Extending Real-Time Analysis for Wormhole NoCs

Qin Xiong, Fei Wu, *Member, IEEE*, Zhonghai Lu, *Member, IEEE*, and Changsheng Xie, *Member, IEEE*

Abstract—The delay upper-bound analysis problem is of fundamental importance to real-time applications in Network-on-Chips (NoCs). In the paper, we revisit two state-of-the-art analysis models for real-time communication in wormhole NoCs with priority-based preemptive arbitration and show that the models only support specific router architectures with large buffer sizes. We then propose an extended analysis model to estimate delay upper-bounds for all router architectures and buffer sizes by identifying and analyzing the differences between upstream and downstream indirect interferences according to the relative positions of traffic flows and taking the buffer influence into consideration. Simulated evaluations show that our model supports one more router architecture and applies to small buffer sizes compared to the previous models.

Index Terms—Wormhole NoC, real-time communication, delay

1 Introduction

NoC is an increasingly important architecture of embedded real-time applications [1]. NoC design can be divided into two steps: 1) expressing a system specification as a set or sets of communicating tasks; and 2) mapping these tasks onto the intellectual property (IP) modules of a NoC [2]. With a mapping, a task injects one or more *traffic flows* into the on-chip network and communicates with other tasks through these traffic flows. A traffic flow is a sequence of *packets* with the same *Quality of Service* (QoS) requirements, like deadline.

In a wide range of timing-sensitive systems, especially for safety critical systems, hard *real-time* communication services, which are required by hard real-time traffic flows, should be provided and guaranteed. Compared to *nonreal-time* traffic flows, aside from the requirement of correct logical results, real-time traffic flows also demand timely delivery to guarantee QoS. This is done by ensuring that all packets in a real-time traffic flow can meet the deadline under any conditions. Hence, delays should be estimated analytically instead of empirically and it is important to derive *delay upper-bounds* to ensure that the real-time communication among tasks can meet the time constraints in the design phase. However, the delay of a traffic flow is inherently nondeterministic since packets in the traffic flow may experience various contentions during the transmission, which are caused by resources sharing, e.g., link sharing.

QoS in NoCs can be classified as two classes, *best effort* (BE) classes, which offer no strong commitment and packets may be delayed arbitrarily or even dropped, and *guaranteed service* (GS) classes, which can guarantee a level of performance specified in *service contract*. To NoCs with real-time communication, hard guarantees are preferred and GS is employed in general. The implementation of GS includes two major methods.

- *Resource reservation* reserves resources in space, e.g., Asynchronous Transfer Mode (ATM), or time and space, e.g., time-division multiplexing (TDM) [3], [4], before transmission and can provide strong guarantees on delays. However, in TDM NoCs, the TDM slot utilization is often low [5] due to unallocated and idle slots. In [6], a TDM highway technique is proposed to utilize free slots and enhance the slot utilization.
- *Aggregate resource allocation* arbitrates requests for communication resources. A contention occurs when more than one traffic flow requests access to the same communication resource simultaneously. Priority preemptive arbitration policy is often used in real-time systems as it resolves the *priority inversion* problem [7], [8].

However, the priority preemptive arbitration policy introduces a lot of contentions and makes it more complex to analyze delay upper-bounds. *Wormhole routing* is a popular flow control technique, which has been widely employed in NoCs since it uses a smaller buffer size, provides greater throughput and minimizes communication delays. Few investigations have been performed to analyze real-time communication in wormhole NoCs with the priority preemptive arbitration policy. Two main methods are employed to obtain bounds on delays in real-time systems: *scheduling theory* [9], [10], [11], which often provides tight

- *Q. Xiong is with the Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan 430074, China. E-mail: qinxiong@kth.se.*
- *F. Wu and C. Xie are with the Wuhan National Laboratory for Optoelectronics and the Key Laboratory of Data Storage System, Ministry of Education of China, Huazhong University of Science and Technology, Wuhan 430074, China. E-mail: {wufei, cs_xie}@hust.edu.cn.*
- *Z. Lu is with the School of Information and Communication Technology, KTH Royal Institute of Technology, Stockholm 16440, Sweden. E-mail: zhonghai@kth.se.*

Manuscript received 19 Apr. 2016; revised 22 Feb. 2017; accepted 6 Mar. 2017. Date of publication 21 Mar. 2017; date of current version 15 Aug. 2017. Recommended for acceptance by A. Yakovlev.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TC.2017.2686391

bounds at the cost of high computational complexity, and *network calculus* [12], [13], which is well suited to large systems, but computes pessimistic results [14]. Employing the real-time scheduling theory, Shi and Burns presented a coarse-grained flow-based analysis model called *SB's model*, which can calculate the delay upper-bounds of real-time traffic flows efficiently in priority-based wormhole NoCs [9]. However, this model can only be used to estimate delay upper-bounds for NoCs with specific router architectures and large buffer sizes since it does not consider the relative positions of flows and buffering impact on delay propagation. Kashif et al. proposed a fine-grained stage-based analysis model called *stage-level analysis* (SLA) model to enhance tightness in the real-time analysis, but the restriction on its application scope, caused by ignoring the buffering impact on delay propagation, also exists in this model [10]. Both the SB's model and the SLA model estimate the delay upper-bounds under the condition that the routers have large buffer sizes. Intending to lift this limitation, Kashif and Patel extended the SLA model with supporting small buffer sizes, which is called *SLA+ model* [11]. However, although the SLA+ model takes back-pressure into consideration, it still can only be capable of estimations for NoCs with specific router architectures and large buffer sizes since *multi-point progressive block (MPB)* (descried in detail in Section 4) is not captured. In the paper, we propose an extended real-time analysis model based on the SB's model for wormhole NoCs with priority-based preemptive arbitration by taking relative flow positions and the buffer influence (including MPB) into consideration.

By providing correct delay upper-bounds for NoCs composed of all kinds of routers with small or large buffer sizes, we make the following specific contributions in this paper.

1) We revisit the SB's model in [9] and the SLA+ model in [11], and then point out the limitations of their application scopes.
2) We illustratively expound the MPB and extend the model in [9] by considering relative flow positions and router buffer impact.
3) A case study to compare extended model to the SB's model and the SLA+ model, and show that the application scopes of the SB's model and the SLA+ model are restricted and our mode supports one more router architecture and small buffer sizes.

The rest of the paper is organized as follows. Section 2 discusses the related work. In Section 3, the delay bound analysis problem is illustratively given, and the SB's model and the SLA+ model are revisited. In Section 4, we propose an extended model, which widens its application scope. In Section 5, we conduct a detailed case to show quantitative results comparing the SB's model, the SLA+ model to ours in terms of application limitations. Finally, we conclude in Section 6.

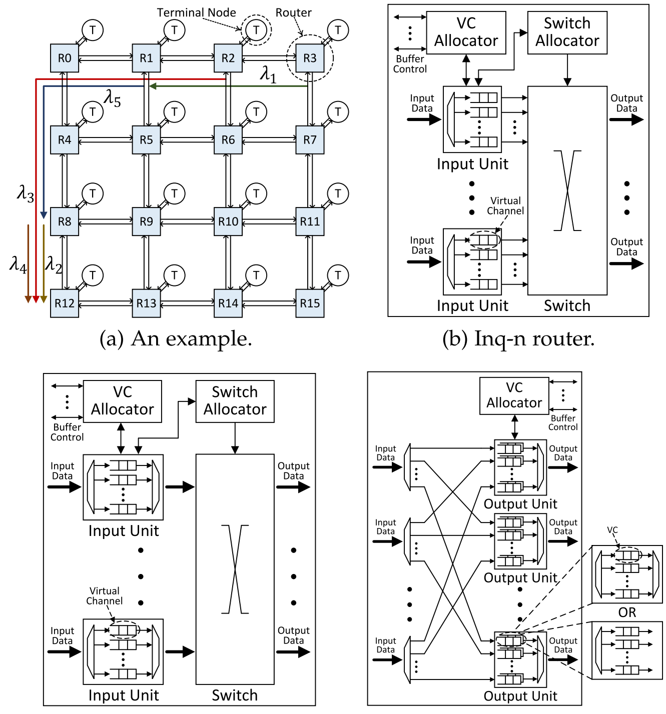
2 Related Work

In wormhole NoCs, the real-time communication analysis has attracted research attentions recently. The real-time analysis for wormhole NoCs builds upon previous investigations of the real-time analysis for off-chip interconnection networks.

One method is network calculus, which has been proposed for Internet QoS [15] and employed in many other fields such as switched Ethernet networks, sensor networks and on-chip networks [16], etc. Network calculus is a set of rules and results based on $(min, +)$ algebra and deals with queuing problems encountered in packet-switched networks. Network calculus contains bounding constraints on packet arrival and service, and derives the bounds of work backlogs and the packet delay, which can be used to quantify the real-time delay upper-bound immediately. In [13], the classic network calculus was extended as real-time calculus (RTC) to analyze hard real-time embedded systems. Two key features of RTC are: 1) it uses a count-based abstraction, which provides upper and lower bounds of the possible number of traffic flows; and 2) it models traffic flows more general than the classical models like sporadic and periodic (with or without jitter) models, etc [17]. However, RTC usually yields estimated pessimistic delay upper-bounds since it does not take the pipelined behavior of networks into consideration. The interference caused by one traffic flow may be counted multiple times [10]. Recently, network calculus is incorporated into the delay bound analysis methodology, which adopts a formalized router model based on Intel's eXtensible Micro-Architectural Specification (xMAS) as an intermediate model between an ad hoc router model and its network calculus analysis model [18].

As another approach for analyzing delay bounds for on-chip networks, the real-time scheduling theory has been borrowed from the uniprocessor scheduling area to address the problem. The foundational investigation in fixed-priority real-time scheduling theory and the basic model of uniprocessor scheduling was proposed in [19]. But this model cannot be used in wormhole network directly since there is only one scheduled resource in this model, while in wormhole network, every link and every *virtual channel* (VC) is a scheduled resource, which makes the contention scenario more complex. Since NoCs need more precise results, based on the earlier methods and results, the contention representing the source of analyzing difficulty has been analyzed more precisely. In [2], Lu et al. formulated a contention tree to present contentions among real-time traffic flows. However, no formula of delay upper-bounds is derived. In [9], an approach raised by Shi and Burns can estimate the worst-case delay based on two quantifiable different interferences: direct interference and indirect interference. This approach treats links on a traffic flow's routing path as one resource in contention classifications and calculations. The formulas of delay upper-bounds are derived based on the works of processor scheduling theory [20], [21]. In a series of publications [22], [23], [24], [25], the application scope of this method was extended by Shi et al. In [26], Burns et al. introduced this analysis model into the mixed criticality system and derived formulas for scheduling analysis of mixed criticality tasks. The details about their analysis are introduced in Section 3.3. Compared to the SB's model, Kashif et al. [27] proposed a *link-level analysis* (LLA) model for real-time communication in NoCs where contention classifications and calculations are processed link by link along the routing path. In this way, the pipelining effect is considered and delay upper-bounds can be tighter. In [28], Kashif and Patel noticed that chain-blocking, which is caused by back-pressure and was ignored by the previous works, would affect delays. In the work, the minimum buffer space requirements are computed to prevent back-pressure in order to validate the SB's model and the LLA model. The SLA model [10] was proposed for producing tighter delay upper-bounds under a key assumption that each VC has enough buffer space to eliminate back-pressure. The SLA model analyzes a traffic flow, which uses pipelined communication resources, stage by stage. Interference suffered on each stage is calculated based on the interference suffered on the previous stage and the contention scenario on the current stage. The SLA model provides tighter delay bounds than the SB's model and is more generally applicable than the LLA model. Nevertheless, due to cost restrictions in NoCs, buffer space of each VC in wormhole NoCs is usually not large enough to eliminate back-pressure. Among the SB's model, the LLA model or the SLA models, none of them are designed for delay upper-bound estimations when buffer sizes are not large enough. An improvement to the SLA model was presented by Kashif and Patel in [11] by relaxing the assumption of infinite buffer sizes. In order to make their new model (the SLA+ model) valid when buffer sizes are small, they account for blocking due to back-pressure. The SLA+ model separates interferences into two types: the interference caused by higher priority traffic flows on a stage and the interference on a stage due to limited buffer space at the downstream router of that stage. The first type of interference is caused by the contention from higher priority traffic flows, and the second type, which is called *blockage*, results from back-pressure. However, aside from back-pressure, the MPB caused by buffers is not captured by the previous works, which leads to the limitations that these models cannot estimate an upper-bound on delay when buffer sizes are small but larger than the minimum[1] and in some situations, even buffer sizes are large enough. The detailed description is in Section 4. Based on the observation in [29], Indrusiak et al. [30] proposed a new tighter model by counting buffer sizes quantitatively. However, since the MPB was not comprehensively revealed in [29], the SLA model was incorrectly marked as "safe" in [30], which means that the SLA model can be applied if buffer sizes are large enough. Actually, as described above, the MPB may exist when buffer sizes are large enough and the SLA model cannot be employed under these situations. In this paper, we focus on comprehensively revealing the MPB and conducting a deep understanding of it.

