta_{R_j,f_i}$ ,  $Delay(f_k)$  has been calculated **then**
- 18:            $\alpha_{R_j,f_i}^l = \alpha_{R_j,f_i}^l + \sum_{f_k \in \Theta_{R_j,f_i}} \alpha_{R_j,f_k}^l;$
- 19:            $\alpha_{R_j,f_i}^u = \alpha_{R_j,f_i}^u + \sum_{f_k \in \Theta_{R_j,f_i}} \alpha_{R_j,f_k}^u;$
- 20:            $\beta_{ST,R_j}^{l'} = (\beta_{ST,R_j}^{l'} - \alpha_{R_j,f_i}^u) \bar{\otimes} 0;$
- 21:            $\beta_{ST,R_j}^{u'} = \max\{(\beta_{ST,R_j}^{u'} - \alpha_{R_j,f_i}^l) \bar{\otimes} 0, 0\};$
- 22:         **end if**
- 23:       **end if**
- 24:        $\beta_{f_i}^l = \beta_{f_i}^l \otimes \beta_{R_j,f_i}^l \otimes \beta_{LT}^l; \beta_{f_i}^u = \beta_{f_i}^u \otimes \beta_{R_j,f_i}^u \otimes \beta_{LT}^u;$
- 25:     **end for**
- 26: **end for**

---

Reduce the initial buffer size gradually as long as the constraint of deadline is not being violated (line 9). This step is abstracted as a procedure, i.e. Buffer\_Reduction( $f_i$ ).  
 475 This procedure tries to reduce the buffer size of each router reserved for the target flow  $f_i$  from its ejection router to its injection router. For each router on the path of  $f_i$ , it decreases the buffer size iteratively, until the buffer size is reduced to one or the deadline constraint is violated. If the iteration stops due to the violation of deadline, the buffer size and related service curves are restored to their previous value.  
 480 To ease the description, we define the operator  $Pre(R_j)$  to identify the predecessor of router  $R_j$ , and let  $Pre(start_i) = NI$ . (4) Calculate the leftover service curve at each router for low-priority flows (lines 10-25). The computation complexity of the

485 Buffer\_Reduction( $f_i$ ) procedure and the entire buffer sizing algorithm are  $O(H^2)$  and  $O(NH^2)$ , where  $N$  and  $H$  denote the number of flows analyzed and the maximum hop count of these flows. This algorithm can be implemented in RTC toolbox [20] to optimize the buffer size automatically.

---

```

1: procedure BUFFER_REDUCTION( $\alpha_{f_i}^u, \mathcal{R}_i, D_i$ )
2:   for each router  $R_j \in \mathcal{R}_i$  from  $end_i$  to  $start_i$  do
3:      $Delay(f_i) = H(\alpha_{f_i}^u, \beta_{NI,f_i}^l \otimes \beta_{FC,NI,f_i}^l \otimes (\bigotimes_{R_k \in \mathcal{R}_i} (\beta_{R_k,f_i}^l \otimes \beta_{LT}^l)))$ ;
4:     while  $Delay(f_i) \leq D_i$  and  $B_{R_j,f_i} > 1$  do
5:        $B_{R_j,f_i} = B_{R_j,f_i} - 1$ ;
6:       for all  $R_k \in \mathcal{R}_i$  from  $Pre(R_j)$  to  $start_i$  do
7:         compute  $\langle \beta_{FC,R_k,f_i}^l, \beta_{FC,R_k,f_i}^u \rangle$  according to Theorem 2;
8:         compute the service curve  $\langle \beta_{R_k,f_i}^l, \beta_{R_k,f_i}^u \rangle$ ;
9:       end for
10:      compute the service curve  $\langle \beta_{FC,NI,f_i}^l, \beta_{FC,NI,f_i}^u \rangle$ ;
11:       $Delay(f_i) = H(\alpha_{f_i}^u, \beta_{NI,f_i}^l \otimes \beta_{FC,NI,f_i}^l \otimes (\bigotimes_{R_k \in \mathcal{R}_i} (\beta_{R_k,f_i}^l \otimes \beta_{LT}^l)))$ ;
12:    end while
13:    if  $Delay(f_i) > D_i$  then
14:       $B_{R_j,f_i} = B_{R_j,f_i} + 1$ ;
15:      recompute  $\langle \beta_{FC,Pre(R_j),f_i}^l, \beta_{FC,Pre(R_j),f_i}^u \rangle$ ;
16:      recompute  $\langle \beta_{Pre(R_j),f_i}^l, \beta_{Pre(R_j),f_i}^u \rangle$  if  $R_j \neq start_i$ ;
17:    end if
18:  end for
19: end procedure

```

---

490 The number of cycles that a packet can be delayed in the network without violating its deadline is referred to ‘slack’ in [29]. Thus, the Buffer\_Reduction procedure of our  
495 buffer sizing algorithm is the process of slack minimization. However, while used for power reduction, our buffer sizing algorithm is significantly different from the DNC-based slack optimization algorithm proposed in [29]. In [29], the energy optimization is achieved by adjusting the voltage, frequency and link bandwidth of on-chip routers for the fixed configuration and deadline. In contrast, our method tries to reduce the buffer size under the deadline constraint, and the buffer reduction directly leads to the area and power saving. In addition, our algorithm can be also used in conjunction with the priority sharing techniques [28] to minimize the hardware cost of priority-aware wormhole-switched NoC.

## 5. Experiments

500 In this section, we validate the correctness and tightness of our performance model by comparing with simulation results and other analytical methods. Several analytical methods exist for the delay analysis of priority-aware NoC, examples include contention tree model [9], lumped link model [10], dependency graph model [11], FLA

[6], LLA [7] and DNC [8], etc. There is also extensive research on the buffer sizing problem of the priority-aware NoC, representative methods include shaping delay analysis [30] and LLBA [12]. Among all these analytical methods, DNC [8] and LLBA [12] based models outperform the others when the tightness of delay bound and buffer requirement is considered. Thus, we will only perform the comparison with LLBA and DNC, as presented in subsection 5.1 and subsection 5.2 respectively. We also present the simulation results to validate the correctness and tightness of our method in subsection 5.3.

### 5.1. Comparison with Link Level Buffer-space Analysis

The traffic pattern discussed in this subsection is shown in Fig. 1. There are four flows (i.e.  $f_1, f_2, f_3$  and  $f_4$ ) in the network, with different priorities  $P_4 > P_1 > P_2 > P_3$ . We perform the comparison on a set of periodical traffic due to the restriction of LLBA method [12]. The packet length (in flits) and injection period (in cycles) of flow  $f_i$  ( $i = 1, 2, 3, 4$ ) are denoted as  $F_i$  and  $I_i$ , respectively. To ease the analysis of buffer requirement, the LLBA method assumes the number of bits in a flit is the same as the physical channel width, and the latency of a router is one cycle [12]. Thus, our performance model for the standard wormhole-switched router should be specialized, which is achieved by letting the service curve of BW, RC, VA, SA and LT stage be a pair of burst delay function  $\langle \delta_0(t), \delta_0(t) \rangle$ . Under this condition, the service curve of the entire router is equal to the service curve provided by the ST stage, which is  $\langle \beta_{ST,R_i}^l, \beta_{ST,R_i}^u \rangle$ . The traffic jitter for all the flows are assumed to be zero for brevity and clarity. In addition, we set the credit feedback delay  $\sigma = 0$  cycle in our model since the LLBA method does not consider the influence of credit feedback delay on the buffer requirement.

Let us suppose that all the flows have the same packet injection period  $I_i = 50$  cycles and packet length  $F_i = 8$  flits ( $i = 1, 2, 3, 4$ ). The delay bounds computed with the LLA method [7] for these four flows are 21, 22, 21 and 13 cycles, respectively. We can also obtain the total buffer size reserved for the four flows (i.e.  $f_1, f_2, f_3$  and  $f_4$ ) with LLBA method [12], which are 11, 12, 18 and 4 flits, respectively. The total buffer size required by these four flows can be obtained by summing up the buffer size reserved for each flow along its path, which is 45 flits. This value is the minimum buffer space required for all these four flows to avoid the back-pressure caused by flow control. By allowing the flow control to be triggered, our buffer sizing algorithm can be utilized to reduce this buffer size further as long as the deadline constraint is not violated. Figure 6 demonstrates the normalized buffer requirement calculated with LLBA and our method under different deadline constraint. It can be found that our method can give much tighter buffer estimation than LLBA, and the improvement of our method over LLBA becomes more and more significant as the deadline constraint is relaxed.

