Efficient Router Bypass via Hybrid Flow Control

Iván Pérez*, Enrique Vallejo* and Ramón Beivide*

*Deparment of Computer Science and Electronics Engineering
University of Cantabria
Santander, Cantabria, Spain
ivan.perezgallardo@unican.es

***Abstract*—Minimizing latency and power are key goals in the design of NoC routers. Different proposals combine lookahead routing and router bypass to skip the arbitration and buffering stages of their pipeline, reducing router delay to a single-cycle. However, the conditions to use the bypass are unnecessarily conservative, requiring completely empty buffers in the intermediate routers. This restricts the amount of flits that use the bypass, increasing average latency and power.**

This paper introduces *Non-Empty Buffer Bypass (NEBB)*, a mechanism that allows to bypass flits even if the buffers to bypass are not empty. The mechanism applies to wormhole and virtual-cut-through, each of them with different advantages. *NEBB-Hybrid* is proposed to employ the best flow control in each situation, maximizing the utilization of the bypass.

The proposals have been evaluated using Booksim. Results show up to 75% reduction of the buffered flits for single-flit packets, which translates into latency and dynamic power reductions of up to 30% and 23% respectively. For bimodal traffic, these improvements are 20% and 21% respectively. Additionally, the bypass utilization is largely independent of the number of VCs when using shared buffers and very competitive with few private ones, allowing to simplify the allocation mechanisms.

***Index Terms*—NoC, bypass router, NEBB, Hybrid**

I. Introduction

NoC latency has a clear impact on memory access time, and thus on the system performance. To minimize such latency, different mechanisms have been proposed to reduce the router pipeline stages, including lookahead routing [1] and router bypass [2]. Together, these mechanisms allow for a single-cycle router implementation, plus one cycle for link traversal. Additionally, bypass mechanisms reduce the use of buffers, which are the most power hungry components in the router. For these reasons, maximizing the utilization of the bypass pipeline is important for NoC latency and power reduction.

The bypass path is used when certain *bypass conditions* hold. The conditions used in previous proposals [2][3] guarantee that packets do not interleave in the same buffer (corrupting data). By construction, they also preserve order for packets sent in the same path and virtual channel (VC), even when they do not belong to the same flow. However, Message ordering is unnecessarily restrictive for bypass conditions, it is not a requirement of many coherence protocols and it is in fact not guaranteed with many other mechanisms such as adaptive routing, deflective routing or dynamic VC assignment.

This paper presents a new approach for the use of the bypass path to increase its utilization, denoted *Non-Empty Buffer Bypass (NEBB)*. With *NEBB*, bypassed packets can overtake buffered packets. *NEBB* can be implemented with different flow control mechanisms, and depending on the buffer occupancy, it is more efficient to employ Wormhole (WH) or Virtual Cut-Through (VCT). Based on this observation, we design *NEBB-Hybrid*, a hybrid mechanism that dynamically selects between WH or VCT forwarding in the bypass path, maximizing the amount of packets that use the shortcut. The mechanisms have been implemented using private or shared buffers. Specifically, the main contributions are:

- *NEBB*, a novel bypass mechanism compatible with different flow control mechanisms.
- *NEBB-Hybrid*, the main contribution of the paper, which dynamically forwards packets based on WH or VCT to maximize bypass utilization.
- A detailed evaluation which shows reductions up to 75% in buffer utilization that translates into latency and power reductions up to 30% and 23% respectively.

II. Background

A. LookAhead Bypass Router Architecture

We consider a typical baseline NoC router with several stages including Buffer Write (BW), Routing (R), VC and Switch Allocation (VA/SA), and Switch and Link Traversal (ST/LT).

Lookahead bypass routers short-cut the buffering (BW) and allocation (VA/SA) stages in the absence of flit conflicts; otherwise, the traditional (non-bypass) pipeline is used. The implementation relies on control packets, denoted *advance bundles*, or *LookAheads* (LA), which setup the bypass one cycle before the arrival of a flit. LAs are generated after flits win access to the crossbar. They are destroyed in the next router, after configuring the path or because of conflicts.