Compared to the LLA model, the SLA model and the SLA+ model, which get tighter delay bounds at the cost of analyzing and computational complexity, the SB's model is a good trade-off between complexity and tightness. To our knowledge, Shi and Burns' work [9] represents one of the state-of-the-arts for the real-time communication delay bound analysis of on-chip wormhole networks. In order to address these problems, especially the limitations on applicability of the previous models, we propose a new model based on the SB's model by taking the relative positions of flows and the buffer influence into consideration to 1) ensure the correctness of delay upper-bounds; and 2) support one more router architecture and small buffer sizes.



(a) An example.

(b) Inq-n router.

(c) Inq-1 router.

(d) Outq router.

Fig. 1. (a) A 4×4 NoC example with five traffic flows. (b) and (c) Input queuing router with an switch input speedup of n (n is the number of VCs per input unit) and 1, respectively. (d) Output queuing router.

This paper is an extended version of [29]. The extensions widen and deepen the content in [29], making new contributions: 1) this paper deals with both the SB's model and the SLA+ model while [29] only considers the SB's model; 2) considering the impact of different router architectures on the analysis; 3) the impact of buffer sizes is qualitatively considered; and 4) extending the case study.

3 The Problem and Previous Analysis

3.1 Problem Illustration

Our real-time analysis problem is to analyze the delay upper-bounds of real-time flows in wormhole NoCs. Fig. 1a illustrates a 2D 4×4 mesh NoC example with five real-time flows. A node in the NoC is either a terminal node (T, round node) or a router (R, rectangular node), and connected with other nodes by bidirectional links. A terminal node processes computational tasks and communicates with other terminal nodes through the network. The packet-switching router performs wormhole routing, which means pipelined packet delivery with VCs allocated at the packet level and the physical channels at the *flit* level (flit is the flow control unit and each packet is divided into a quantity of fixed size flits). The dimension-order X-Y routing is employed for simplicity and deadlock freedom. A packet is routed first in the X (horizontal) dimension until it arrives at the same column as its destination and then in the Y (vertical) dimension to the destination. For example, λ_5 starts in R_1 and is transmitted to R_0 in the X dimension, then passes through R_4 and terminates in R_8 along the Y dimension.

In wormhole NoCs with the same topology and task mapping, contention scenarios may be various since different router architectures are employed, which means router

[1] The MPB does not exists when buffer can only store pipelined flit (s), which is the minimum buffer size. Hence, the small buffer size in this paper means that the buffer size is larger than the minimum value. Our model supports the minimum buffer size.

architectures may affect the delay of a traffic flow. A wormhole router can be classified as an *input queuing router* or an *output queuing router* based on its buffering strategy, which determines buffers' location in the router [5]. *Switch input speedup* of input queuing routers is the ratio of the switch bandwidth provided to the minimal switch bandwidth needed to support full throughput on all input ports. Figs. 1b, 1c, and 1d show three common router architectures in wormhole NoCs. The first two show two kinds of input queuing routers while the last one uses the output buffering strategy. For convenience, we call these three router architectures in Figs. 1b, 1c, and 1d as *Inq-n*, *Inq-1* and *Outq*, respectively.

- *Inq-n*. Flits are stored at the input of the router. Each input unit is connected to the switch by as many links as VCs per input unit, shown in Fig. 1b. In this type of router, all packets from different VCs in an input unit can pass through these internal links parallelly.
- *Inq-1*. As in Inq-n router, input units store flits. In Fig. 1c, all VCs in an input unit are multiplexed on one internal link to connect to the switch. In this situation, if two flits from different VCs in an input unit request different output ports simultaneously, only one flit can be forwarded at a time and this means a contention occurs.
- *Outq*. In an output queuing router, flit queues are located at the output of the router as shown in Fig. 1d. There are a number of buffers associated with each output port and each of these buffers contains multiple VCs. A flit is directly forwarded to the requested output unit and stored in a VC. Since all flits stored in a buffer request the same output port, only one flit gets allocated at a time and the contention scenarios are the same for both Outq routers with and without a multiplexer in each buffer, as shown in Fig. 1d. Therefore, we regard these two architectures as one in this paper.

Hence, towards Inq-n and Outq routers, only external links can be contended among traffic flows. For Inq-1 routers, not only external links, but also internal links can cause contentions.

3.2 Symbols, Assumptions and System Model

We globally use the symbols in Table 1 throughout the paper. The symbols for a specific definition, formula or proof are defined within their own section.

We make the following basic assumptions:

1) We consider real-time communication in wormhole NoCs with VCs and deterministic routing. The priority-based preemptive scheduling is used for link (internal and external) arbitrations.
2) A flow λ_i is a stream of periodic or sporadic packets and $D_i \leq (T_i - J_i)$. Each flow λ_i has a unique priority, with a lower index indicating a higher priority.
3) The router uses per-flow buffering, meaning that the number of VCs in each input unit (for an input queuing router) or in each output unit for each input port (for an output queuing router) is no less than the number of flow priorities such that each VC is exclusively used by one flow.

TABLE 1
Symbols and Explanations

Symbol	Explanation
Λ	The set of real-time traffic flows in a NoC
λ_i	A real-time traffic flow
C_i	The maximum non-contentional delay of λ_i
T_i	The period for periodic traffic flow or the minimum release interval of sporadic traffic flow, λ_i
D_i	The end-to-end delay constraint of λ_i
J_i	The release jitter of λ_i
Z_i	The number of flits of a packet of λ_i
H_i	The hop count of λ_i
ℓ_i^n	The nth unidirectional link along the routing path of λ_i
$\mathscr{L}_i$	The set of unidirectional links along the routing path of λ_i
S_{D_i}	The set of traffic flows, which directly contend with λ_i
S_{I_i}	The set of traffic flows, which indirectly contend with λ_i
$S_{I_i}^{U_j}$	The set of traffic flows, which indirectly contend with λ_i through $\lambda_j \in S_{D_i}$ at upstream position
$S_{I_i}^{D_j}$	The set of traffic flows, which indirectly contend with λ_i through $\lambda_j \in S_{D_i}$ at downstream position
R_i	The delay upper-bound of λ_i
I_i	The maximum interference (in time) suffered by λ_i
I_{ji}	The maximum interference (in time) suffered by λ_i from λ_j
IP_{ji}	The maximum interference (in time) suffered by λ_i from a packet of λ_j
IT_{ji}	The maximum number of packets from λ_j that can interfere with one packet of λ_i
I_{ji}^D	The maximum interference (in time) suffered by λ_j from flows in $S_{I_i}^{D_j}$
B	Buffer size per VC

A wormhole real-time network includes n real-time traffic flows $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$. Each real-time flow can be characterized by four parameters $\lambda_i = \{C_i, T_i, D_i, J_i\}$. Every packet of λ_i should be delivered within D_i, and J_i is the upper-bound of deviation between the minimum release interval of successive packets and T_i. By definition, for an *observed* flow λ_i, the delay upper-bound R_i of a packet is

$$R_i = C_i + I_i. \tag{1}$$

If $R_i \leq D_i$, λ_i can meet its deadline under any scenario. For simplicity, time unit is not explicitly given since the same time base is used throughout this paper.

3.3 Previous Analysis Models

Applying the real-time scheduling theory, Shi and Burns [9] and Kashif and Patel [11] presented analysis methods for the problem, respectively. Next, we summarize and discuss the application limitations of their models.

3.3.1 The SB's Model

The basic assumptions of this model are the same as ours. This model is only applicable to Inq-n/Outq routers with large buffer sizes (no back-pressure).

Direct and Indirect Interferences. The analysis quantifies two distinct interferences from higher priority flows: *direct interference* and *indirect interference*. If flow λ_j has a higher priority than λ_i and shares at least one link with λ_i, then λ_j has a direct interference relation with λ_i. If flow λ_j has no shared link with λ_i and directly contends with an intermediate flow, which has a direct interference relation with λ_i, then λ_j has an indirect interference relation with λ_i. The direct and the indirect interference flow sets S_{D_i} and S_{I_i} of λ_i are defined as

$$S_{D_i} = \{\lambda_j | \mathscr{L}_i \cap \mathscr{L}_j \neq \phi, i > j, \forall \lambda_j \in \Lambda\},$$
$$S_{I_i} = \{\lambda_j | \mathscr{L}_i \cap \mathscr{L}_j = \phi, \forall \lambda_k \in S_{D_i}, \forall \lambda_j \in S_{D_k}\}.$$

Analysis of Direct Interference. The maximum direct interference suffered by a packet from λ_i when indirect interference is not considered is

$$I_i = \sum_{\forall \lambda_j \in S_{D_i}} \left\lceil \frac{R_i^{SB} + J_j}{T_j} \right\rceil C_j. \tag{2}$$

Analysis of Indirect Interference. If S_{I_i} is non-empty, a flow in S_{I_i} may cause interference to the observed flow λ_i through an intermediate flow in S_{D_i}. When considering the indirect interference, the upper-bound of interference from higher priority flows to a packet of λ_i is

$$I_i = \sum_{\forall \lambda_j \in S_{D_i}} \left\lceil \frac{R_i^{SB} + J_j + J_j^I}{T_j} \right\rceil C_j, \tag{3}$$

where J_j^I means the deviation of λ_j between the release interval of consecutive packets and T_j, which is caused by flows in S_{D_j}, and is called *interference jitter*. The upper-bound of the interference jitter is

$$J_j^I = I_j = R_j^{SB} - C_j.$$

Delay Bound Model. Using Eq. (3) to substitute I_i in Eq. (1), the formula of the delay upper-bound is

$$R_i^{SB} = \sum_{\forall \lambda_j \in S_{D_i}} \left\lceil \frac{R_i^{SB} + J_j + J_j^I}{T_j} \right\rceil C_j + C_i. \tag{4}$$

Note that, in this formula, J_j^I exists and is equal to $R_j^{SB} - C_j$ if and only if λ_j is an intermediate flow between λ_i and traffic flows in S_{I_i}, i.e., $S_{D_j} \cap S_{I_i} \neq \phi$. If $S_{D_j} \cap S_{I_i} = \phi$, on the contrary, J_j^I does not exist or can be regarded as 0.