### 5.2. Comparison with Network Calculus based Delay Analysis

In this subsection, we present the numerical results to demonstrate the improvement of our method over the DNC method proposed in [8]. The traffic pattern discussed in this subsection is shown in Fig. 1. The priorities of these four flows (i.e.  $f_1, f_2, f_3$

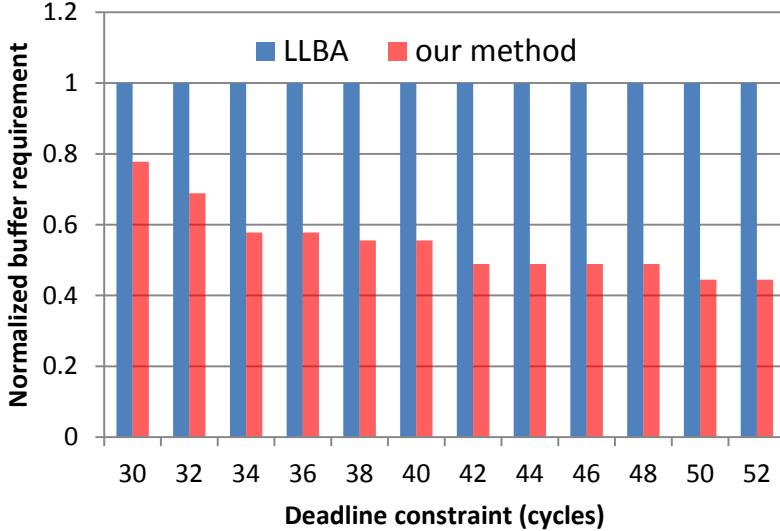


Figure 6: Buffer requirement computed with our method for different deadline

and  $f_4$ ) in the network satisfy  $P_4 > P_1 > P_2 > P_3$ . We perform the comparison on a set of periodical traffic. The packet length (in flits) and injection period (in cycles) of flow  $f_i$  ( $i = 1, 2, 3, 4$ ) are denoted as  $F_i$  and  $I_i$ , respectively. The router architecture we considered in this subsection adopts the lookahead pipeline [23], which removes the RC stage from the critical path of the router pipeline. Thus, our service model is customized by letting the service curve of RC stage to be  $\langle \delta_0(t), \delta_0(t) \rangle$ . The RTC arrival curve  $\langle \alpha_{f_i}^l, \alpha_{f_i}^u \rangle$  of flow  $f_i$  can be obtained according to the method introduced in subsection 4.1. The DNC arrival curve for  $f_i$  is  $\alpha_{f_i} = V_i t + F_i$ , where  $V_i = F_i/I_i$  represents the average arrival rate.

We assume the VC buffer size of each router is 32 flits, and the credit feedback delay  $\sigma = 0$  cycle. We change the injection rate  $V_i$  ( $i = 1, 2, 3, 4$ ) from  $1/3$  to  $1/6$  (flits/cycle) and the packet length  $F_i$  from 1 to 8 flits. The end-to-end delay of flow  $f_3$  calculated with the DNC method and our method are presented in Fig. 7. By comparison, we find that our method can give a much tighter delay bound than the DNC-based method proposed in [8]. The root cause for this improvement lies in the fact that our method utilizes the upper service curve to limit the output upper arrival curve, which further leads to a tighter leftover service curve for the low-priority flows.

### 5.3. Comparison with Simulation

Our performance model is also verified by simulation. We modified the Booksim 2.0 [31], a cycle-accurate NoC simulator, to support the specified traffic pattern and injection process. The traffic patterns investigated in this subsection include the example shown in Fig. 1 and an real application provided by Ericsson Radio Systems [32][33]. We adopt the optimized lookahead router [23] to construct the mesh network. To fit this optimization, our service model is customized by letting the service curve of RC

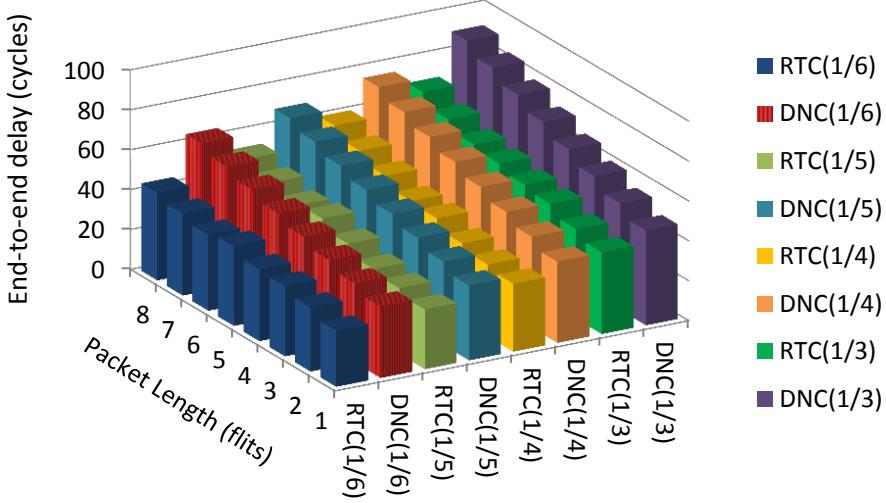


Figure 7: Delay bound comparison with network calculus under different injection rate and packet length

stage be  $\langle \delta_0(t), \delta_0(t) \rangle$ . Other architecture and simulation parameters used in the simulation are listed in the Table 1.

Table 1: Architecture parameters used in the simulation

network topology	$4 \times 4$ mesh	routing algorithm	X-Y routing
credit delay	0 cycle	channel width	128 bits
buffer size	32 flits	switch allocator	priority-based
link latency	1 cycle	sampling period	$1 \times 10^5$ cycles
clock cycle	1 ns	warmup period	$3 \times 10^5$ cycles

We first investigate the traffic pattern shown in Fig. 1. Suppose that the injection process of each flow is the strictly periodical traffic without any jitter, and the priorities of these four flows satisfy  $P_4 > P_1 > P_2 > P_3$ . However, even under this assumption, determining the worst-case delay by simulation for general scenarios is still a non-trivial task, because the worst-case delay of each flow depends on the traffic pattern and offset (i.e. the injection time of the first packet in a flow  $f_i$ , denoted as  $O_i$ ) of all the flows in the network.

We set the injection rate  $V_i$  ( $i = 1, 2, 3, 4$ ) to  $1/6$  (flits/cycle), and change the packet length  $F_i$  from 2 flits to 9 flits. Under these conditions, we find that  $f_1$  will experience the maximum delay in the network if its offset  $O_1$  is equal to that of  $f_4$ . Similarly, the worst-case interference of  $f_2$  occurs when its offset  $O_2 = O_1 + (4 + 1)$ , where  $O_1 + (4 + 1)$  is the time instance that the first packet of  $f_1$  leaves router  $R_{16}$ , since the basic latency of a lookahead router and physical channel are 4 cycles and 1 cycles, respectively. Flow  $f_3$  experiences the worst-case delay when  $O_3 = O_2 + F_1 + 2 \times$

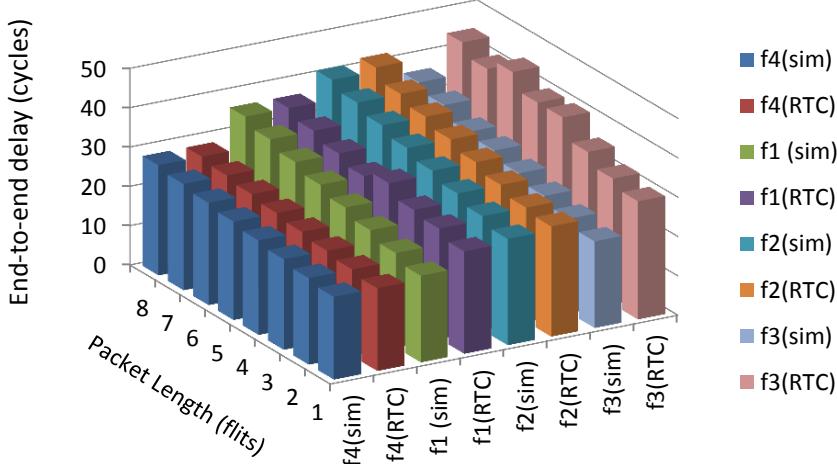


Figure 8: Delay comparison with simulation under different packet length

(4 + 1). Thus, we set  $O_1 = O_4 = 0$  cycle,  $O_2 = 5$  cycles,  $O_3 = 15 + F_1$  cycles, and run the simulation. The collected maximum end-to-end delay of these four flows under the given offset combination are compared with our RTC method, as shown in Fig. 8. 590 As indicated in this figure, for the given configurations, the delay bounds calculated with our method are indeed the upper bounds of simulation results, which verifies the correctness of our method.

We also take a real application discussed in [32][33] as an example to demonstrate the tightness and ability of our method to analyze complex scenarios. This application 595 is comprised of 16 IP cores. The 26 communication flows among these 16 IPs are classified into nine groups, and each group has their bandwidth requirement. When mapped to a  $4 \times 4$  mesh network, the traffic pattern of this application is demonstrated in Fig. 9(a). We assume that all the flows in a group send packets periodically with the same injection period and priority, as listed in Table. 9(b). We set the packet size to 128 bits, and collect the maximum end-to-end delay of each flow obtained by simulation. 600 The comparison results between one simulation run and the delay bound calculated with our method are shown in Fig. 10. The offset of each flow in this simulation run is zero cycle. We can see that the calculated delay bounds constrain the simulation results well, which verifies the correctness of our method. This comparison also demonstrates 605 the ability of our method to analyze the real system with large number of flows.