Fig. 1 depicts the router pipeline stages for flits (white) and LAs (red) based on the organization in [4]. Routing is implemented one hop in advance (*Lookahead Routing, LA-R*) in parallel with BW. Two consecutive routers $R0$ and $R1$ present the non-bypass and bypass pipelines, respectively.

This work was supported by the Spanish Ministry of Science, Innovation and Universities, FPI grant BES-2017-079971, the Spanish Ministry of Science, Innovation and Universities under contract TIN2016-76635-C2-2-R (AEI/FEDER, UE) and the European HiPEAC Network of Excellence. The Mont-Blanc project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 671697.

978-1-5386-8552-5/18/$31.00 ©2018 IEEE

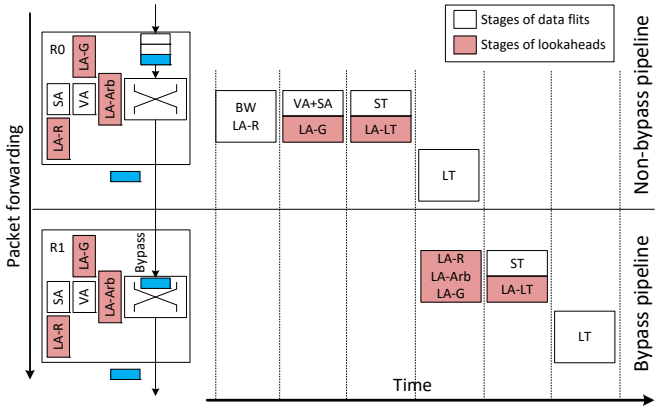
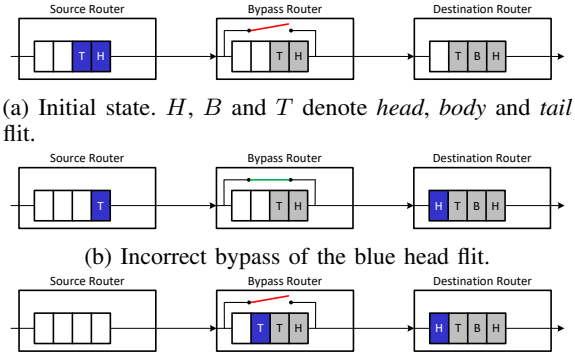


Fig. 1: Lookahead bypass router pipelines. Acronyms are defined in the text in Section II-A.



(a) Initial state. H, B and T denote *head*, *body* and *tail* flit.

(b) Incorrect bypass of the blue head flit.

(c) Blue tail flit is buffered at the *bypass router* because there is no room at the *destination router*, causing packet interleaving in the same buffer and data corruption.

Fig. 2: Incorrect packet bypass with buffer interleaving.

The functions associated to LAs are described next:

LookAhead Generation (LA-G) creates the control information for a flit and *Link Traversal (LA-LT)* sends it during its ST stage. In this way, LAs arrive to the next router one cycle before their flits.

LookAhead Arbiter (LA-Arb) handles conflicts between different LAs requesting the same output port. LAs must reserve the desired output port in order to set the bypass for the upcoming flit in the next cycle. Different implementations are considered: either no LA can proceed in case of conflict (no arbiter) or an arbiter per output port is used to select one winner (LA-Arb). In addition, when buffered flits in the SA/VA stages conflict with LAs, priority can be given to either buffered flits [2] or LAs [3], [4].

B. Lookahead Bypass Router Policies

In a Lookahead bypass router, the bypass path is used only if the following *bypass conditions* [2] are met:

1) The buffer at the input port that receives the LA is empty.
2) There is no output port conflict with buffered flits.
3) There are no conflicts between LAs arriving in the same cycle.

Condition 1 guarantees that packets do not interleave in the same buffer and are forwarded in order. With multiple VCs, this restriction applies to the buffer of the VC where the flit would be stored in case of using the non-bypass pipeline. Many proposals employ a large number of VCs [2], [3] to avoid limitations in the bypass from this condition. However, this requires a large buffer area and complicates the VC allocator, which typically sets the critical path delay of the router [5].