3.3.2 The SLA+ Model

The basic assumptions of this model are the same as ours except $D_i \leq (T_i - J_i)$. Since the situation when $D_i > (T_i - J_i)$ is beyond the scope of this paper and this model supports $D_i \leq (T_i - J_i)$, here we summarize the SLA+ model when $D_i \leq (T_i - J_i)$. This model intends to support small buffer sizes (with back-pressure). However, it can only apply to Inq-n/Out-q routers with large buffer size (no back-pressure).

Direct and Indirect Interferences. The SLA+ model analyzes a delay bound of a flow along the stages on its routing path. A path δ_i of λ_i is a sequence of stages $(s_{i,1}, s_{i,2}, \ldots, s_{i,|\delta_i|})$, where $|\delta_i|$ denotes the number of stages on δ_i. A subpath $\sigma_i(s_{i,l})$ of λ_i is a sequence of stages $(s_{i,1}, s_{i,2}, \ldots, s_{i,l})$. The direct and the indirect interference flow sets on a stage s, $\underset{s}{S}_{D_i}$ and $\underset{s}{S}_{I_i}$, are defined as

$$\underset{s}{S}_{D_i} = \{\lambda_j | s \in \delta_i \cap \delta_j, i > j, \forall \lambda_j \in \Lambda\},$$

$$\underset{s}{S}_{I_i} = \{\lambda_j | s \neq \hat{s}, \hat{s} \notin \delta_i, \forall \lambda_k \in \underset{s}{S}_{D_i}, \forall \lambda_j \in \underset{\hat{s}}{S}_{D_k}, \forall \hat{s} \in \sigma_k(s)\}.$$

Analysis of Direct Interference. The term $\underset{s}{\omega}_i$ indicates the delay upper-bound of λ_i on a stage s and is given by

$$\underset{s}{\omega}_i = \underset{s}{I}_i + \underset{s}{IB}_i(\underset{s}{\omega}_i) + L_i, \tag{5}$$

where $\underset{s}{I}_i$ means the maximum interference suffered by λ_i from higher priority traffic flows on stage s, $\underset{s}{IB}_i$ represents the blockage suffered by λ_i on stage s, and $\overset{s}{L}_j$ denotes the basic stage delay, which is the maximum delay of a packet on one stage when it does not suffer any interference on that stage. The maximum direct interference suffered by a packet from λ_i on stage $s \in \delta_i$ when indirect interference is not considered is

$$\underset{s}{I}_i = \underset{s^-}{I}_i + \sum_{\forall \lambda_j \in \underset{s}{S}_{D_i}} \left\lceil \frac{\underset{s}{\omega}_i + J_j}{T_j} \right\rceil L_j - \sum_{\forall \lambda_j \in \underset{s}{S}_{D_i} \cap \underset{s^-}{S}_{D_i}} \left\lceil \frac{\underset{s^-}{\omega}_i + J_j}{T_j} \right\rceil L_j, \tag{6}$$

where s^- denotes the stage preceding s on δ_i.

Analysis of Indirect Interference. The terms $\underset{s}{\omega}_j(\lambda_i)$ and $\underset{s}{I}_j(\lambda_i)$ denote the delay upper-bound and the maximum interference of λ_j, respectively, only due to traffic flows in $\underset{s}{S}_{D_j} \cap \underset{s}{S}_{I_i}$. $\underset{s}{J}{}_j^I$ is the interference jitter on stage s and is given by

$$\underset{s}{J}{}_j^I = \underset{s}{\omega}_j(\lambda_i) - L_j,$$

where

$$\underset{s}{\omega}_j(\lambda_i) = \underset{s}{I}_j(\lambda_i) + L_j,$$

$$\underset{s}{I}_j(\lambda_i) = \underset{s^-}{I}_j(\lambda_i) + \sum_{\forall \lambda_k \in \underset{s}{S}_{D_j} \cap \underset{s}{S}_{I_i}} \left\lceil \frac{\underset{s}{\omega}_j(\lambda_i) + J_k + \underset{s}{J}{}_k^I}{T_k} \right\rceil L_k - \sum_{\forall \lambda_k \in \underset{s}{S}_{D_j} \cap \underset{s^-}{S}_{D_j} \cap \underset{s}{S}_{I_i}} \left\lceil \frac{\underset{s^-}{\omega}_i(\lambda_i) + J_k + \underset{s}{J}{}_k^I}{T_k} \right\rceil L_k.$$

The maximum interference to λ_i suffering indirect interference on stage $s \in \delta_i$ is

$$\underset{s}{I}_i = \underset{s^-}{I}_i + \sum_{\forall \lambda_j \in \underset{s}{S}_{D_i}} \left\lceil \frac{\underset{s}{\omega}_i + J_j + \underset{s}{J}{}_j^I}{T_j} \right\rceil L_j - \sum_{\forall \lambda_j \in \underset{s}{S}_{D_i} \cap \underset{s^-}{S}_{D_i}} \left\lceil \frac{\underset{s^-}{\omega}_i + J_j + \underset{s}{J}{}_j^I}{T_j} \right\rceil L_j. \tag{7}$$

Analysis of Blockage. If the buffer size is large enough, no back-pressure exists and the blockage is 0. If the buffer size is small, λ_i may be blocked from accessing stage s since the corresponding buffer in the downstream router is full. The blockage is given by

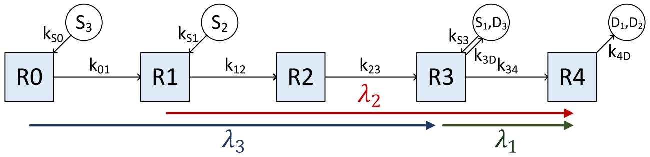


Fig. 2. An example showing the limitations of application scopes of the SB's model and the SLA+ model.

$$\underset{s}{IB}_{i}(\underset{s}{\omega}_{i}) = \begin{cases} 0, & \text{if } B \geq L_i \\ max(0, \sum_{\forall \lambda_j \in \underset{s^+}{S}_{D_i} \setminus \underset{s}{S}_{D_i}} \left\lceil \frac{\underset{s}{\omega}_{i} + J_j + \underset{s}{J}^{I}_{j}}{T_j} \right\rceil L_j & \\ \quad + \underset{s^+}{IB}_{i}(\underset{s}{\omega}_{i}) - B + CF + 1), & \text{if } B < L_i, \end{cases} \quad (8)$$

where s^+ denotes the stage succeeding s on δ_i, and CF is the the credit feedback delay. The buffer size satisfies $B \geq CF + 1$.

Delay bound model. The delay upper-bound of λ_i is

$$R_i^{SLA+} = \underset{s}{\omega}_{i} + |\delta_1| - 1, \quad (9)$$

where $\underset{s}{\omega}_{i}$ is calculated by Eqs. (5), (6), (7), and (8).

3.3.3 The Application Limitation

We use a simple example to show the application limitations of the SB's model and the SLA+ model. In other words, the SB's model and the SLA+ model cannot estimate delay upper-bounds if the employed routers in a NoC are not Inq-n with large enough buffer sizes.

Example. *Assume that there are three flows in a wormhole NoC formed by single-cycle*[2] *Inq-n routers whose VC buffer size is 10, as shown in Fig. 2. Parameters of the flows:* $\lambda_1 = \{21, 100, 100, 0\}$, $\lambda_2 = \{24, 100, 100, 0\}$, $\lambda_3 = \{14, 100, 40, 0\}$ *(*$\lambda_i = \{C_i, T_i, D_i, J_i\}$*).* λ_1*,* λ_2 *and* λ_3 *are injected from source node at time 3, 1 and 0, respectively.*

The hop-counts of λ_1*,* λ_2 *and* λ_3 *are 3, 5 and 5, respectively, and the flit numbers of these three flows are 19, 20 and 10, respectively. The timing of inter-router link utilizations is described as below and shown in Fig. 3.*

Timing Description.

At t=1, λ_3 *arrives at* R_0*;*

At t=1∼2, λ_3 *utilizes* k_{01}*;*

At t=2, λ_2 *and* λ_3 *arrive at* R_1*;*

At t=2∼3, λ_2 *wins* λ_3 *in the arbitration of* k_{12} *and utilizes* k_{12}*,* λ_3 *utilizes* k_{01}*;*

At t=3, λ_2 *arrives at* R_2 *and* λ_3 *is stored in* R_1*;*

At t=3∼4, λ_2 *wins* λ_3 *in the arbitration of* k_{12} *and utilizes* k_{12} *and* k_{23}*,* λ_3 *utilizes* k_{01}*;*

At t=4, λ_1 *and* λ_2 *arrive at* R_3*,* λ_3 *is stored in* R_1*;*

At t=4∼11, λ_1 *wins* λ_2 *in the arbitration of* k_{34} *and utilizes* k_{34}*,* λ_2 *wins* λ_3 *in the arbitration of* k_{12} *and utilizes* k_{12} *and* k_{23}*,* λ_3 *utilizes* k_{01}*;* λ_2 *and* λ_3 *are stored in* R_3 *and* R_1*, respectively;*

At t=11, the tail flit of λ_3 *arrives at* R_1 *and the whole packet (10 flits) is stored in* R_1*;*

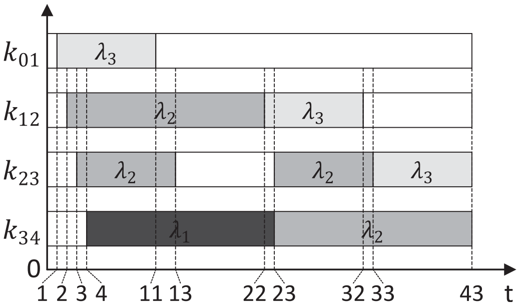


Fig. 3. Timing diagram of inter-router link utilizations.

At t=11∼13, λ_1 *wins* λ_2 *in the arbitration of* k_{34} *and utilizes* k_{34}*,* λ_2 *wins* λ_3 *in the arbitration of* k_{12} *and utilizes* k_{12} *and* k_{23}*;* λ_2 *is stored in* R_3*;*

At t=13, 10 flits of λ_2 *are stored in* R_3 *and the corresponding buffer is filled up;*

At t=13∼22, λ_1 *wins* λ_2 *in the arbitration of* k_{34} *and utilizes* k_{34}*,* λ_2 *wins* λ_3 *in the arbitration of* k_{12} *and utilizes* k_{12} *(*λ_2 *cannot be forwarded to* R_3 *through* k_{23} *because of backpressure);* λ_2 *is stored in* R_2*;*

At t=22, the tail flit of λ_2 *arrives at* R_2 *and each of* R_2 *and* R_3 *stores half a packet (10 flits);*

At t=22∼23, λ_1 *wins* λ_2 *in the arbitration of* k_{34} *and utilizes* k_{34}*,* λ_3 *utilizes* k_{12}*;*

At t=23, the tail flit of λ_1 *arrives at* R_4 *and* λ_3 *arrives at* R_2*;*

At t=23∼32, λ_2 *wins* λ_3 *in the arbitration of* k_{23} *and utilizes* k_{23} *and* k_{34}*,* λ_3 *utilizes* k_{12}*;* λ_3 *is stored in* R_2*;*

At t=32, the tail flit of λ_3 *arrives at* R_2 *and the whole packet (10 flits) is stored in* R_2*;*

At t=32∼33, λ_2 *wins* λ_3 *in the arbitration of* k_{23} *and utilizes* k_{23} *and* k_{34}*;*

At t=33, the tail flit of λ_2 *arrives at* R_3 *and 10 flits are stored in* R_3*;*

At t=33∼43, λ_2 *and* λ_3 *utilizes* k_{34} *and* k_{23}*, respectively;*

At t=43, the tail flits of λ_2 *and* λ_3 *arrive at* R_4 *and* R_3*, respectively;*

At t=44, the whole packet of λ_3 *is ejected to the destination node.*

We can observe that the delays of λ_1 *and* λ_2 *are 21 and 43, respectively, both of which meet the deadlines, and the delay of* λ_3 *under this situation is 44, invalidating the time constraint, 40.*

Analytic Results.