## 6. Conclusion

The priority-aware wormhole-switched NoC is a promising platform for the on-chip real-time communication if the worst-case performance can be accurately analyzed and guaranteed. Simulation is not well suited for this purpose because it is 610 difficult to cover all the corner cases. In this paper, we proposed an RTC-based performance model to achieve this goal. We first built the traffic model and service model for

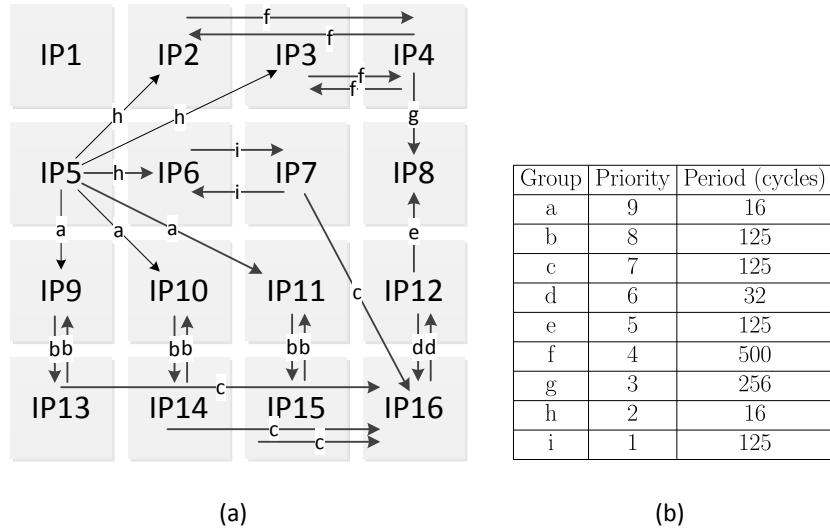


Figure 9: Traffic pattern of ericsson radio system application

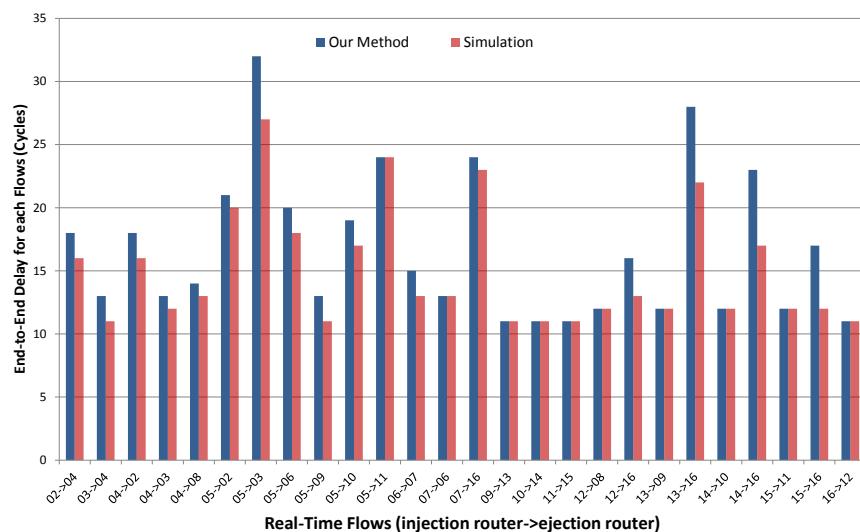


Figure 10: Delay comparison with simulation

the priority-aware NoC, and then proposed a novel method to derive the upper service curve of credit-based flow control. Based on the proposed RTC model, we then proposed an end-to-end delay analysis algorithm and a buffer sizing algorithm. The delay analysis algorithm can be implemented to compute the end-to-end delay for each flow automatically, and verify whether all these flows meet their deadline under the given configuration. Compared with the DNC-based performance model, our model can give tighter delay bound since it takes the upper service curve and lower arrival curve into consideration. The proposed buffer sizing algorithm can optimize the buffer size from high-priority flows to low-priority flows. It can also be implemented to perform the buffer reduction automatically under the constraint of deadline. Compared with the LLBA method, our method can give much tighter buffer bound. Experimental results also illustrate that our method indeed outperforms the other analytical methods, e.g. LLBA and DNC, when the tightness of delay bound and buffer requirement are considered. Our method can be applied to the task mapping, routing and power reduction of priority-aware NoC.

### Acknowledgements

The authors thank the reviewers for their suggestions and comments, and all the experiments are carried out at the Integrated Microsystem Lab (IML) of McGill University. The first author also thank Ari Ramdial at McGill University for his helpful comments. This research is supported by High Technology Research and Development Program of China (Grant No. 2012AA012201).

### References

#### References

- [1] K. Goossens, J. Dielissen, A. Radulescu, *Æthereal network on chip:concepts, architectures, and implementations*, IEEE Design & Test of Computers 22 (5) (2005) 414–421. doi:<http://doi.ieee.org/10.1109/MDT.2005.99>.
- [2] S. Liu, A. Jantsch, Z. Lu, Analysis and evaluation of circuit switched noc and packet switched noc, in: Digital System Design (DSD), 2013 Euromicro Conference on, 2013, pp. 21–28. doi:[10.1109/DSD.2013.13](https://doi.org/10.1109/DSD.2013.13).
- [3] C. Paukovits, H. Kopetz, Concepts of switching in the time-triggered network-on-chip, in: Embedded and Real-Time Computing Systems and Applications, 2008. RTCSA '08. 14th IEEE International Conference on, 2008, pp. 120–129. doi:[10.1109/RTCSA.2008.18](https://doi.org/10.1109/RTCSA.2008.18).
- [4] P. Kundu, On-die interconnects for next generation cmps, in: 2006 Workshop on On- and Off-Chip Interconnection Networks for Multicore Systems, Stanford, CA, USA, 2006.

- 650 [5] G. Michelogiannakis, D. Sanchez, W. Dally, C. Kozyrakis, Evaluating bufferless  
flow control for on-chip networks, in: Networks-on-Chip (NOCS), 2010 Fourth  
ACM/IEEE International Symposium on, 2010, pp. 9–16. doi:10.1109/  
NOCS.2010.10.
- 655 [6] Z. Shi, A. Burns, Real-time communication analysis for on-chip networks  
with wormhole switching, in: Networks-on-Chip, 2008. NoCS 2008. Second  
ACM/IEEE International Symposium on, 2008, pp. 161–170. doi:10.1109/  
NOCS.2008.4492735.
- 660 [7] H. Kashif, H. D. Patel, S. Fischmeister, Using link-level latency analysis for path  
selection for real-time communication on nocs, in: Proceedings of the Asia South  
Pacific Design Automation Conference (ASPDAC), Sydney, Australia, 2012, pp.  
499–504. doi:10.1109/ASPDAC.2012.6165004.
- [8] Y. Qian, Z. Lu, Q. Dou, Qos scheduling for nocs: Strict priority queueing versus  
weighted round robin, in: Computer Design (ICCD), 2010 IEEE International  
Conference on, 2010, pp. 52–59. doi:10.1109/ICCD.2010.5647577.
- 665 [9] Z. Lu, A. Jantsch, I. Sander, Feasibility analysis of messages for on-chip networks  
using wormhole routing, in: Design Automation Conference, 2005. Proceedings  
of the ASP-DAC 2005. Asia and South Pacific, Vol. 2, 2005, pp. 960–964 Vol. 2.  
doi:10.1109/ASPDAC.2005.1466499.
- 670 [10] S. Balakrishnan, F. Özgürer, A priority-driven flow control mechanism for real-  
time traffic in multiprocessor networks, IEEE Trans. Parallel Distrib. Syst. 9 (7)  
(1998) 664–678. doi:10.1109/71.707545.
- [11] B. Kim, J. Kim, S. Hong, S. Lee, A real-time communication method for worm-  
hole switching networks, in: Parallel Processing, 1998. International Conference  
on, 1998, pp. 527–534. doi:10.1109/ICPP.1998.708526.
- 675 [12] H. Kashif, H. Patel, Bounding buffer space requirements for real-time priority-  
aware networks, in: Design Automation Conference (ASP-DAC), 2014 19th A-  
sia and South Pacific, 2014, pp. 113–118. doi:10.1109/ASPDAC.2014.  
6742875.
- 680 [13] S. Chakraborty, S. Kunzli, L. Thiele, A general framework for analysing system  
properties in platform-based embedded system designs, in: Design, Automation  
and Test in Europe Conference and Exhibition, 2003, 2003, pp. 190–195. doi:  
10.1109/DATE.2003.1253607.
- [14] E. Bolotin, I. Cidon, R. Ginosar, A. Kolodny, Qnoc: Qos architecture and design  
process for network on chip, J. Syst. Archit. 50 (2-3) (2004) 105–128. doi:  
10.1016/j.sysarc.2003.07.004.
- 685 [15] Z. Shi, Real-time communication services for networks on chip, Ph.D. thesis, The  
University of York (2009).

- [16] Y. Qian, Z. Lu, W. Dou, Analysis of worst-case delay bounds for best-effort communication in wormhole networks on chip, in: Networks-on-Chip (NoCs), 2009. 3rd ACM/IEEE International Symposium on, 2009, pp. 44–53. doi:10.1109/NOCS.2009.5071444.
- [17] J.-Y. Le Boudec, P. Thiran, Network Calculus: A Theory of Deterministic Queuing Systems for the Internet, Springer-Verlag, Berlin, Heidelberg, 2001.
- [18] D. Chokshi, P. Bhaduri, Modeling fixed priority non-preemptive scheduling with real-time calculus, in: Embedded and Real-Time Computing Systems and Applications, 2008. RTCSA '08. 14th IEEE International Conference on, 2008, pp. 387–392. doi:10.1109/RTCSA.2008.28.
- [19] A. Hagiescu, U. D. Bordoloi, S. Chakraborty, P. Sampath, P. V. V. Ganesan, S. Ramesh, Performance analysis of flexray-based ecu networks, in: Proceedings of the 44th Annual Design Automation Conference, DAC '07, ACM, New York, NY, USA, 2007, pp. 284–289. doi:10.1145/1278480.1278554.
- [20] E. Wandeler, L. Thiele, Real-time calculus (rtc) toolbox, <http://www.mpa.ethz.ch/Rtctoolbox> (2006).
- [21] S. Hary, F. Ozguner, Feasibility test for real-time communication using wormhole routing, Computers and Digital Techniques, IEE Proceedings - 144 (5) (1997) 273–278. doi:10.1049/ip-cdt:19971369.
- [22] W. Dally, B. Towles, Principles and Practices of Interconnection Networks, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.
- [23] N. Jerger, L. Peh, On-chip networks, Synthesis Lectures on Computer Architecture 4 (1) (2009) 1–141.
- [24] W. Dally, S. Tell, The even/odd synchronizer: A fast, all-digital, periodic synchronizer, in: Asynchronous Circuits and Systems (ASYNC), 2010 IEEE Symposium on, 2010, pp. 75–84. doi:10.1109/ASYNC.2010.20.
- [25] S. Lee, Real-time wormhole channels, J. Parallel Distrib. Comput. 63 (3) (2003) 299–311. doi:10.1016/S0743-7315(02)00055-2.
- [26] H. Schiøler, J. J. Jessen, J. D. Nielsen, K. G. Larsen, Network calculus for real time analysis of embedded systems with cyclic task dependencies., in: Proc. 20th International Conference on Computers and Their Applications, CATA 2005, 2005, pp. 326–332.
- [27] L. Thiele, N. Stoimenov, Modular performance analysis of cyclic dataflow graphs, in: Proceedings of the Seventh ACM International Conference on Embedded Software, EMSOFT '09, ACM, New York, NY, USA, 2009, pp. 127–136. doi:10.1145/1629335.1629353.
- [28] Z. Shi, A. Burns, Real-time communication analysis with a priority share policy in on-chip networks, in: Real-Time Systems, 2009. ECRTS '09. 21st Euromicro Conference on, 2009, pp. 3–12. doi:10.1109/ECRTS.2009.17.