Condition 2 gives absolute priority to flits in the non-bypass pipeline, this is, those already stored in the pipeline buffers. Note that the opposite priority may also be considered to maximize the utilization of the bypass path.

Condition 3 implies that there is no arbiter between LAs: when multiple LAs contend for the same output, they are all discarded and the associated flits use the non-bypass pipeline.

Different implementations may modify slightly these conditions. In [3], [4] absolute priority is given to LAs over packets in the buffers (condition 2 is removed); and an arbiter per output port is implemented to select one wining LA in case of conflict for an output port (condition 3).

C. Wormhole and Virtual Cut-Through Flow Control

Flow control in NoCs is often implemented using wormhole (WH) flow control. WH forwards traffic on a flit-by-flit basis, based on the availability of space for each flit in the buffer of the next router. WH allows to reduce buffer size, since routers do not need to accommodate a complete packet.

However, to avoid unnecessary throttling buffers need to be sized for the buffer turnaround time, this is, the minimum idle time required for successive flits to reuse a buffer. A typical turnaround time is 4 to 6 cycles [5], [6], similar to typical NoC packet sizes in flits. Such buffering allows to implement Virtual-Cut Through (VCT) flow control, which forwards data on a packet-by-packet basis.

Many NoC designs employ link widths that accommodate a whole packet [7], [8]. In such case, packets are single-flit and WH and VCT behave equally. Other designs present bimodal traffic, with 1 and 5 flits being typical values [9]. For such cases, WH credit accounting is performed for each flit of the packet, whereas in VCT it is done for the whole packet.

III. Efficient Bypass Mechanisms

A. Non-Empty Buffer Bypass

Non-Empty Buffer Bypass (NEBB) performs bypass when the router buffer is not empty, as long as this does not interleave flits from different packets in a single VC. NEBB relaxes the bypass condition 1 introduced in Sect. II-B to the following two general conditions:

1a) No packet in the bypassed input VC is already advancing to an output port.
1b) The packet may be forwarded without packet interleaving in a buffer, according to the flow control employed.

Condition 1a is required to avoid that bypassed packets conflict with other packets that have already won allocation and have their status recorded in these VC control registers.

Condition 1b is required to prevent data corruption, and it is dependent on the flow control used. Fig. 2 shows an example of *incorrect* packet bypass using WH, to illustrate this requirement. There are three routers: *source*, *bypass* and *destination*[1], with packets stored in their buffers. The dark blue packet in the *source* router in Fig. 2a tries to bypass the intermediate router. The Head (H) flit is bypassed in Fig. 2b because there is a free buffer in the *destination* router buffer (following WH) and overtakes the packet in the non-empty buffer of the *bypass* router. The next flit (tail, T) cannot be bypassed because there is no more room in the *destination router* buffer, so it is stored in the buffer of the *bypass router* in Fig. 2c, behind the existing packet. In this situation, the grey packet is interleaved in the same VC with the blue packet, so data is corrupted. Even if the grey packet is forwarded to a different output, the blue tail flit has lost its routing and status information from the VC registers, so it cannot be forwarded.

The specific forwarding conditions for *NEBB* using WH or VCT are presented next. These conditions rely on the occupancy level of the buffers in the *bypass* and *destination* routers, and they are summarized in Table I.

1) NEBB-WH: Under WH, arbitration is performed flit by flit. The bypass of one flit does not guarantee that the following flits of the packet will be also bypassed, as presented in Fig. 2. For this reason, NEBB-WH forbids bypassing multi-flit packets when the *bypass router* buffer is not empty. By contrast, it can bypass single-flits packets, which is a frequent case. Therefore, condition 1 under WH flow control results as follows:

1a) No packet in the input buffer (VC) is already advancing to an output port.

1b) The packet is single-flit or the bypass buffer is empty.