If using the SB's model and the SLA+ model to analyze this case, we can get:

$R_1^{SB} = 21 < D_1$;

$R_2^{SB} = 45 < D_2$;

$R_3^{SB} = 38 < D_3$;

$R_1^{SLA+} = 21 < D_1$;

$R_2^{SLA+} = 43 < D_2$;

$R_3^{SLA+} = 34 < D_3$.

λ_1*,* λ_2 *and* λ_3 *all satisfy their corresponding deadlines.*

The analytics results in this situation are however incorrect. This implies that the SB's model and the SLA+ model are not capable of this application.

While the analysis models for direct interference (Eqs. (2) and (6)) are correct, we find that the treatments for indirect interference (Eqs. (3) and (7)) are only partially valid in the sense that they work for indirect interference present at the

[2] Single-cycle router switches any flit from one hop to the next hop in one cycle and credit feedback delay (CF) is also one cycle.

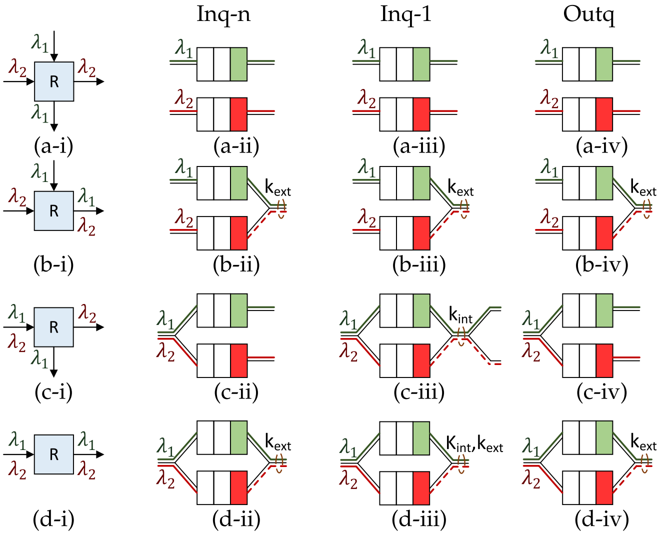


Fig. 4. Block diagram and contention models of two traffic flows within a shared router.

upstream of the observed flow, but not for indirect interference present at the *downstream* of the observed flow when the buffer size is small or Inq-1 is employed. We call the former *upstream indirect interference* and the latter *downstream indirect interference*. The upstream indirect interference is caused by flows in S_{I_i} that share links with intermediate flows at the upstream positions of links shared by intermediate flows and the observed flow, and the same goes for downstream indirect interference, except that the positions are at downstream. For example, as shown in Fig. 1a, λ_1 and λ_2 induce upstream and downstream indirect interferences to λ_5 through λ_3, respectively.

In Section 4, we first illustrate different behaviors of upstream and downstream indirect interferences to point out the reason that why the SB's model and the SLA+ model cannot be used to estimate delay upper-bounds for NoCs composed of Inq-n/Outq routers with small buffer sizes or of Inq-1 routers, then give new formulas for the delay bound analysis.

4 Extended Analysis Model

4.1 Contention Model within a Shared Router

A contention among traffic flows occurs when more than one traffic flow requests the same internal or external link simultaneously. We use k_{int} and k_{ext} to represent an internal link and an external link, respectively. Hence, a traffic flow can contend with others only when they are in the same node. There are four relations between any two traffic flows in a shared router.

- *2in-2out*. Two traffic flows are transmitted into a shared router through two input ports and stored in VCs of the corresponding input units (for an Inq-n or Inq-1 router) or output units (for an Outq router), then they are forwarded to the two different output ports, shown in Fig. 4a-i. No contention is caused since there is no shared link between these two traffic flows for all of the three architectures. Figs. 4a-ii, 4a-iii, and 4a-iv illustrate the contention models within a router with the three architectures, respectively.
- *2in-1out*. Two traffic flows arrive at a shared router on different input ports and request the same external link, shown in Fig. 4b-i. The simultaneous requests to an external link can lead to a contention, and the contention models are the same for the three architectures (shown in Figs. 4b-ii, 4b-iii, and 4b-iv).
- *1in-2out*. As shown in Fig. 4c-i, a shared router receives two traffic flows on the same input port and transmits them to two different downstream nodes through two external links. For an Inq-n or Inq-1 router, two traffic flows are stored in different VCs of the same input unit. For an Inq-n router, they are sent to the next nodes through different internal and external links. For an Inq-1 router, they share the same internal link to the switch and then pass through different external links to the next nodes. For an Outq router, two traffic flows are stored in different output units, which they request, and forwarded to the corresponding output ports. Hence, under this situation, two traffic flows can traverse this router without a contention and the contention models are the same for Inq-n and Outq routers (shown in Figs. 4c-ii and 4c-iv, respectively), but as seen in Fig. 4c-iii, the contention can occur in an Inq-1 router because of internal link sharing.
- *1in-1out*. Two traffic flows come from the same upstream node, and go to the other downstream node (shown in Fig. 4d-i). The shared links can result in a contention and the contention models of the three architecture routers are shown in Figs. 4d-ii, 4d-iii, and 4d-iv, respectively. For an Inq-1 router, the two flows share internal and external links simultaneously. If a flit wins the arbitration of the internal link, it also wins the arbitration of the external link since another traffic flow does not request the external link until it gets the allocation of the internal link. Hence, we can take the internal and the external links as one shared resource in this model, as shown in Fig. 4d-iii.

4.2 Upstream and Downstream Indirect Interference

It is clear from the discussion above that the contention models when employing Inq-n routers are the same as those when Outq routers are used. Hence, we divide routers into two categories: Inq-n/Outq routers and Inq-1 routers, to explore the influence of contentions among traffic flows. In this section, small buffer size means the buffer space of a VC is not large enough and back-pressure exists, and we use conceptually infinite buffer size to represent the situation that the buffer size is large enough to eliminate back-pressure.

4.2.1 Upstream Indirect Interference

Fig. 5a shows an upstream indirect interference. Flow λ_1 located at the *upstream* position of λ_3 indirectly interferes with λ_3 through λ_2. Figs. 5b and 5c, and Figs. 5d and 5e illustrate the contention models when employing Inq-n/Outq routers with small buffer sizes and infinite buffer sizes, respectively, as described below.

1) λ_1 blocks the intermediate flow λ_2 on the link k_{s0}, and λ_2 cannot interfere with λ_3, as shown in Figs. 5b and 5d.

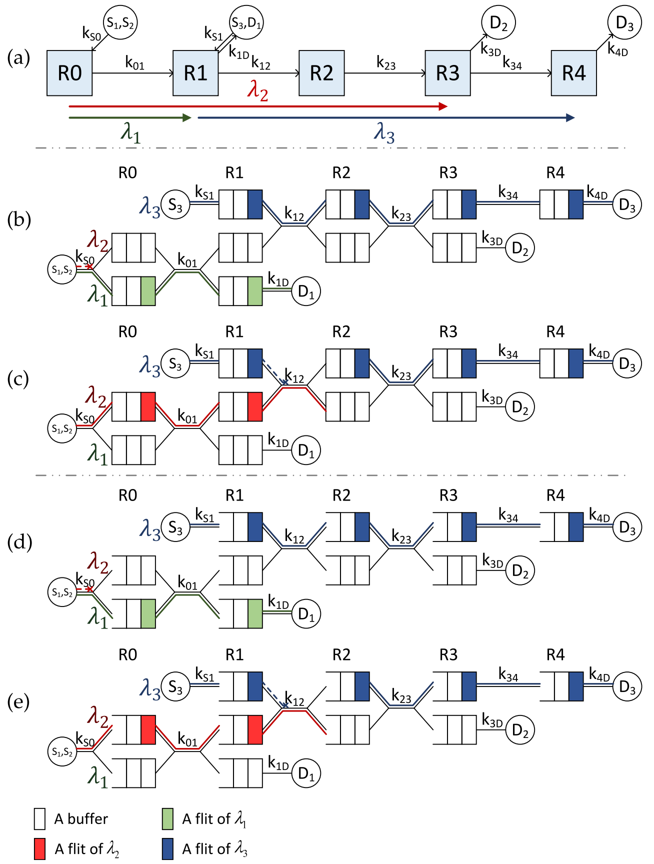


Fig. 5. Illustration of upstream indirect contention when employing Inq-n/Outq routers. (a) Block diagram. (b) and (c) Contention models when the buffer size is small. (d) and (e) Contention models when the buffer size is infinite.

2) After the packet of λ_1 leaves R_1, λ_2 is transmitted to D_2 and wins the arbitration of k_{12} against λ_3, i.e., λ_3 is blocked by λ_2, as shown in Figs. 5c and 5e.

In Fig. 5, we observe that the packet from λ_2 can be blocked by upstream indirect higher priority traffic flow λ_1 before it arrives at k_{12}. This may lead to a deviation between the presumed arrival time and the real arrival time of the packet from λ_2 reaching k_{12}. This deviation is called *interference jitter* and the quantitative analysis is detailed introduced in Section 4.3.2.

For Inq-1 routers, the models are slightly different from and the contention scenario is similar to those for Inq-n/Outq routers. The upstream indirect higher priority flow λ_1 also interferes with λ_3 by causing the interference jitter. Hence, the models are omitted for brevity.

The interference jitter is handled properly by the SB's model and the SLA+ model, and can occur in wormhole NoCs composed of Inq-n, Inq-1 or Outq routers with any buffer size. However, this analysis is *not applicable to downstream indirect interference* due to different behaviors, as we illustrate below.

4.2.2 Downstream Indirect Interference

Fig. 6a shows a downstream indirect interference, symmetric to Fig. 5a. Flow λ_1 located at the *downstream* position of λ_3 indirectly interferes with λ_3 through λ_2. Under this situation, one flit from the intermediate flow λ_2 may block the observed flow λ_3 multiple times at multiple shared links when the intermediate flow progressively moves forward. We illustrate this phenomena step by step. Figs. 6b, 6c, and 6d, and Figs. 6e, 6f, and 6g show the contention models when employing Inq-n/Outq routers with small buffer sizes and infinite buffer sizes, respectively, as described below.