- [29] J. Zhan, N. Stoimenov, J. Ouyang, L. Thiele, V. Narayanan, Y. Xie, Designing energy-efficient noc for real-time embedded systems through slack optimization, in: Design Automation Conference (DAC), 2013 50th ACM / EDAC / IEEE, 2013, pp. 1–6. doi:10.1145/2463209.2488780.
- 730 [30] S. Manolache, P. Eles, Z. Peng, Buffer space optimisation with communication synthesis and traffic shaping for nos, in: Design, Automation and Test in Europe, 2006. DATE '06. Proceedings, Vol. 1, 2006, pp. 1–6. doi:10.1109/DAT.2006.244069.
- 735 [31] N. Jiang, D. Becker, G. Michelogiannakis, J. Balfour, B. Towles, D. Shaw, J. Kim, W. Dally, A detailed and flexible cycle-accurate network-on-chip simulator, in: Performance Analysis of Systems and Software (ISPASS), 2013 IEEE International Symposium on, 2013, pp. 86–96. doi:10.1109/ISPASS.2013.6557149.
- 740 [32] Z. Lu, A. Jantsch, Tdm virtual-circuit configuration for network-on-chip, Very Large Scale Integration (VLSI) Systems, IEEE Transactions on 16 (8) (2008) 1021–1034. doi:10.1109/TVLSI.2008.2000673.
- [33] F. Jafari, Z. Lu, A. Jantsch, M. H. Yaghmaee, Buffer optimization in network-on-chip through flow regulation, Trans. Comp.-Aided Des. Integ. Cir. Sys. 29 (12) (2010) 1973–1986. doi:10.1109/TCAD.2010.2063130.



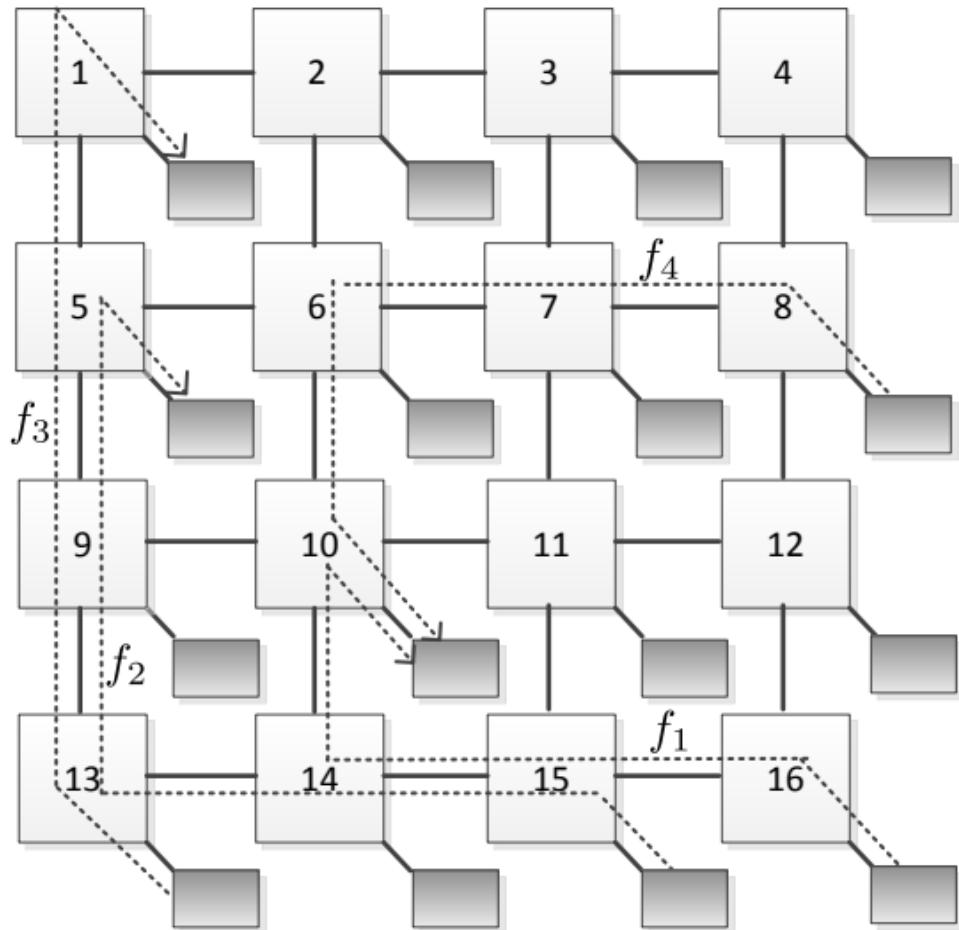
**Baoliang Li** was born in 1987. He received the B.S. degree in Computer Science and Technology from Tsinghua University, P.R. China, in 2009. He is currently working towards the Ph.D. degree from College of Computer at National University of Defense Technology (NUDT), P.R. China. His research interests is performance analysis of computer networks and Networks-on-Chip, with a special interest on the analytical methods of network calculus.



**Zeljko Zilic** received the B.Eng. degree from the University of Zagreb, Zagreb, Croatia, and the M.Sc. and Ph.D. degrees from the University of Toronto, Toronto, ON, Canada, all in electrical and computer engineering. He joined McGill University, Montreal, QC, Canada, in 1998, where he is currently an Associate Professor with the Department of Electrical and Computer Engineering. He conducts research on various aspects of the design and test of microsystems including programmable logic cores.



**Wenhua Dou** was born in 1946. He received the B.S. degree in computer science from Harbin Military Engineering College in 1970. He has been working at National University of Defense Technology (NUDT), P.R. China since 1970. He was vice dean of College of Computer, NUDT from 1999 to 2003. He is currently a professor of College of Computer, NUDT with research focusing on computer networks.