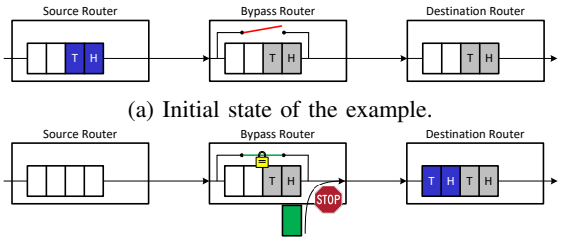
2) NEBB-VCT: Under Virtual Cut-Through, arbitration is performed once per packet and the assigned resources remain allocated for the duration of the packet forwarding, preventing any packet interleaving. To forward a packet VCT requires space for the whole packet at the destination buffer. This means that in NEBB-VCT multi-flit packets cannot be bypassed if the *destination router* buffer can only accommodate part of the packet. Additionally, the buffer in the *bypass router* also needs room to accommodate the whole packet, even if it is not used: otherwise, the *source router* would not start sending the packet. Therefore, condition 1 under VCT results as follows:

1a) No packet in the input buffer (VC) is already advancing to an output port.

1b) The bypass and destination buffers have room for the whole packet.

Fig. 3b illustrates the bypass of packets in VCT. As there is room for the whole packet in the *bypass* and *destination routers*, the blue packet can be bypassed, independently of the emptiness of the buffer in the *bypass router*. During packet bypass the resources are reserved (locked), so other buffered flits or lookaheads (e.g. the green flit in a different port) cannot obtain the output port that is using the blue packet, until the tail flit of the blue packet reaches the ST stage of *bypass router*.

[1]They do not necessarily refer to the source and destination of the packet.



(a) Initial state of the example.

(b) Bypass allowed with resource locking.

Fig. 3: NEBB VCT bypass. A packet is bypassed to the *destination router*, with room for the whole packet in the *destination* and a not empty buffer in the *bypass router*.

B. *Improved Bypass in NEBB via* Hybrid *Flow Control*

The *NEBB-Hybrid* mechanism (or simply *Hybrid*) dynamically selects between WH and VCT to maximize the utilization of the bypass. Sect. III-A and Table I present the limitations of *NEBB* when using each mechanism: *NEBB-WH* does not bypass multi-flit packets when the *bypass router* buffer is not empty; *NEBB-VCT* does not bypass multi-flit packets when the *bypass router* buffer cannot accommodate the whole packet. *Hybrid* selects the most suitable mechanism in each case.

In *Hybrid* the standard pipeline uses WH. The bypass pipeline uses both flow controls: if the buffer to bypass is empty, the router checks if there is room for a flit in the destination VC, following WH; otherwise, the router checks if there is room for the whole packet, following VCT. For single-flit packets, both mechanisms are equivalent.

Combining two flow control mechanisms introduces subtle complexities. *Hybrid* employs WH, so flits from different VCs may alternate in the same physical channel, generating a "hole" in the forwarding of the flits of a packet. Allocation in *Hybrid* is implemented flit by flit, to take advantage of such holes and maximize forwarding. However, the original NEBB-VCT forwarding for multi-flit packets is safe because it guarantees that the whole packet is forwarded consecutively (Sect. III-A2). To preserve this safeguard with flit by flit allocation and VCT, *Hybrid* employs variable priority arbiters in the LA Arbiter. Maximum priority is assigned to the LAs of multi-flit packets bypassed by VCT. For each output port, there can be at most one packet with maximum priority. The holes of this packet are leveraged to bypass flits from other packets, but only following WH (including single-flit packets), to avoid having to assign maximum priority to two packets in the same output port.