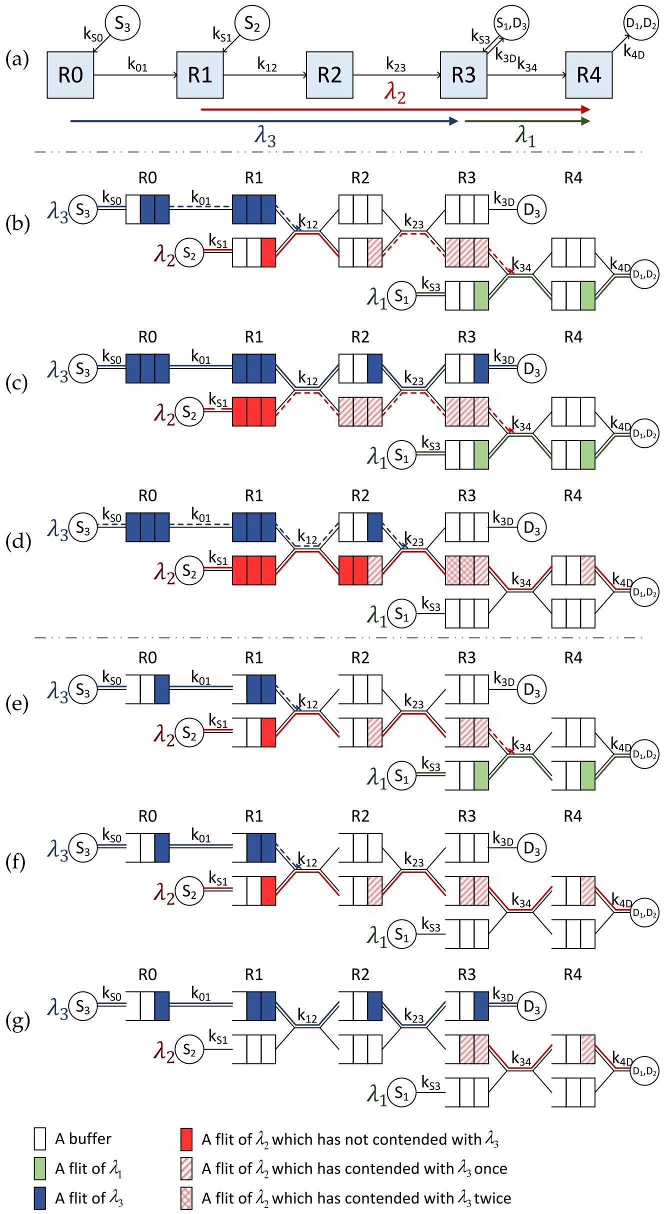


Fig. 6. Illustration of downstream indirect contention when employing Inq-n/Outq routers. (a) Block diagram. (b), (c) and (d) Contention models when the buffer size is small. (e), (f) and (g) Contention models when the buffer size is infinite.

1) λ_1 blocks the intermediate flow λ_2 on k_{34}. Before the buffers allocated to λ_2 in R_2 and R_3 are filled up, the flits of λ_2 are transmitted to and stored in R_2 and R_3. λ_3 is blocked by λ_2 on k_{12}, as shown in Figs. 6b and 6e.
2) If the buffer size is small, λ_2 can fill up its buffers in R_2 and R_3. Once it happens, a backpressure signal stops transmission from the preceding router to R_2 and R_3 while λ_1 is still contenting with λ_2 on k_{34}. Thus, λ_3 gets the allocations of the shared links and is transmitted without suffering interference, as shown in Fig. 6c. If the buffer size is infinite, the

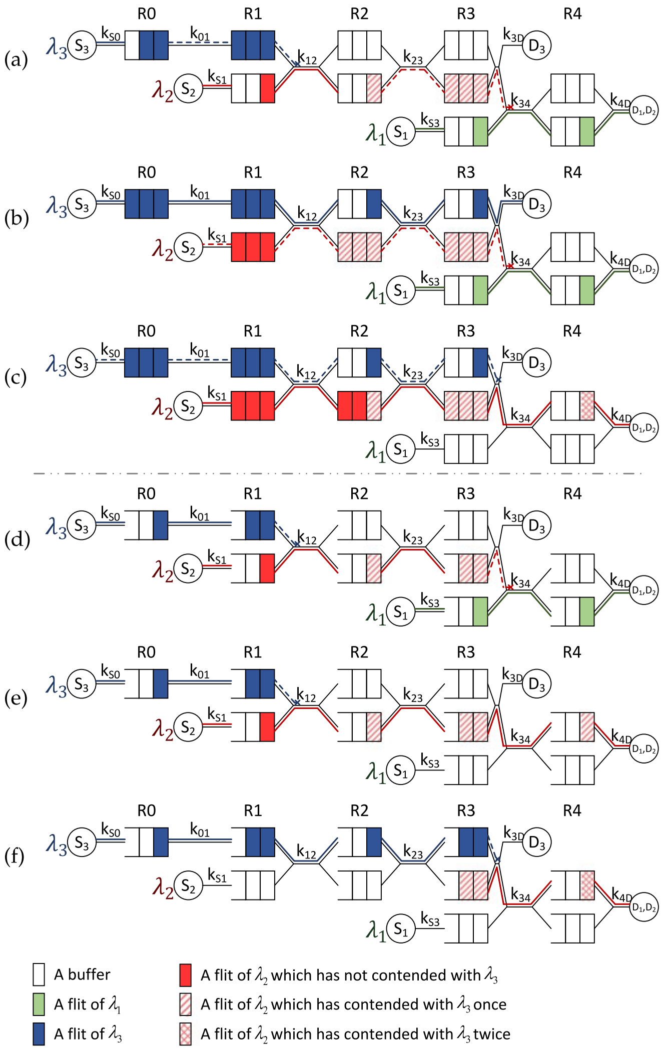


Fig. 7. Illustration of downstream indirect contention when employing Inq-1 routers. (a) Block diagram. (b), (c) and (d) Contention models when the buffer size is small. (e), (f) and (g) Contention models when the buffer size is infinite.

buffers cannot be filled up and λ_3 is always blocked by λ_2 on k_{12} until the packet of λ_2 leaves k_{12}.

3) If the buffer size is small, the transmission of λ_2 is resumed after the entire packet from λ_1 passes k_{34}. λ_2 wins the arbitration of k_{12} and k_{23} against λ_3. Note that the flits of λ_2, which are stored in R_2 in Fig. 6c, have contended with λ_3 once and can interfere with λ_3 again, shown as the red grid rectangles in Fig. 6d. This phenomenon, which we call *multi-point progressive blocking*, leads to the result that the interference caused by a packet of λ_2 may exceed C_2. If the buffer size is infinite, since the buffer of λ_2 in R_2 only contains pipelined flit(s), once λ_3 arrives at R_2, the tail flit of λ_2 reaches R_3. Hence, the flit(s) of λ_2 stored in R_2 interfere with λ_3 at most once. Since the flits of λ_2 stored in R_3 cannot interfere with λ_3, there is no flit of λ_2 that can contend with λ_3 more than once, as shown in Fig. 6g.

Hence, the MPB exists in wormhole NoCs composed of Inq-n/Outq routers with small buffer sizes.

Figs. 7a, 7b, and 7c, and Figs. 7d, 7e, and 7f show contention models when Inq-1 routers with small buffer sizes and infinite buffer sizes are employed, respectively. Since λ_2 and λ_3 share the internal link in R_3, the flits of λ_2 stored in R_3 can interfere with λ_3. As shown in Fig. 7c, if the buffer size is small, the flits of λ_2, which have blocked λ_3 once and are stored in both R_2 and R_3, can interfere with λ_3 again. This means that the MPB exists and it is more obvious than that in a NoC consisting of Inq-n/Outq routers (in this situation, as shown in Fig. 6d, only flits stored in R_2 block λ_3 again). If the buffer size is infinite, unlike when Inq-n/Outq routers are employed, the flits of λ_2, which have blocked λ_3 once and are stored in R_3, can block λ_3 again on the internal link, as shown in Fig. 7f. Hence, there is also the MPB when Inq-1 routers with large buffer sizes are employed.

TABLE 2
Existence of MPB

Architecture	Small Buffer Size	Large Buffer Size
Inq-n	Y	N
Inq-1	Y	Y
Outq	Y	N

As we can observe, this behavior has an impact on delay upper-bounds and largely differs from that of the upstream indirect interference. The SB's model handles the downstream indirect interference in the same way as the upstream indirect interference and the SLA+ model only considers the blockage caused by back-pressure (for example, in Fig. 6b, λ_3 is blocked from accessing k_{01} since the corresponding buffer in R_1 is full). The SB's model and the SLA+ model fail to catch such MPB behavior. This implies that the SB's model and the SLA+ model are not capable of estimating delay upper-bounds when the MPB exists. Note that the interference jitter also exists besides the MPB for downstream indirect interference [30].

Therefore, we summarize the relationships of the MPB with router architectures and buffer sizes in Table 2, in which Y means MPB existence and N MPB non-existence. We can observe from Table 2 that the SB's model and the SLA+ model only apply to Inq-n/Outq routers with large buffer sizes.

4.3 Up/Down-Stream Indirect Interference Analysis

4.3.1 Definition

To handle indirect interference properly, we need to further divide indirect interference flows into upstream indirect interference flow set denoted by $S_{I_i}^{U_j}$, and downstream indirect interference flow set denoted by $S_{I_i}^{D_j}$. Let γ_{ji}^n indicate a shared link, which is the nth link in λ_j's routing path, between λ_j and λ_i. Γ_{ji} denotes the set of γ_{ji}^n. Since the superscripts of links in λ_j's routing path are in sequence, the link in Γ_{ji} with the minimum superscript is the first shared link between λ_i and λ_j. Let m_{ji} be the minimum superscript in Γ_{ji}. $S_{I_i}^{U_j}$ and $S_{I_i}^{D_j}$ are defined as

$$\begin{aligned} S_{I_i}^{U_j} &= \{\lambda_k | \exists \gamma_{jk}^n \in \Gamma_{jk} : n < m_{ji} \\ &\quad \lambda_j \in S_{D_i}, \forall \lambda_k \in S_{D_j} \cap S_{I_i}\}, \\ S_{I_i}^{D_j} &= \{\lambda_k | \exists \gamma_{jk}^n \in \Gamma_{jk} : n > m_{ji} \\ &\quad \lambda_j \in S_{D_i}, \forall \lambda_k \in S_{D_j} \cap S_{I_i}\}, \end{aligned}$$

and their relationship to S_{I_i} is

$$S_{I_i} = \bigcup_{\forall \lambda_j \in S_{D_i}} (S_{I_i}^{U_j} \cup S_{I_i}^{D_j}).$$

Since indirect higher priority traffic flow $\lambda_k \in S_{I_i}$ shares no link with λ_i, $\nexists \gamma_{jk}^n \in \Gamma_{jk} : n = m_{ji}$.

4.3.2 Analysis

The interference I_i suffered by the observed traffic flow λ_i is composed of all I_{ji}, which is caused by $\lambda_j \in S_{D_i}$ itself and the indirect higher priority traffic flows in S_{D_j}, i.e.,

$$I_i = \sum_{\forall \lambda_j \in S_{D_i}} I_{ji}. \tag{10}$$

Theorem 1. *The upper-bound of interference suffered by λ_i from a direct higher priority flow λ_j can be given by*

$$I_{ji} = IT_{ji} \cdot IP_{ji}. \tag{11}$$

Proof. A packet of λ_i may be blocked by multiple packets from λ_j. We use the terms IP_{ji}^k and IT'_{ji} to denote the interference to λ_i caused by a packet of λ_j and the number of packets from λ_j that contend with a packet of λ_i, respectively. The upper-bounds of IP_{ji}^k and IT'_{ji} are IP_{ji} and IT_{ji}, respectively.