Router



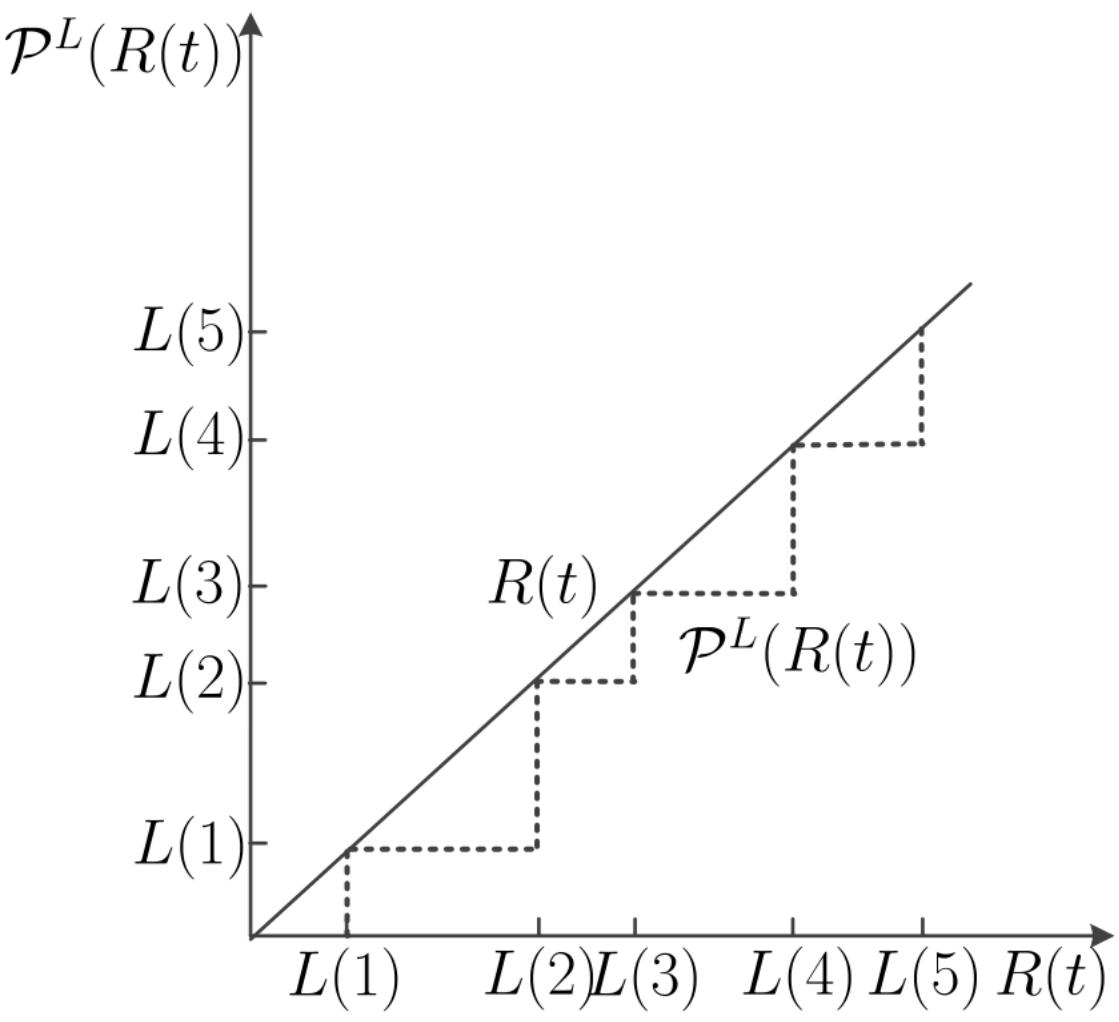
Network Interface

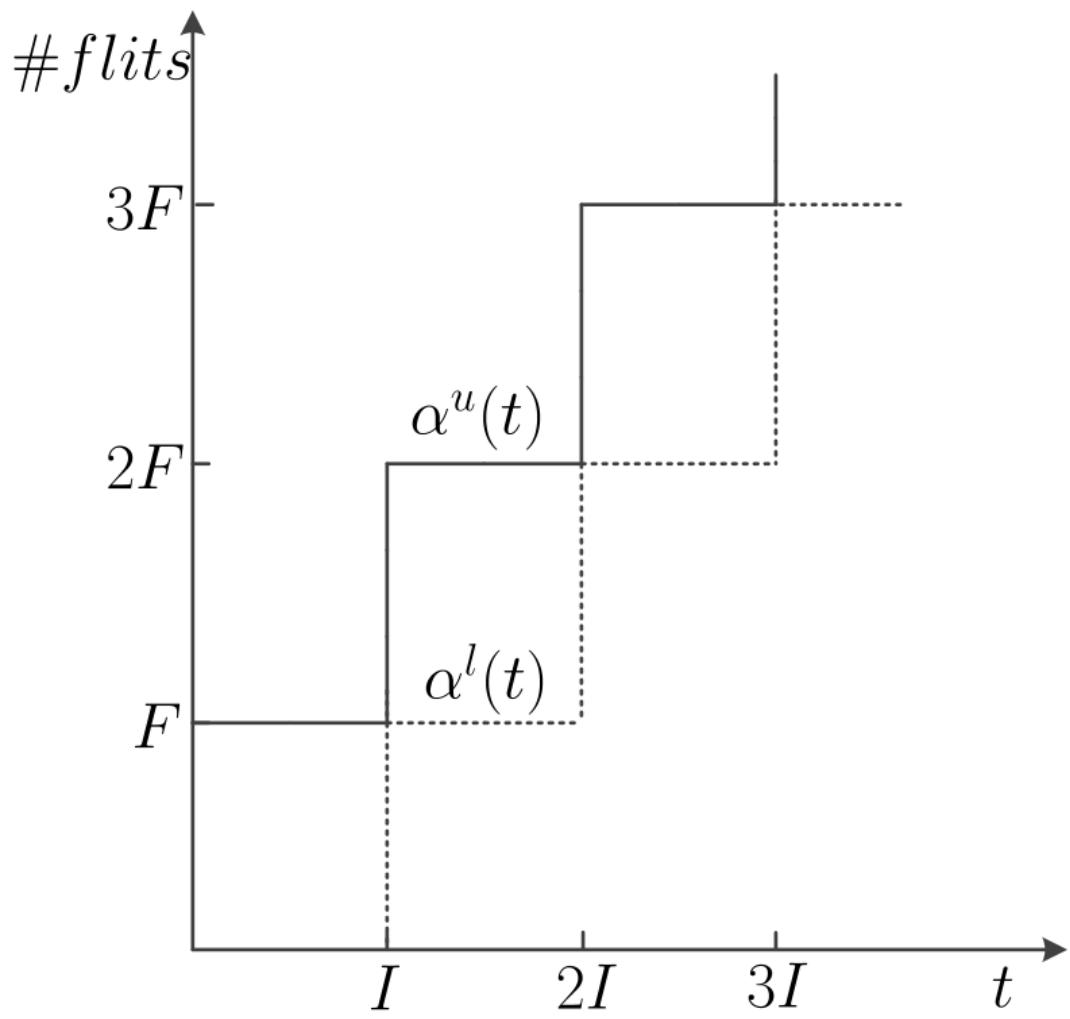


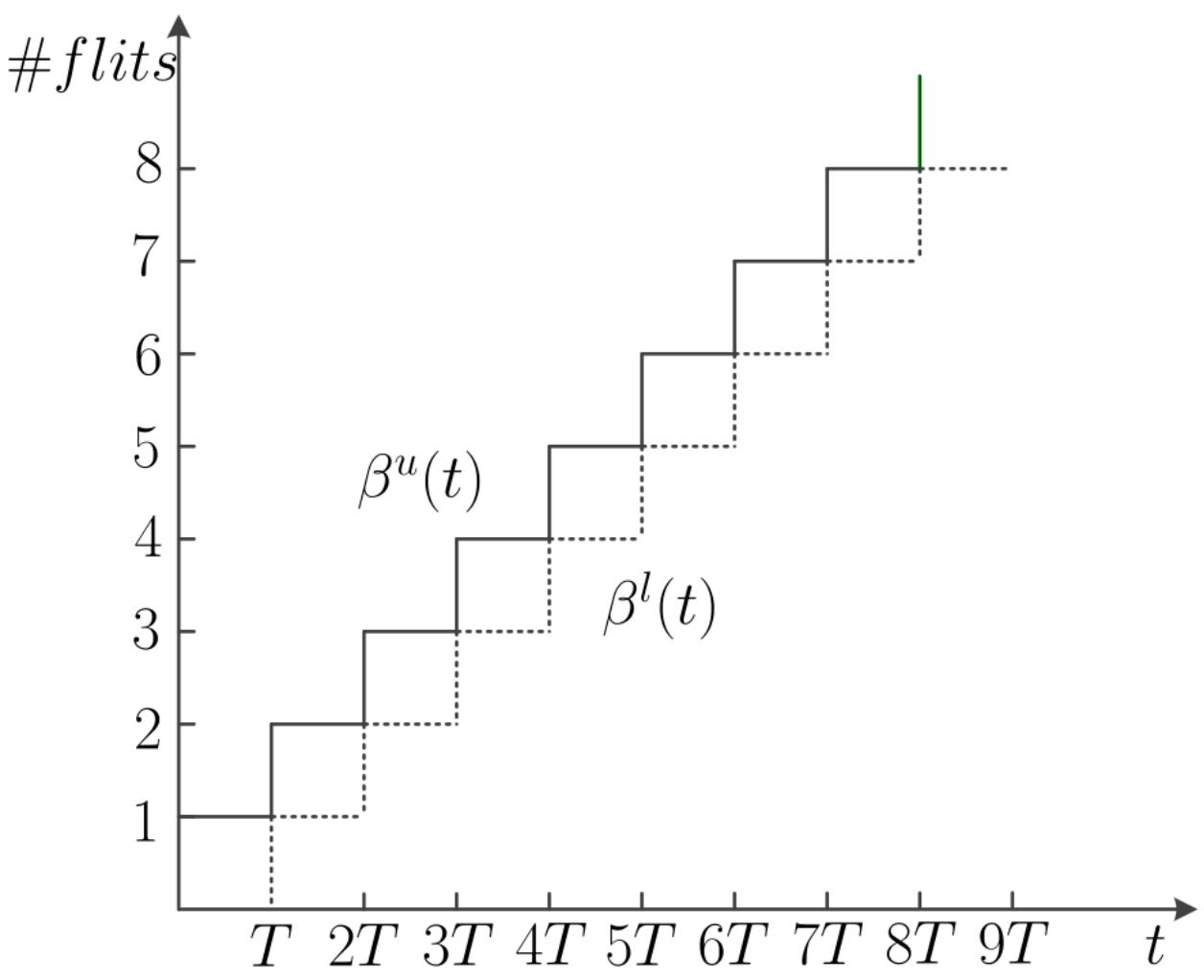
Physical Link



Traffic Flow