C. *Implementation Details of* Hybrid

1) Switch allocator: The switch allocator often employs a two-stage implementation [3], [4], first an arbiter for the inputs and then an arbiter for each output. Simple round-robin (RR) arbiters are often used. RR input arbiters cycle through all available VCs, selecting one at a time consecutively. If one of the output VCs is not available (for example, there are no credits in the destination VC) it does not proceed, wasting one cycle. However, such implementation simplifies the router design, since output availability does not need to be propagated

TABLE I: Bypass buffer conditions for different mechanisms. *bypass buffer* refers to the buffer in the bypass router being empty or not. *Dest. buffer* indicates if the destination buffer may accommodate the whole *packet* or only some flits (*partial*)

Bypass type (Required buffer size)	**Bypass buffer: empty**			**Bypass buffer: not empty**		
	Single-flit packet	**Multi-flit packet**		**Single-flit packet**	**Multi-flit packet**	
		Dest. buffer: packet	**Dest. buffer: partial**		**Dest. buffer: packet**	**Dest. buffer: partial**
VCT (packet size)	✓	✓	X	X	X	X
WH - Baseline (1 flit)	✓	✓	✓	X	X	X
NEBB-WH (1 flit)	✓	✓	✓	✓	X	X
NEBB-VCT (packet size)	✓	✓	X	✓	✓	X
NEBB-Hybrid (1 flit[2])	✓	✓	✓	✓	✓	X

to the inputs. Also, such implementation inherently multiplexes packet flits, generating packet holes in WH.

Holes are undesirable when VCT is also employed in the bypass in *Hybrid*, as discussed in Sect. III-B. To minimize them, we use a variable priority input arbiter, selecting the same VC until the packet tail flit is forwarded, similar to VCT. However, this might introduce performance and deadlock issues when WH and VCT are combined: First, a packet may be forwarded by WH without space at the destination buffer for the complete packet, introducing delays until the buffer becomes available. Second, this introduces a dependence between the input VCs, which generates a deadlock when occurring in multiple routers simultaneously. To avoid these issues, we give priority to body flits to minimize packet holes, but remove this priority when a flit does not advance, so the following ready VC is selected in the next arbitration cycle.

2) Shared buffers - credit management: Shared buffers [10], [11] improve efficiency. Shared buffer capacity accounting needs to consider the dual flow control. Packets following VCT have to reserve room for their size in advance. Otherwise, other packet advancing to another VC of the same input (using WH) may invade slots initially intended for the first packet.

The evaluations of this work use credits, so the reservation is done decrementing the credit count by the packet size when bypassing a packet via VCT, or flit by flit via WH.

IV. Methodology

We have implemented the router architecture described in Section II and the bypass schemes described in Section III in BookSim [12]. We model a 256-core network, arranged as an 8×8 mesh with concentration $c = 4$. The router employs combined allocators [13] similar to [3], [4] to balance pipeline stages. Priority is given to LAs over buffered flits. In the non-bypass pipeline, priority is given to body flits as presented in Section III-C1. Simulation parameters are shown in Table II.

We employ five bypass models. *Baseline* and *Baseline+Arb* are WH references without/with an arbiter between LAs. Additionally, we implement the three *NEBB* variants introduced in Section III: *NEBB-WH*, *NEBB-VCT* (which also employs VCT in the non-bypass pipeline) and *NEBB-Hybrid*.

Experiments use synthetic traffic, with single-flit packets or bimodal traffic. Bimodal traffic resembles a coherence protocol using packets of 1 (control) and 5 (data) flits. A single-flit packet ratio of 80% is used [9]. The traffic pattern employed

[2]Packets larger than the buffer size cannot be bypassed by VCT rules.

TABLE II: Default simulation parameters.

Topology	8×8 mesh, concentration $c = 4$
Link latency	1 cycle
Router architecture	2/4-stage bypass router
Router size	8 ports (4 transit, 4 injection/ejection)
Buffer implementation	Shared (DAMQ, [10])
Buffer size	12 flits (1 private flit per VC)
Num. VCs	2
Packet size	1 and 5 flits
Routing	DOR
SA input arbiters	8 Round Robin arbiters, #VCs:1
SA output arbiters	8 Matrix arbiters, 8:1
LA arbiters	8 Matrix arbiters, 8:1
VA policy	Highest number of credits
Frequency	1 GHz
Technology	Tri-Gate 11nm LVT process
Channel width	128 bits

is random uniform, but we also evaluate bit-reversal, transpose and hotspot (with hotspots in nodes 0, 15, 240 and 255). We measure relevant metrics such as average packet latency, dynamic power, and percentage of buffered flits. The latter divides the amount of times flits are buffered by the total number of times a flit is forwarded (averaged for all flits).