According to the assumption that $D_i \leq (T_i - J_i)$, there is at most one packet from λ_j that contends with λ_i at a time, i.e., the interferences caused by the packets of λ_j are independent of each other. The interference from λ_j to λ_i, I'_{ji}, can be given by

$$I'_{ji} = \sum_{k=1}^{IT'_{ji}} IP_{ji}^k \leq \sum_{k=1}^{IT'_{ji}} IP_{ji} \leq IT_{ji} \cdot IP_{ji}.$$

Hence, we get $I_{ji} = IT_{ji} \cdot IP_{ji}$. □

The maximum interference suffered by λ_j from traffic flows in $S_{I_i}^{D_j}$ is as follows:

$$I_{ji}^D = \sum_{\lambda_k \in S_{I_i}^{D_j}} I_{kj}.$$

In the following we reason about the bounds for IT_{ji} and IP_{ji} through two theorems, one for each.

Theorem 2. *The maximum number of packets from λ_j that can contend with a packet of λ_i is*

$$IT_{ji} = \left\lceil \frac{R_i + J_j + I_j}{T_j} \right\rceil, \tag{12}$$

where I_j exists if and only if $S_{I_i} \cap S_{D_j} \neq \phi$.

Proof.

Case 1: $S_{I_i} \cap S_{D_j} = \phi$.

λ_i only suffers direct interference from λ_j. Hence

$$IT_{ji} = \left\lceil \frac{R_i + J_j}{T_j} \right\rceil.$$

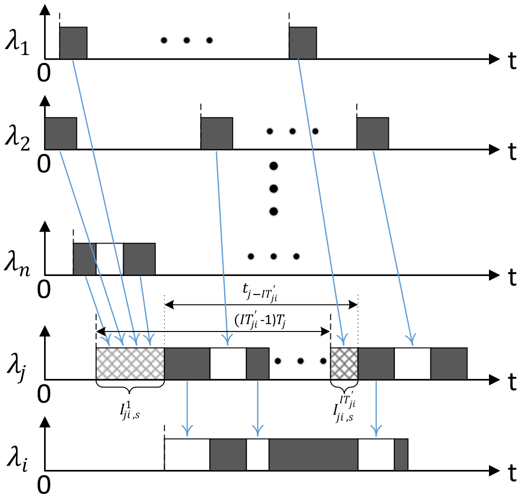


Fig. 8. Analysis of upstream indirect interference.

Case 2: $S_{I_i} \cap S_{D_j} \neq \phi$.

λ_i suffers direct interference from λ_j and indirect interference through λ_j. Fig. 8 shows the time diagrams of the packet transmission of the observed traffic flow (λ_i) and a direct higher priority traffic flow (λ_j), and a set of flows in S_{D_j} ($\lambda_1, \lambda_2, \ldots, \lambda_n$), which affect λ_i through λ_j. The packet of λ_i suffers the interference from IT'_{ji} packets of λ_j and those packets are labeled from 1 to IT'_{ji} in order. The vertical dash line is the time of a packet starting to be served without suffering any interference, of which the interval between the 1st and the kth packets of λ_j is $(k-1)T_j$, $k = 1, 2, \ldots, IT'_{ji}$. The gray rectangle means that a packet is being transmitted, and both the white and the grid rectangles indicate that a packet is delayed by release jitter, or blocked and fails to get the allocation of shared links. Specifically, the grid rectangle, of which the length in time is denoted by $I_{ji,s}^k$ for the kth packet, includes the release jitter and the interference suffered by a packet of λ_j before it starts to be served. Hence, the service start time interval between the 1st and the kth packets of λ_j is given by

$$S_j^{1,k} = (k-1)T_j - I_{ji,s}^1 + I_{ji,s}^k. \tag{13}$$

In Fig. 8, we observe that the packet of λ_i suffers interference caused by IT'_{ji} packets from λ_j when the following condition is met: the tail flit of the packet from λ_i leaves the shared links after the real service start time of the IT'_{ji} packet from λ_j and no later than that of the $IT'_{ji} + 1$ packet, i.e.,

$$S_j^{1,IT'_{ji}} < R_i \leq S_j^{1,IT'_{ji}+1} \quad \Rightarrow$$

$$(IT'_{ji} - 1)T_j - I_{ji,s}^1 + I_{ji,s}^{IT'_{ji}} < R_i \leq IT'_{ji} T_j - I_{ji,s}^1 + I_{ji,s}^{IT'_{ji}+1}.$$

Thus, we can get

$$\frac{R_i + I_{ji,s}^1 - I_{ji,s}^{IT'_{ji}+1}}{T_j} \leq IT'_{ji} < \frac{R_i + I_{ji,s}^1 - I_{ji,s}^{IT'_{ji}}}{T_j} + 1. \tag{14}$$

As shown in Fig. 8, Eqs. (13) and (14), the service start time affects the service start time interval and then IT'_{ji}. If

$I^1_{ji,s} > I^{IT'_{ji}}_{ji,s}$, the reduction of $S^{1,IT'_{ji}}_j$ may lead to the result that a packet of λ_i suffers more interference from λ_j in the same period and even increases IT'_{ji}. We observe in Eq. (14) that when $I^1_{ji,s} - I^{IT'_{ji}}_{ji,s}$ reaches its maximum, the upper-bound of IT'_{ji} can go for a maximum, i.e., for any contention scenario, IT'_{ji} cannot be larger than

$$IT_{ji} = max\{IT'_{ji}\} < \frac{R_i + max\{I^1_{ji,s} - I^{IT'_{ji}}_{ji,s}\}}{T_j} + 1, \tag{15}$$

and the largest integer satisfying Eq. (15) is

$$IT_{ji} = \left\lceil \frac{R_i + max\{I^1_{ji,s} - I^{IT'_{ji}}_{ji,s}\}}{T_j} \right\rceil. \tag{16}$$

The grid rectangle cannot be larger than the sum of the release jitter (max. J_j) and the interference (max. I_j) that a packet of λ_j suffers. Hence, $I^k_{ji,s}$ satisfies

$$0 \le I^k_{ji,s} \le I_j + J_j,$$

and we can get

$$max\{I^1_{ji,s} - I^{IT'_{ji}}_{ji,s}\} = I_j + J_j. \tag{17}$$

Substituting Eq. (17) into Eq. (16), we get

$$IT_{ji} = \left\lceil \frac{R_i + J_j + I_j}{T_j} \right\rceil.$$

□

Lemma 1. *The multi-point progressive blocking of λ_i can only be caused by traffic flows in $S^{D_j}_{I_i}$.*

Proof. According to the analyses of the cases in Figs. 6 and 7, the MPB of the observed traffic flow λ_i is contributed by the flits of λ_j, which interfere with λ_i more than once. For indirect higher priority traffic flow λ_k, which can lead to the MPB, a necessary condition is that some flits of λ_j, which are stored in buffers caused by the blocking of λ_k, can contend with λ_i before they are blocked by λ_k. Hence, the shared links between λ_k and λ_j should be at the downstream of the shared links between λ_j and λ_i, i.e., λ_k must be at the downstream position, $\lambda_k \in S^{D_j}_{I_i}$. Therefore, only traffic flows in S_{I_i} may lead to the MPB to λ_i. □

Theorem 3. *The maximum interference suffered by λ_i from a packet of λ_j is*

$$IP_{ji} = C_j + I^D_{ji}. \tag{18}$$

Proof. The interference caused by a packet of λ_j to λ_i is given by

$$IP^k_{ji} = \sum_{n=1}^{Z_j} N^n_{ji} I_{flit}, \tag{19}$$

where N^n_{ji} is the number of times that the nth flit of a λ_j's packet contends with λ_i, and I_{flit} means the interference caused by a flit at one time.

If the MPB through λ_j to λ_i does not exist, the number of contentions of a flit, N^n_{ji}, satisfies

$$N^n_{ji} \le 1, \ \forall\, n = 1, 2, \ldots, Z_j,$$

and if it exists, N^n_{ji} satisfies

$$N^n_{ji} > 1, \ \exists\, n = 1, 2, \ldots, Z_j.$$

Thus, IP^k_{ji} satisfies

$$IP^k_{ji} \le Z_j I_{flit} + \sum_{\forall n \in M_{ji}} (N^n_{ji} - 1) I_{flit}, \tag{20}$$

where M_{ji} denotes the set of the flits of a λ_j's packet that cause the MPB to λ_i. The first term implies that each flit contends with λ_i once, thus is no larger than C_j. The second term stands for the extra interference caused by the MPB. From Lemma 1, the MPB can only be caused by downstream higher priority traffic flows. The flits in M_{ji} can contend with λ_i more than once because they are blocked by the flows in $S^{D_j}_{I_i}$. Hence, the second term in Eq. (20) is no larger than the sum of interference suffered by λ_j from traffic flows in $S^{D_j}_{I_i}$, which is I^D_{ji}.

The maximum of IP^k_{ji} is given by

$$IP_{ji} = max\{IP^k_{ji}\} = C_j + I^D_{ji}.$$

□

In summary, the interference jitter is used to determine the number of times λ_j can interfere with a packet of λ_i (IT_{ji}) while the interference of each λ_j's packet (IP_{ji}) can be inflated by the MPB. Thus, with respect to Eq. (4), we are maintaining J^I_i (using IT_{ji}) while increasing C_j (using IP_{ji}). The influence of buffers in the delay upper-bound estimation is qualitatively bounded by I^D_{ji} in Eq. (18) and independent of specific buffer sizes.

4.4 Our Analysis Model

We now account for both direct and indirect interferences. The delay upper-bound of a real-time flow λ_i, which suffers one or more kinds of interference, is given by Theorem 4.

Theorem 4. *The delay upper-bound R_i of λ_i under any contention scenario is given by*

$$R_i = \sum_{\forall \lambda_j \in S_{D_i}} \left\lceil \frac{R_i + J_j + I_j}{T_j} \right\rceil (C_j + I^D_{ji}) + C_i, \tag{21}$$

where I_j exists if and only if $S_{I_i} \cap S_{D_j} \neq \phi$.

Proof. According to Eqs. (1) and (10), and Theorems 1, 2, and 3, the delay upper-bound can be obtained as shown in Eq. (21).

For λ_i, there exist five kinds of contention scenarios. We now validate the correctness of Eq. (21) one by one.

- *No contention*. The delay of λ_i is equal to C_i. For Eq. (21), S_{D_i} is empty. Thus, $R_i = C_i$.
- *Direct contention only.* For Eq. (21), S_{I_i} is empty and $I_j = I^D_{ji} = 0$. Consequently, $R_i = \sum_{\forall \lambda_j \in S_{D_i}} \lceil (R_i + J_j)/T_j \rceil C_j + C_i$ is consistent with the SB's model, which is correct in this situation.