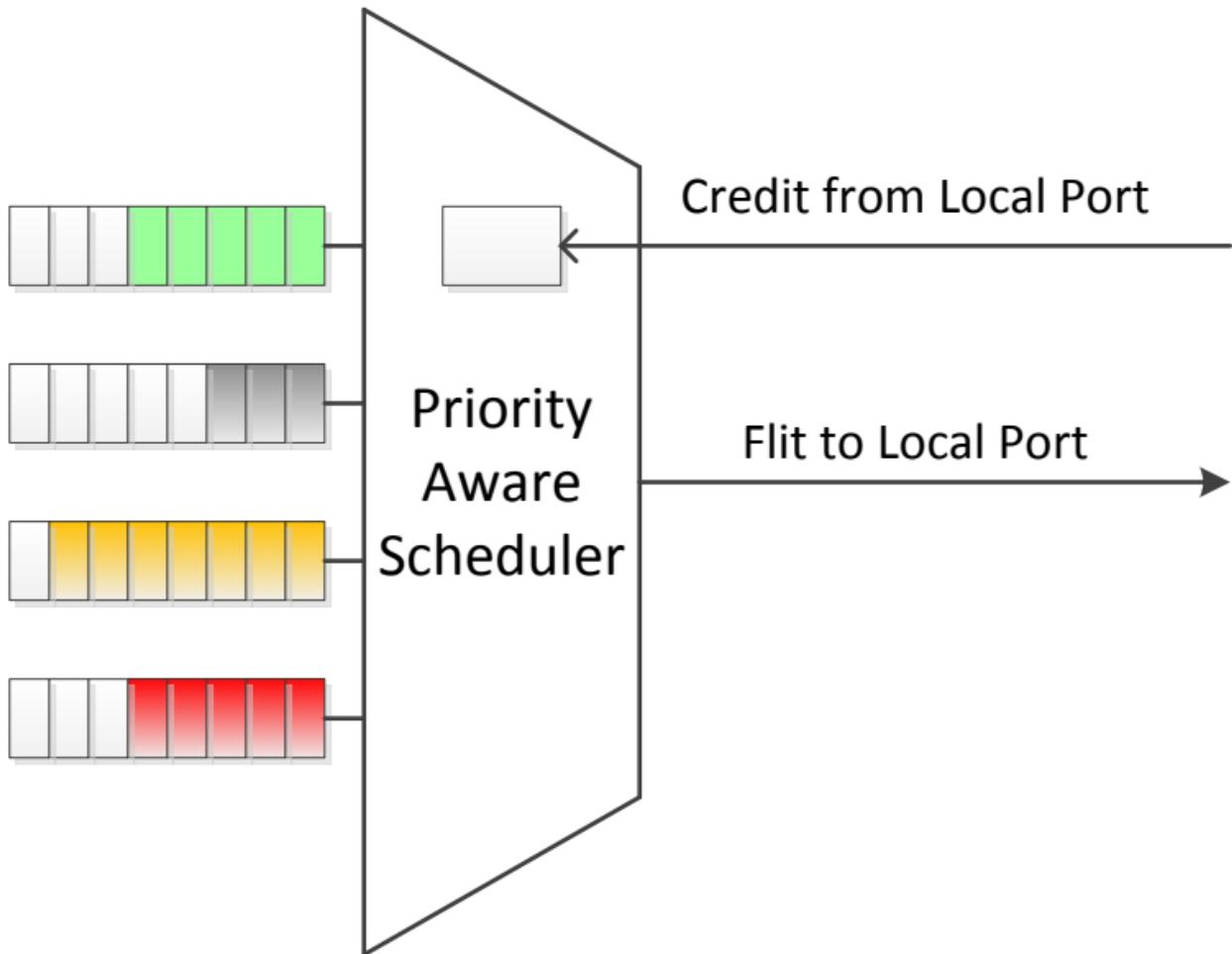


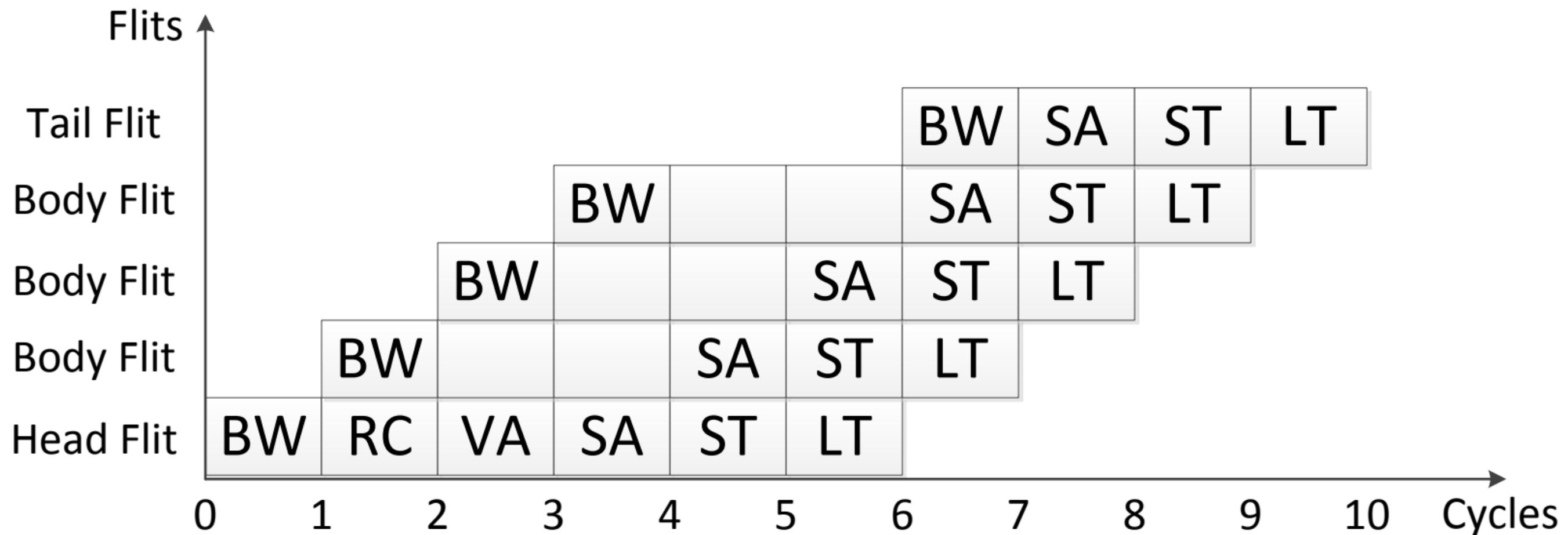
$$f_1 : P_1 = 3$$

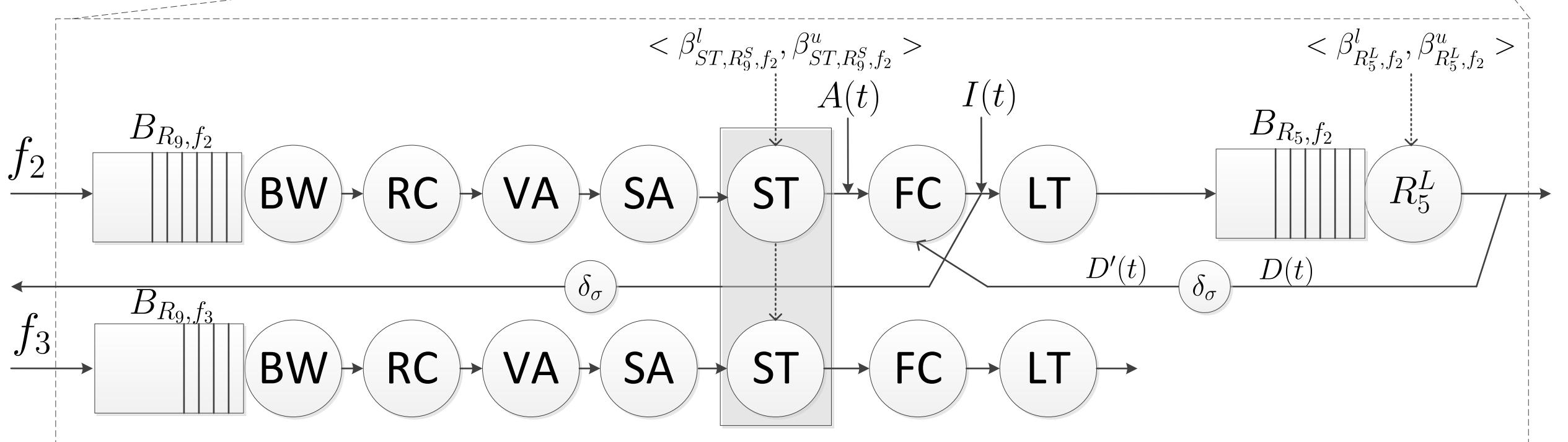
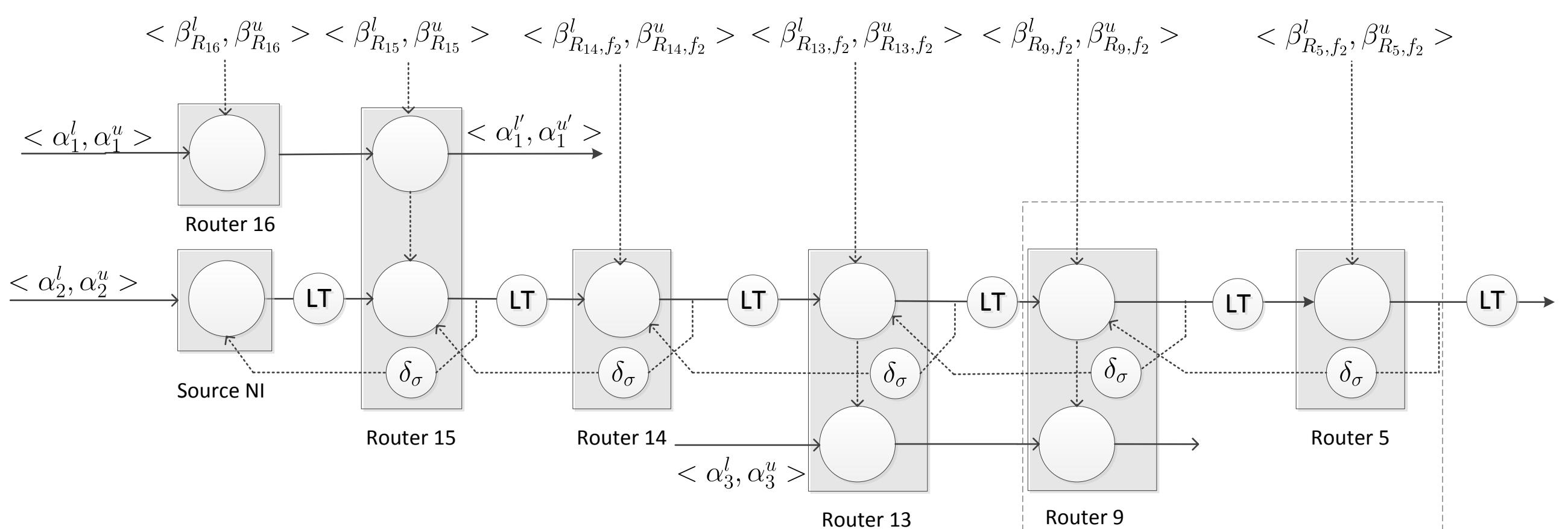
$$f_2 : P_2 = 2$$