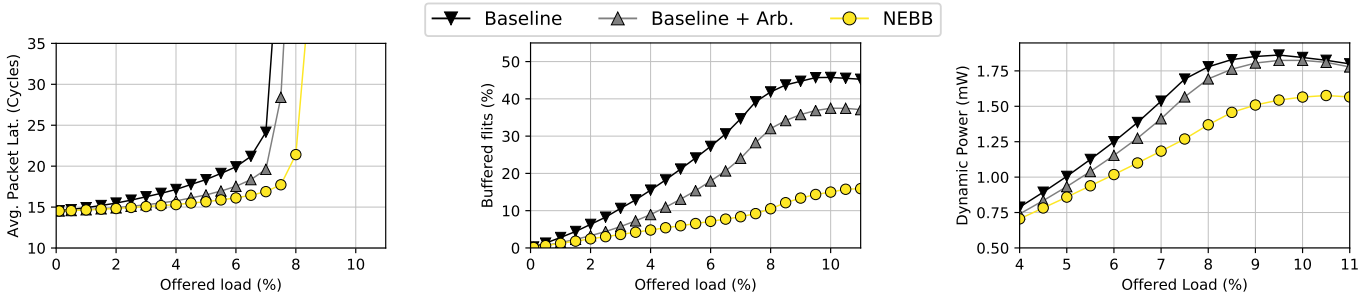
Dynamic power results are obtained using DSENT [14]. We have implemented a model of the bypass router based on the default four-stage router model of DSENT. The dynamic power of the buffers and allocators from DSENT is multiplied by the ratio of buffered flits over all the received flits per router. The LA arbiters employed are equal to the arbiters in the second stage of the switch allocator. Therefore, the LA arbiters power equals the power of the second stage of the switch allocator provided by DSENT. We use no correction factor because these arbiters are used for every LA, and one LA is received for each flit. We estimate that the extra control logic (such as the checks of the packet size, the occupancy of the buffer to bypass, etc) is negligible compared to the consumption of the buffers, crossbar or arbiters.

V. Results

A. NEBB using Single-Flit Packets

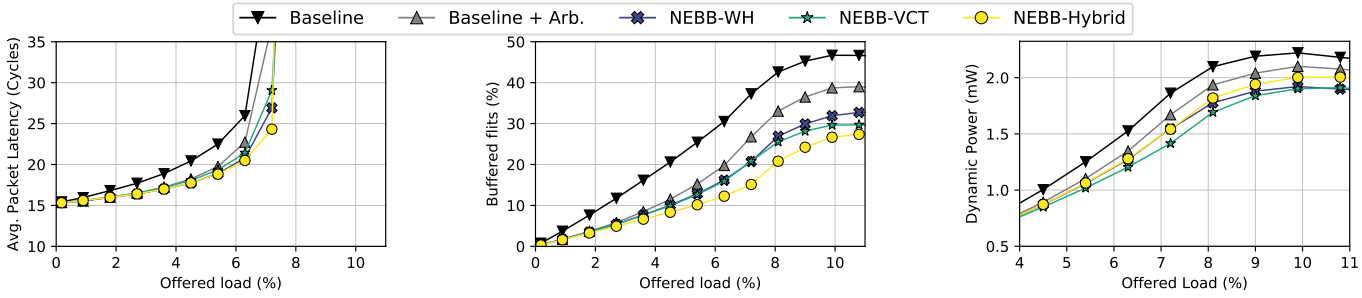
Fig. 4 compares packet latency, buffered flits and dynamic power; in all cases, lower results are better. These results use single-flit traffic, so all *NEBB* variants are equivalent. 6 slots per shared buffer are used, adapted to the small packet size.

The amount of buffered flits in 4b grows with the network load. *Baseline* buffers flits when the buffers are non-empty or there are LA conflicts (all the conflicting LAs are discarded). The *Baseline+Arb* model is similar, but one LA proceeds in



(a) Packet latency. (b) Buffered flits. (c) Router dynamic power.