- *Direct and upstream indirect contentions only.* λ_i experiences direct interference from direct contention and interference jitter from upstream indirect contention. For Eq. (21), $S_{I_i}^{D_j}$ is empty and $I_{ji}^D = 0$. Therefore, $R_i = \sum_{\forall \lambda_j \in S_{D_i}} \lceil (R_i + J_j + I_j)/T_j \rceil C_j + C_i$. According to Theorem 2, this equation correctly handles both direct and upstream indirect contentions.
- *Direct and downstream indirect contentions only.* λ_i suffers direct interference from direct contention and interference jitter and MPB from downstream indirect contention. For Eq. (21), $R_i = \sum_{\forall \lambda_j \in S_{D_i}} \lceil (R_i + J_j + I_j)/T_j \rceil (C_j + I_{ji}^D) + C_i$. According to Theorem 3, this equation correctly deals with direct and downstream indirect contentions.
- *Direct, and upstream and downstream indirect contentions.* λ_i sustains direct interference, interference jitter and MPB. For Eq. (21), $R_i = \sum_{\forall \lambda_j \in S_{D_i}} \lceil (R_i + J_j + I_j)/T_j \rceil (C_j + I_{ji}^D) + C_i$. In this situation, according to Theorems 2 and 3, this equation estimates the correct delay upper-bound for λ_i.

Hence, the correctness of Eq. (21) under any contention scenario is validated. □

Note that Eq. (21) is different from Eq. (17) in [29] in the calculation of the interference jitter since the interference jitter can be caused by all flows in S_{D_j} ($\lambda_j \in S_{D_i}$ and $S_{I_i} \cap S_{D_j} \neq \phi$) according to [30]. In Eq. (21), since the buffering impact is generally counted, the tightness of our model can be improved in the future by quantitatively taking specific buffer sizes into consideration. The iterative algorithm is used to solve this equation to obtain the delay upper-bound. $R_i^{(n)}$ indicates the nth iterative value of R_i, the iterative initial value is $R_i^{(0)} = \sum_{\forall \lambda_j \in S_{D_i}} \lceil (J_j + I_j)/T_j \rceil (C_j + I_{ji}^D) + C_i$ and the iteration terminates when $R_i^{(n+1)} = R_i^{(n)}$ or $R_i^{(n)} > D_i$ or the number of iteration exceeds a pre-set limitation. If the iteration terminates with $R_i^{(n+1)} = R_i^{(n)}$, the delay upper-bound of λ_i is $R_i = R_i^{(n)}$.

We can now qualitatively compare our model in Eq. (21) to the SB's model in Eq. (4) and the SLA+ model in Eq. (9), and summarize two cases:

1) For Inq-n/Outq routers with large buffer sizes, all models work, but the SLA+ model gives the tightest delay bound, followed by the SB's model and our model. Specifically, for S_{D_i} only or S_{D_i} plus *only* $S_{I_i}^{U_j}$, the SB's model and our model are consistent; and for S_{D_i} plus *any* $S_{I_i}^{D_j}$ (with or without $S_{I_i}^{U_j}$), which causes I_{ji}^D, the SB's model gives a tighter result.
2) For Inq-n/Outq routers with small buffer sizes or for Inq-1 routers, MPB exists, and the SB's model and the SLA+ model cannot be used to estimate delay upper-bounds. Our model can give the correct estimations in this situation.

5 Experimentation

In this section, we quantitatively evaluate the SB's model, the SLA+ model and our extended model on priority preemptive wormhole NoCs with different router architectures and buffer sizes. We conduct a case study with both analysis and simulations to: 1) give a detailed analysis example; 2) quantitatively contrast the analytic results of the SB's model, the SLA+ model and our model against simulated results; and 3) discuss application limitations in a concrete context.

TABLE 3
Flow Parameters

Traffic Flow	C_i (Z_i, H_i)	T_i	D_i	J_i
λ_1	30 $(27, 4)$	150	100	0
λ_2	30 $(28, 3)$	150	100	0
λ_3	150 $(144, 7)$	600	300	0
λ_4	100 $(98, 3)$	600	550	0
λ_5	100 $(96, 5)$	300	250	0

5.1 Setup

We use the illustrative example in Fig. 1a as our case study, which is simple to understand and has necessary interference patterns to compare the three models. There are five periodic flows with their parameters listed in Table 3.

Given the flows and their interference patterns in the network, we can obtain per-flow delay upper bounds using the three models. To obtain simulation results, we build a network behavior simulator fulfilling the assumptions for the delay bound analysis problem. Specifically, the network performs wormhole switching, and the router uses flit-sized virtual channels, per-flow buffering and priority-based link arbitration. The router is a single-cycle router, meaning that switching any flit from one hop to the next hop and transmitting a credit from one hop back to the previous hop take one cycle each. If a flow has flits stored in a router and the corresponding VC buffer in the downstream router has free space, this flow requests the internal and the external links. If there are more than one flow in a router requesting the shared links simultaneously, the flow with the highest priority is transmitted. When a higher priority flow is blocked in a router because the buffer in the downstream router is full, it does not request to be transmitted, and the router allows a lower priority flow, which requests the link, to proceed. As a result of wormhole transmission, the non-contentional delay C_i in the SB's and our models and the basic stage delay L_i in the SLA+ model equal $(Z_i + H_i - 1)$ and Z_i, respectively.

The simulations are performed in NoCs composed of Inq-n, Inq-1 and Outq routers, respectively. According to the analysis in [28], the lower bound of the buffer space in a VC to eliminate back-pressure could not be larger than $max\{L_1, L_2, L_3, L_4, L_5\}$, namely, 144. Hence, we configure buffer space in a VC as two sizes in the simulations: 10 and 1,000. When the buffer size is 10, back-pressure occurs in our simulations. When the buffer size is 1,000, in contrast, no back-pressure exists.

Traffic flows generate fixed-size packets periodically according to Table 3. All flows are injected synchronously. To simulate all possible contention scenarios, they are generated with phase shifts up to their corresponding periods. For example, flow λ_1 has an injection phase of 0 to 149 cycles, and flow λ_5 0 to 299 cycles. For each combination of the phase shifts under a particular buffer size setting, the simulations include two basic phases: warm-up and

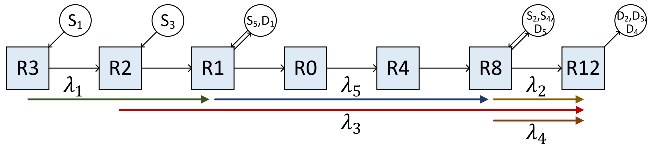


Fig. 9. Flow interference graph for the example.

measurement. Both phases are the least common multiple (600) of the periods of the five flows in length. After the warm-up phase, the measurement phase begins and records the delay of each packet in this phase. Then we compare the statistics to the analytic results.

5.2 Basic Comparison

To show the direct and indirect interferences more clearly, we draw a flow interference graph for the example in Fig. 9.

To clearly associate this quantitative comparison with the qualitative comparison (Section 4.4), we also list the *direct*, *upstream indirect* and *downstream indirect interference sets*, i.e., S_{D_i}, $S_{I_i}^{U_j}$ and $S_{I_i}^{D_j}$, respectively, for all flows, in Table 4. Table 5 gives the three analytic delay bounds (independent of the buffer size), and the maximum simulated delays for each flow with the different router architectures and buffer sizes (B). We combine the results of the simulations when the router architectures are Inq-n and Outq since the results are totally the same. Since the worst-case delay of λ_5 arises under many sets of conditions, we give one set of conditions for each configuration. For Inq-n/Outq routers with $B = 1,000$, it is that λ_1, λ_2, λ_3, λ_4 and λ_5 are all injected at time 0. For Inq-n/Outq routers with $B = 10$ and Inq-1 routers with $B = 10$ and $B = 1,000$, all λ_1, λ_3, λ_4 and λ_5 are injected at time 0 and λ_2 is injected at time 32. Here the warm-up phase is not counted in the injection times.

From Table 5, we observe matching results with the qualitative comparison. Specifically,

1) For Inq-n/Outq routers with $B = 1,000$, all models obtain the delay bounds. For λ_3, which has only direct contentions (non-empty S_{D_3}), and for λ_4, which has direct (non-empty S_{D_4}) and only upstream indirect contentions (non-empty $S_{I_4}^{U_3}$), the SLA+ model gets the tightest result, and the SB's model and ours obtain the same delay bounds; and for λ_5, which has direct (non-empty S_{D_5}) and downstream indirect interference (non-empty $S_{I_5}^{D_3}$), the SLA+ model still gets the tightest delay bound (244), followed by the SB's model (250) and ours (310).
2) For Inq-n/Outq routers with $B = 10$ or for Inq-1 routers, our model still correctly estimates the delay upper-bounds, and the SB's model does not support the estimation. Although the SLA+ model claims to support small buffer sizes and consider back-pressure, it remains inapplicable in these situations, as shown in Table 5.

TABLE 4
Relationships Among All Traffic Flows

λ_i	S_{D_i}	S_{I_i}	$S_{I_i}^{U_j}$	$S_{I_i}^{D_j}$
λ_1	ϕ	ϕ	ϕ	ϕ
λ_2	ϕ	ϕ	ϕ	ϕ
λ_3	$\{\lambda_1, \lambda_2\}$	ϕ	ϕ	ϕ
λ_4	$\{\lambda_2, \lambda_3\}$	$\{\lambda_1\}$	$\{\lambda_1\}(j = 3)$	ϕ
λ_5	$\{\lambda_3\}$	$\{\lambda_1, \lambda_2\}$	$\{\lambda_1\}(j = 3)$	$\{\lambda_2\}(j = 3)$

TABLE 5
Analytic and Simulation Results

λ_i	SB's	SLA+		Our	Simulation			
					Inq-n/Outq		Inq-1	
		10*	1,000*		10*	1,000*	10*	1,000*
λ_1	30	30	30	30	30	30	30	30
λ_2	30	30	30	30	30	30	30	30
λ_3	270	260	260	270	233	233	233	233
λ_4	340	300	300	340	300	300	300	300
λ_5	250	244	244	310	**264**	244	**274**	**272**

* *Buffer size per VC* (B).

From Table 5, we observe that the delay bounds of the SLA+ model when the buffer sizes are set as 10 and 1,000 are consistent in this situation. This is because the buffer influence is not considered properly in the SLA+ model.

5.3 Buffer Size Influence

If the buffer size is large enough (no back-pressure), the variation of the buffer size cannot affect the delay of a traffic flow anymore since a flit can always be allocated a buffer in the associated VC when it requests one. In this section, to further look into the impact of buffering on the maximum delay, we simulate with different buffer sizes, which are not large enough to guarantee that the MPB does not exist under all situations.

Fig. 10 shows the distributions of λ_4's delay when the VC buffer size is 2, 4, 8, 16 and 32. All of the five distributions have the similar features and the worst-case delays are the same, 300. Since the preemptive arbitration policy is employed and λ_4 has direct and only upstream indirect contentions (no downstream indirect contention), the buffer size can only slightly influence the distributions. In this case, λ_4 suffers similar contention scenarios when the different router architectures are employed. Hence, the delay distributions of λ_4 in Figs. 10a and 10b are about the same. The worst-case delays are less than our delay bound (340).

Fig. 11 gives the distributions of λ_5's delay under the same conditions as Fig. 10. Compared to the distributions of λ_4, the five curves are more irregular and the worst-case delay varies with the buffer size. Since λ_5 has downstream indirect contentions, the buffer size and the router architecture can have more influence on the distributions through the MPB. The worst-case delays are 250, 254, 262, 257 and 244 in Fig. 11a, and 253, 259, 271, 273 and 272 in Fig. 11b, respectively. Under the various buffer sizes and the different router architectures, no packet's delay exceeds our delay bound (310).