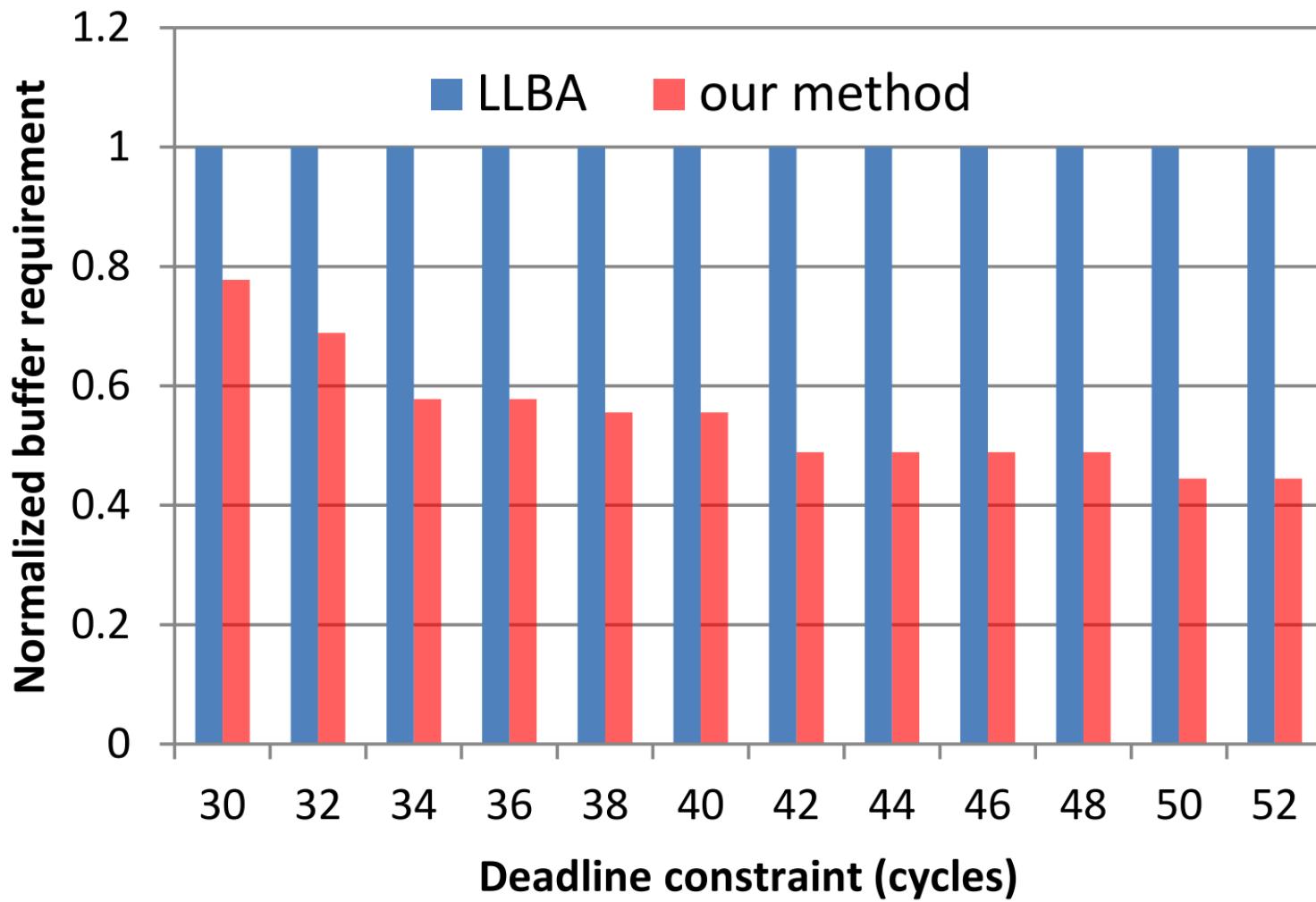
$$f_3 : P_3 = 2$$

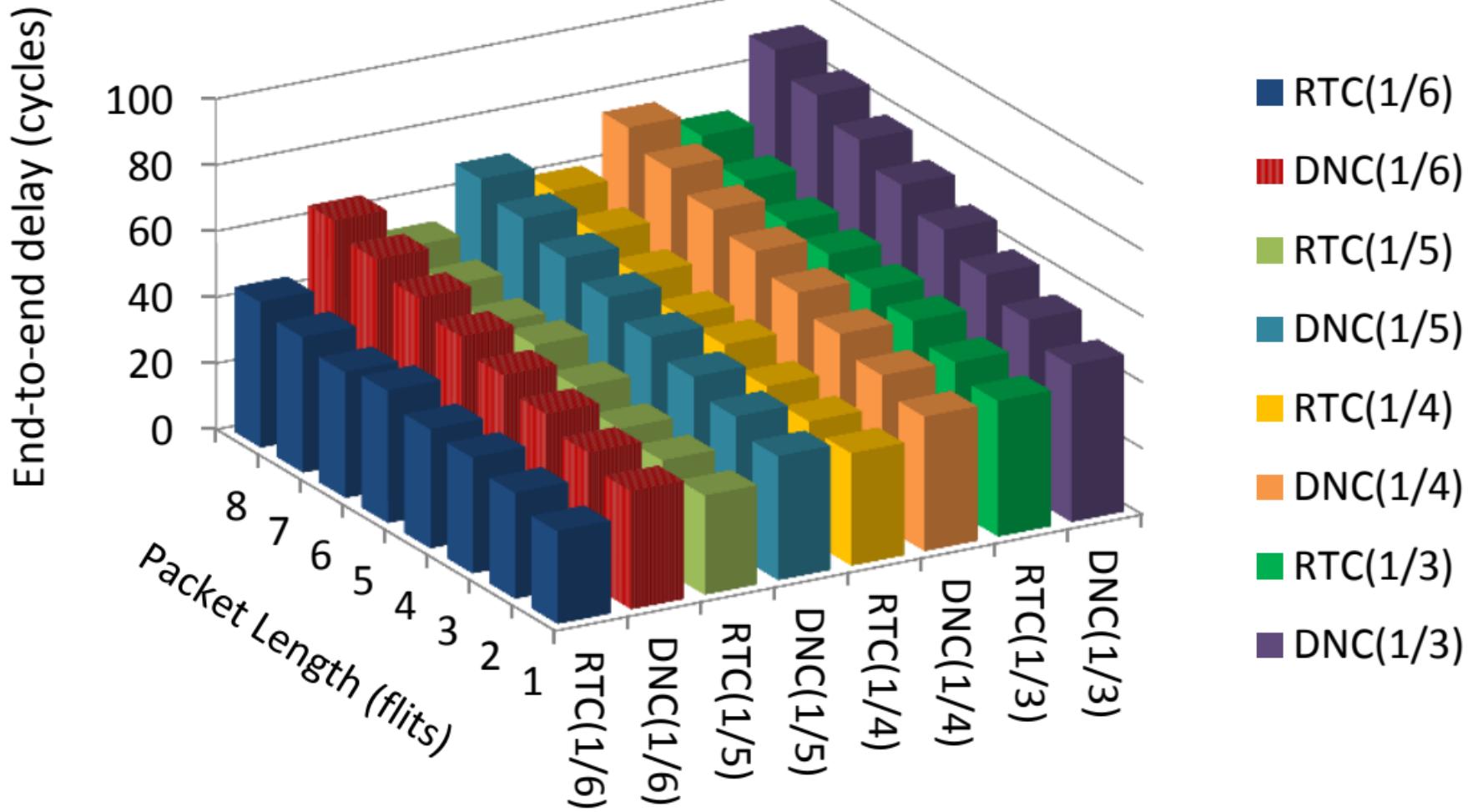
$$f_4 : P_4 = 1$$

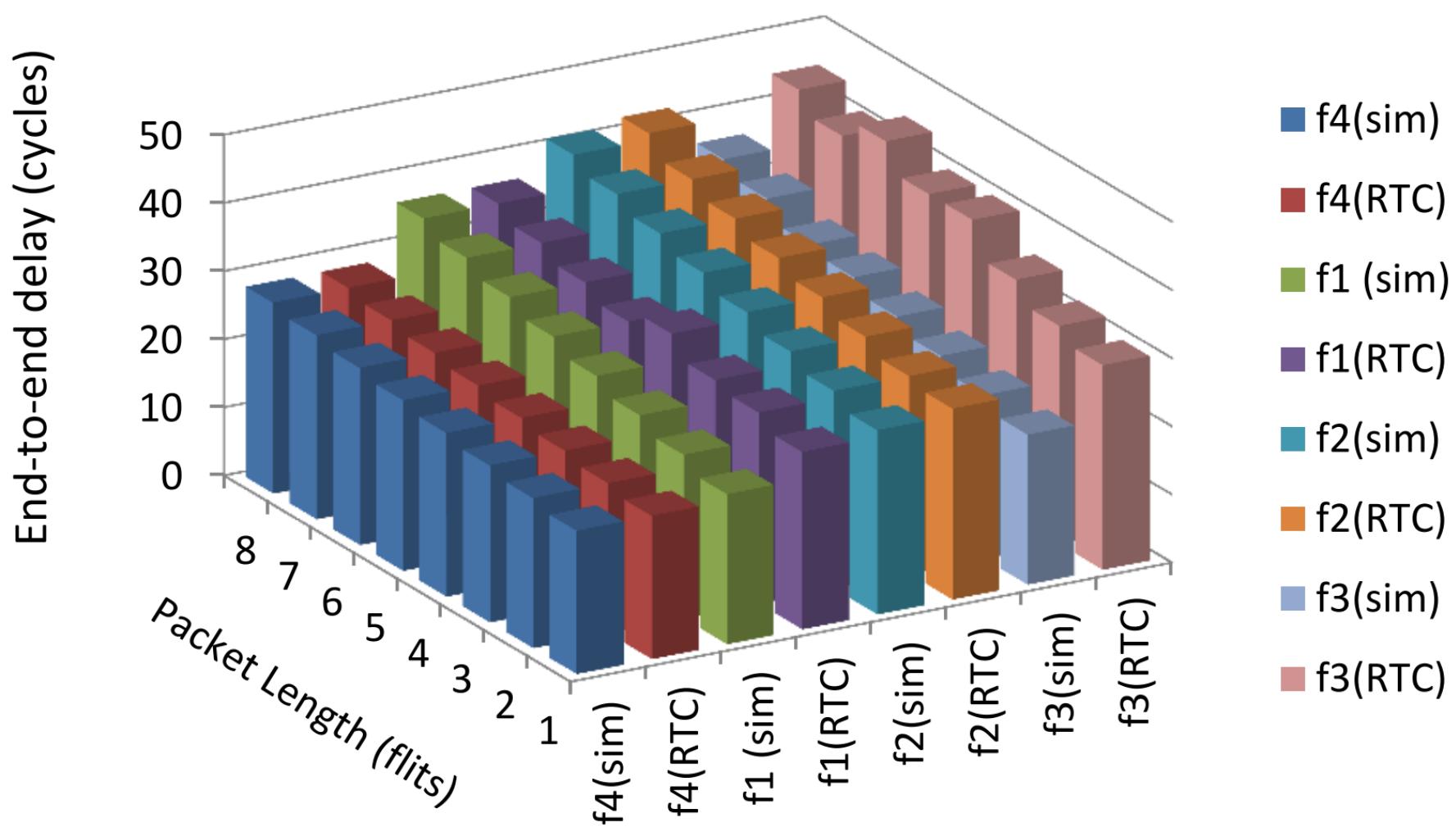


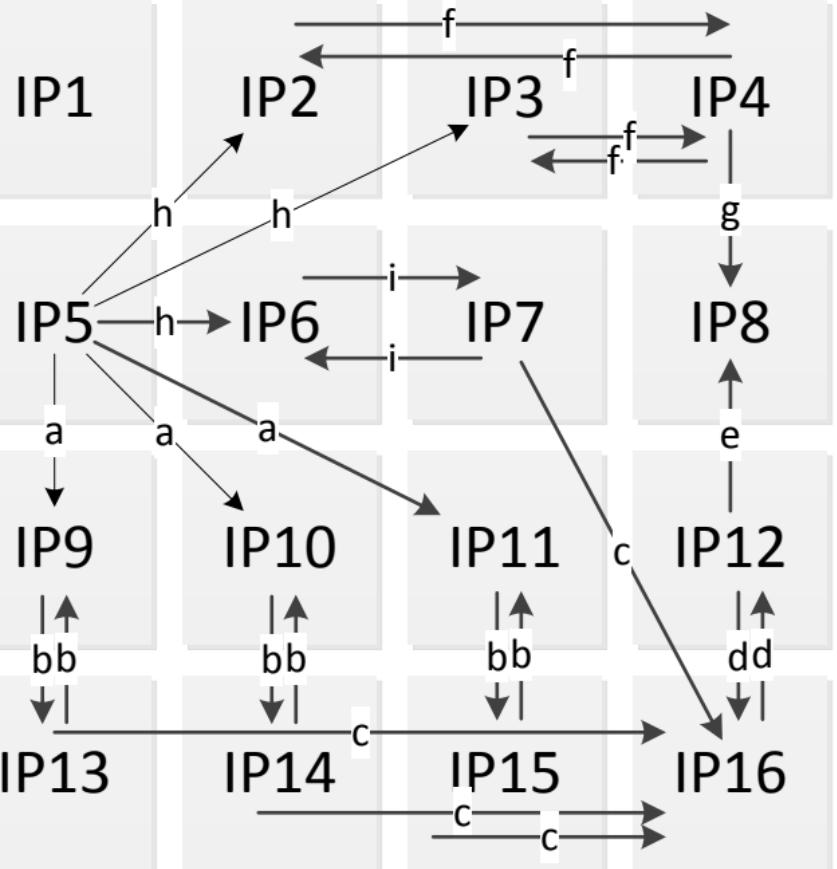








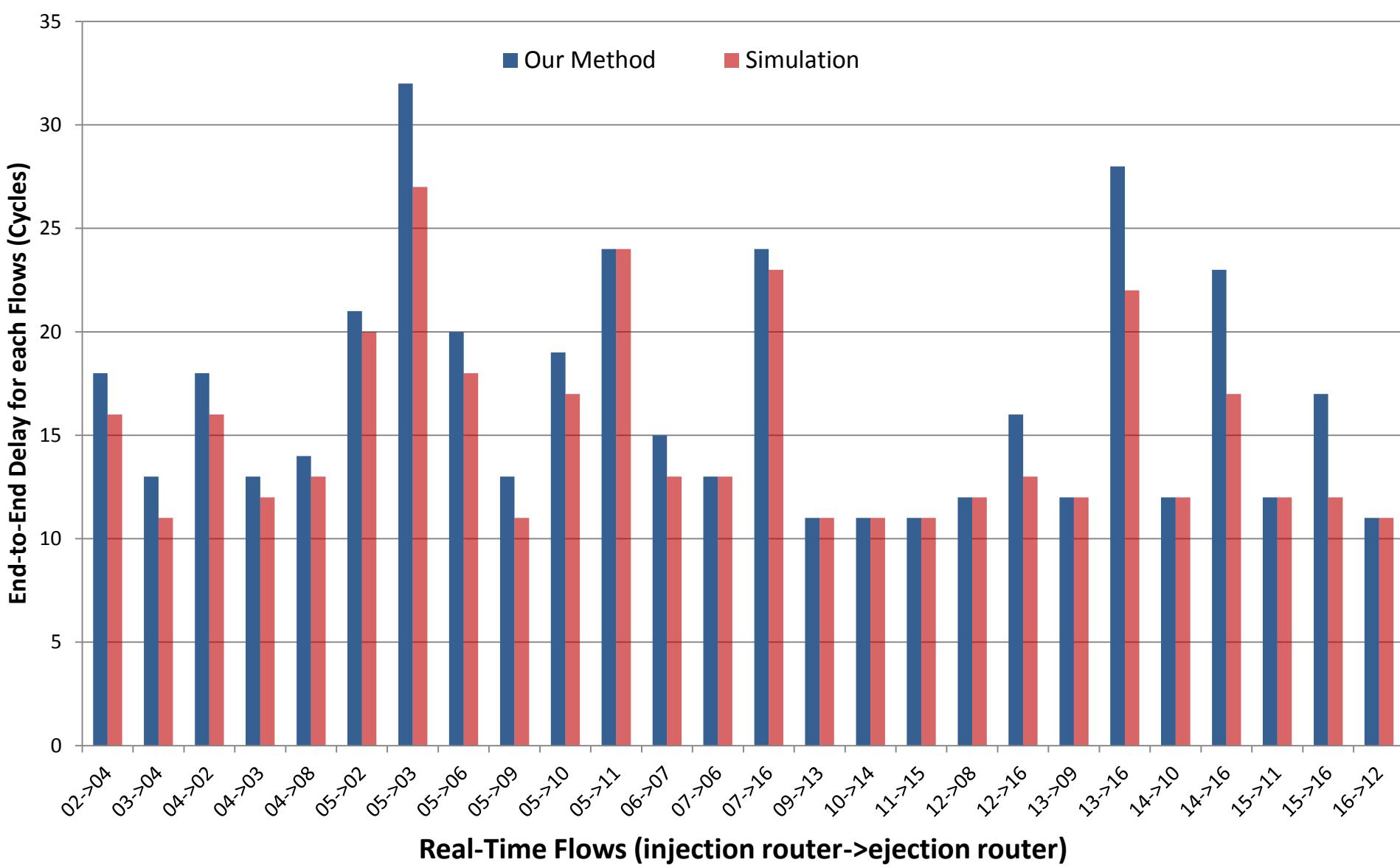




(a)

Group	Priority	Period (cycles)
a	9	16
b	8	125
c	7	125
d	6	32
e	5	125
f	4	500
g	3	256
h	2	16
i	1	125

(b)



## \*Biography of all Authors

[Click here to download Biography of all Authors: biography.txt](#)

Baoliang Li was born in 1987. He received the B.S. degree in Computer Science and Technology from Tsinghua University, P.R. China, in 2009. He is currently working towards the Ph.D. degree from College of Computer at National University of Defense Technology (NUDT), P.R. China. His research interests is performance analysis of computer networks and Networks-on-Chip, with a special interest on the analytical methods of network calculus.

Zeljko Zilic received the B.Eng. degree from the University of Zagreb, Zagreb, Croatia, and the M.Sc. and Ph.D. degrees from the University of Toronto, Toronto, ON, Canada, all in electrical and computer engineering. He joined McGill University, Montreal, QC, Canada, in 1998, where he is currently an Associate Professor with the Department of Electrical and Computer Engineering. He conducts research on various aspects of the design and test of microsystems including programmable logic cores.

Wenhua Dou was born in 1946. He received the B.S. degree in computer science from Harbin Military Engineering College in 1970. He has been working at National University of Defense Technology (NUDT), P.R. China since 1970. He was vice dean of College of Computer, NUDT from 1999 to 2003. He is currently a professor of College of Computer, NUDT with research focusing on computer networks.

**\*Separate Photo**

**Click here to add photo**



**\*Separate Photo**

**[Click here to edit photo](#)**



**\*Separate Photo**

**Click here to d**



**LaTeX Souce Files**

[Click here to download LaTeX Souce Files: manuscript.zip](#)