Fig. 4: 8×8 $c = 4$ mesh performance with uniform random single-flit traffic and buffer size of 6 flits.



(a) Packet latency. (b) Buffered flits. (c) Router dynamic power.

Fig. 5: 8×8 $c = 4$ mesh performance with uniform random and bimodal traffic

case of conflicts, reducing the use of buffers. In *NEBB*, buffers are only used when conflicts occur, not because of non-empty buffers, minimizing the buffer utilization. This translates into latency and power savings, particularly at intermediate loads. At 7% load (0.07 flits/node/cycle), *NEBB* reduces *Baseline's* latency in 30.1% and buffered flits in 75.9%. From these values, 18.8% and 30.7% respectively come from the LA arbiter, as observed in *Baseline+Arb* results. Regarding dynamic power, *NEBB* saves 23.0% over *Baseline* at 7% load.

B. NEBB *Flow Control and* Hybrid

Fig. 5 compares the *NEBB* alternatives using bimodal traffic. The three *NEBB* variants outperform the baselines, and *Hybrid* presents the best results since it maximizes the cases in which bypass is used.

Both *NEBB-WH* and *NEBB-VCT* present similar results. VCT has slightly higher latency and lower throughput, which translate into slightly lower power results after saturation.

NEBB-Hybrid has the best results in latency, buffered flits and dynamic power with a reduction of 20.6%, 60.1% and 21.1%, respectively, over *Baseline* at a load around 6%.

Fig. 6 depicts the buffer utilization for different traffic patterns. The results are similar to the previous ones with uniform random traffic, with *NEBB* mechanisms improving the utilization of the bypass and *Hybrid* being the optimal version.

C. Buffer depth and number of VCs

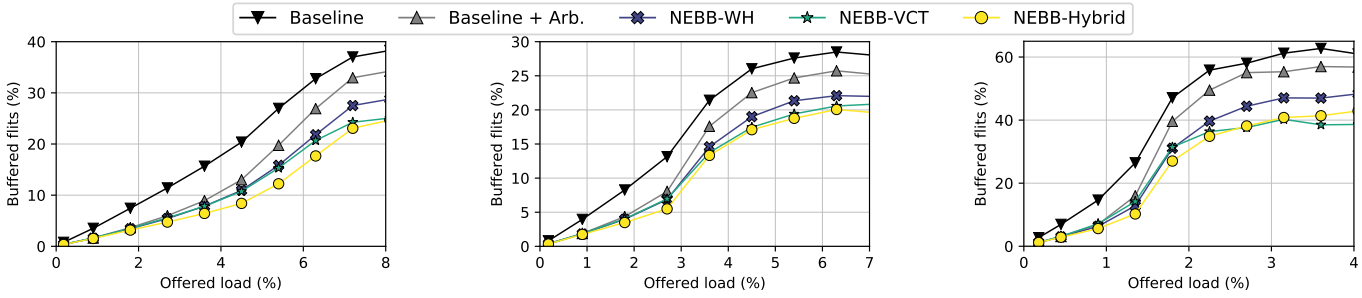
Fig. 7 depicts the buffer utilization for *Baseline+Arb* and *Hybrid* with different combinations of VCs and total buffer sizes. Each plot represents the same buffer space with variable number of VCs, either shared (7a and 7b) or private buffers per VC (7c and 7d). With shared buffers, *Hybrid* clearly outperforms *Baseline+Arb*, particularly when the shared buffer size is not very small. With 20 flits per port, no amount of VCs in *Baseline+Arb* matches the result of *Hybrid*. The amount of VCs used in *Hybrid* has a small impact on buffered flits.

In the private buffers evaluations in 7c and 7d the total amount of storage increases with the VC count. If buffers are very small (7c, buffer per VC equals the maximum packet size of 5 flits) *Hybrid* is better than *Baseline+Arb* for the same number of VCs, but the improvement is modest. Indeed, this is the minimum buffer size for *Hybrid* to use VCT. With larger buffers in 7d, the results of *Hybrid* with half the VCs approximately match the result of *Baseline+Arb* before saturation, and get better after this point.