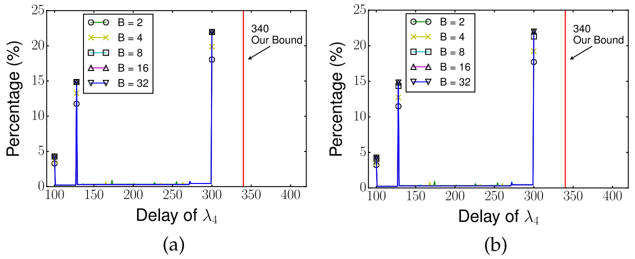
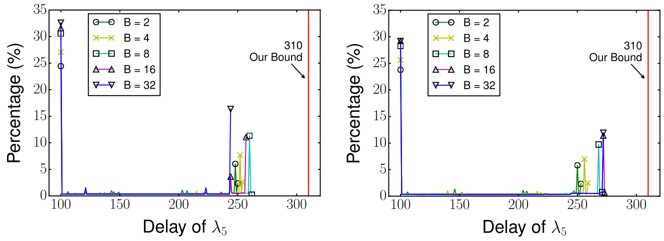


Fig. 10. λ_4 delay distributions. (a) Inq-n/Outq. (b) Inq-1.



(a) (b)

Fig. 11. λ_5 delay distributions. (a) Inq-n/Outq. (b) Inq-1.

6 Conclusion

We have presented a new delay bound analysis model for real-time communication in wormhole NoCs with priority-based preemptive arbitration. This model confirms the application restrictions of the state-of-the-art analysis models called the SB's model [9] and the SLA+ model [11], respectively, in the paper. The main cause of the restriction in the SB's model lies in the uniform treatment of *upstream* and *downstream* indirect interferences. The two different indirect interferences actually show very different blocking behaviors when packets move from one buffer to the next. The SLA+ model noticed the differences between downstream indirect interference and upstream indirect interference as well as back-pressure caused by limited buffer sizes. But this model also cannot support Inq-1 routers and small buffer sizes since the downstream indirect interference exhibits a MPB phenomenon, which was not captured by the the SB's model and the SLA+ model. We have given theorems to prove our analysis method. Furthermore, we have qualitatively and quantitatively compared the three models. In summary, our model supports one more router architecture and applies to small buffer sizes.

In our analysis, the buffering effect is qualitatively counted, and the delay bound formula is independent of buffer sizes. In the future, we plan to include buffer sizes and router architectures in the analysis model in order to more precisely quantify its impact on delay bounds and tighten delay bounds based on the SLA+ model through considering downstream indirect interference on each stage.

Acknowledgments

This research is sponsored by the National Natural Science Foundation of China under Grant No. 61300047, No. 61472152 and No. 61572209, National Natural Science Foundation of Hubei Province No. 2015CFB315, the Fundamental Research Funds for the Central Universities No. 2016YXMS019, and the 111 Project No. B07038. The authors would like to thank Leandro Soares Indrusiak, Alan Burns and Borislav Nikolić for valuable discussions about the real-time analysis models. Fei Wu is the corresponding author.

References

[1] H. Kashif, "Deployment and debugging of real-time applications on multicore architectures," Ph.D. dissertation, Dept. Elect. Comput. Eng., Univ. Waterloo, Waterloo, ON, Canada, 2015.

[2] Z. Lu, A. Jantsch, and I. Sander, "Feasibility analysis of messages for on-chip networks using wormhole routing," in *Proc. 10th IEEE Asia South Pacific Des. Autom. Conf.*, Jan. 2005, pp. 960–964.

[3] M. Millberg, E. Nilsson, R. Thid, and A. Jantsch, "Guaranteed bandwidth using looped containers in temporally disjoint networks within the nostrum network on chip," in *Proc. 7th Des. Autom. Test Eur. Conf. Exhib.*, Feb. 2004, pp. 890–895.

[4] Z. Lu and A. Jantsch, "TDM virtual-circuit configuration for network-on-chip," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 16, no. 8, pp. 1021–1034, Aug. 2008.

[5] E. Rijpkema, et al., "Trade-offs in the design of a router with both guaranteed and best-effort services for networks on chip," *IEE Proc. Comput. Digit. Tech.*, vol. 150, no. 5, pp. 294–302, Sep. 2003.

[6] S. Liu, Z. Lu, and A. Jantsch, "Highway in TDM NoCs," in *Proc. 9th ACM/IEEE Int. Symp. Netw.-on-Chip*, Sep. 2015, pp. 15–22.

[7] K. Toda, K. Nishida, S. Sakai, and T. Shimada, "A priority forwarding scheme for real-time multistage interconnection networks," in *Proc. 13th IEEE Real-Time Syst. Symp.*, Dec. 1992, pp. 208–217.

[8] B. Kim, J. Kim, S. Hong, and S. Lee, "A real-time communication method for wormhole switching networks," in *Proc. 27th Int. Conf. Parallel Process.*, Aug. 1998, pp. 527–534.

[9] Z. Shi and A. Burns, "Real-time communication analysis for on-chip networks with wormhole switching," in *Proc. 2nd ACM/IEEE Int. Symp. Netw.-on-Chip*, Apr. 2008, pp. 161–170.

[10] H. Kashif, S. Gholamian, and H. Patel, "SLA: A stage-level latency analysis for real-time communication in a pipelined resource model," *IEEE Trans. Comput.*, vol. 64, no. 4, pp. 1177–1190, Apr. 2015.

[11] H. Kashif and H. Patel, "Buffer space allocation for real-time priority-aware networks," in *Proc. 22nd IEEE Real-Time Embedded Technol. Appl. Symp.*, Apr. 2016, pp. 1–12.

[12] R. L. Cruz, "A calculus for network delay, part I: Network elements in isolation," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 114–131, Jan. 1991.

[13] L. Thiele, S. Chakraborty, and M. Naedele, "Real-time calculus for scheduling hard real-time systems," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2000, pp. 101–104.

[14] M. Boyer and D. Doose, "Combining network calculus and scheduling theory to improve delay bounds," in *Proc. 20th Int. Conf. Real-Time Netw. Syst.*, Nov. 2012, pp. 51–60.

[15] J.-Y. Le Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. Berlin, Germany: Springer-Verlag, 2001.

[16] Y. Qian, Z. Lu, and W. Dou, "Analysis of worst-case delay bounds for best-effort communication in wormhole networks on chip," in *Proc. 3rd ACM/IEEE Int. Symp. Netw.-on-Chip*, May 2009, pp. 44–53.

[17] L. T. X. Phan, S. Chakraborty, P. S. Thiagarajan, and L. Thiele, "Composing functional and state-based performance models for analyzing heterogeneous real-time systems," in *Proc. 28th IEEE Real-Time Syst. Symp.*, Dec. 2007, pp. 343–352.

[18] X. Zhao and Z. Lu, "Per-flow delay bound analysis based on a formalized microarchitectural model," in *Proc. 7th ACM/IEEE Int. Symp. Netw.-on-Chip*, Apr. 2013, pp. 1–8.

[19] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *J. ACM*, vol. 20, no. 1, pp. 46–61, Jan. 1973.

[20] N. Audsley, A. Burns, M. Richardson, K. Tindell, and A. J. Wellings, "Applying new scheduling theory to static priority pre-emptive scheduling," *Softw. Eng. J.*, vol. 8, no. 5, pp. 284–292, Sep. 1993.

[21] A. Burns, M. Nicholson, K. Tindell, and N. Zhang, "Allocating and scheduling hard real-time tasks on a point-to-point distributed system," in *Proc. 1st IEEE Workshop Parallel Distrib. Real-Time Syst.*, Apr. 1993, pp. 11–20.

[22] Z. Shi and A. Burns, "Priority assignment for real-time wormhole communication in on-chip networks," in *Proc. 29th IEEE Real-Time Syst. Symp.*, Nov. 2008, pp. 421–430.

[23] Z. Shi and A. Burns, "Real-time communication analysis with a priority share policy in on-chip networks," in *Proc. 21st Euromicro Conf. Real-Time Syst.*, Jul. 2009, pp. 3–12.

[24] Z. Shi and A. Burns, "Schedulability analysis and task mapping for real-time on-chip communication," *Real-Time Syst.*, vol. 46, no. 3, pp. 360–385, Sep. 2010.

[25] Z. Shi, A. Burns, and L. S. Indrusiak, "Schedulability analysis for real time on-chip communication with wormhole switching," *Int. J. Embedded Real-Time Commun. Syst.*, vol. 1, no. 2, pp. 1–22, Jun. 2010.

[26] A. Burns, J. Harbin, and L. S. Indrusiak, "A wormhole NoC protocol for mixed criticality systems," in *Proc. 35th IEEE Real-Time Syst. Symp.*, Dec. 2014, pp. 184–195.

[27] H. Kashif, H. D. Patel, and S. Fischmeister, "Using link-level latency analysis for path selection for real-time communication on NoCs," in *Proc. 17th IEEE Asia South Pacific Des. Autom. Conf.*, Jan. 2012, pp. 499–504.

[28] H. Kashif and H. Patel, "Bounding buffer space requirements for real-time priority-aware networks," in *Proc. 19th IEEE Asia South Pacific Des. Autom. Conf.*, Jan. 2014, pp. 113–118.

[29] Q. Xiong, Z. Lu, F. Wu, and C. Xie, "Real-time analysis for wormhole NoC: Revisited and revised," in *Proc. 26th ACM Great Lakes Symp. VLSI*, May 2016, pp. 75–80.

[30] L. S. Indrusiak, A. Burns, and B. Nikolic, "Analysis of buffering effects on hard real-time priority-preemptive wormhole networks," *arXiv:1606.02942*, Jun. 2016, https://arxiv.org/abs/1606.02942

Qin Xiong received the BS degree in electronic science and technology from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2012. He is working towards the PhD degree in the Wuhan National Laboratory for Optoelectronics at HUST. His current research interests include interconnection networks, real-time systems, and non-volatile memories.

Fei Wu received the BS and MS degrees in automation from Wuhan University of Technology, Wuhan, China, respectively, in 1997 and 2000, and the PhD degree in computer architecture from Huazhong University of Science and Technology, Wuhan, China, in 2005. She is currently an associate professor with the Division of Information Storage and Optical Display, Wuhan National Laboratory for Optoelectronics, HUST. She has published more than 30 papers, including MSST, Eurosys, ICCD, DATE and TOS etc. Her research interests include computer architecture, interconnection networks, performance analysis, and data storage. She is a member of the IEEE.

Zhonghai Lu received the BS degree in radio and electronics from Beijing Normal University, Beijing, China, in 1989 and the MS degree in system-on-chip design and the PhD degree in electronic and computer system design from KTH Royal Institute of Technology, Stockholm, Sweden, in 2002 and 2007, respectively. From 1989 to 2000, he worked extensively as an engineer in the area of electronic and embedded systems. He is currently an associate professor in the Department of Electronics, KTH. He is the author of more 120 peer-reviewed papers. His research interests include interconnection networks, performance analysis, and real-time systems. He is a member of the IEEE.

Changsheng Xie received the BS and MS degrees in computer science from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 1982 and 1988, respectively. Presently, he is a professor in the Department of Computer Engineering, HUST. He is also the director of the Data Storage Systems Laboratory of HUST and the deputy director of the Wuhan National Laboratory for Optoelectronics. His research interests include computer architecture, disk I/O system, networked data storage system, and digital media technology. He is the vice chair of the expert committee of Storage Networking Industry Association (SNIA), China. He is a member of the IEEE.