VI. Related Work

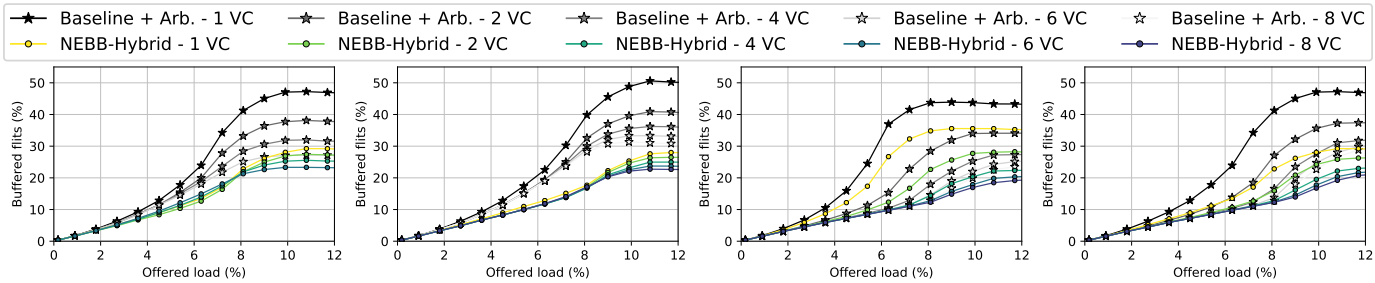
Sections I and II have already presented LookAhead [1] and bypass [2] mechanisms. Token Flow Control [3] sends information about the availability of resources among nodes in a neighborhood. The objective of the mechanism is the improvement of the bypass utilization by choosing low congested paths, exploiting path diversity with adaptive routing.

Our *Hybrid* approach combines two flow controls, WH and VCT. Whole Packet Forwarding (WPF, [9]) applies packet-based flow control in a WH network, but they do it to relax VC re-allocation requirements in deadlock-free fully adaptive routing NoCs, without considering bypass. In [15]



(a) Bit-reversal. (b) Transpose. (c) Hotspot.

Fig. 6: Buffered flits in an 8×8 $c = 4$ mesh for different traffic patterns, using bimodal traffic.



(a) DAMQ. Total size: 10 flits. (b) DAMQ. Total size: 20 flits. (c) Private VC. VC size: 5 flits. (d) Private VC. VC size: 10 flits.

Fig. 7: Buffer utilization for a mesh with different number of VCs and buffer sizes using bimodal traffic.

they suggest using two different types of flow control, buffered and unbuffered, but again without considering bypass.

SMART [16] extends bypass to skip multiple routers in a single cycle, multiplying the savings in latency and power. The bypass conditions need to be satisfied in all the routers in the path. For this reason, we believe that our *Hybrid* bypass can be applied to such designs. A study of multihop bypass based on *Hybrid* is left for future work.

VII. Conclusions

Bypass reduces packet latency and power consumption, which are key aspects of NoC designs. The Non-Empty Buffer Bypass proposal is based on a proper analysis and relaxing of the original bypass conditions. Variants of *NEBB* following WH and VCT rules are introduced, and the combination of them, denoted *Hybrid*, maximizes the utilization of the bypass.

We show the effectiveness of *NEBB* and *Hybrid*. Our proposals decrease by up to 30% packet latency and up to 23% dynamic power savings. Additionally, results show that *Hybrid* outperforms prior proposals with shared buffers, and requires half the VCs for the same result with private buffers, which simplifies the VC allocation. Altogether, these results present *Hybrid* as a competitive and cost-effective alternative to improve the design and performance of the NoC bypass.

References

[1] M. Galles, "Spider: a high-speed network interconnect," *IEEE Micro*, vol. 17, pp. 34–39, Jan 1997.

[2] A. Kumar, P. Kunduz, A. Singhx, *et al.*, "A 4.6 Tbits/s 3.6 GHz single-cycle NoC router with a novel switch allocator in 65nm CMOS," in *ICCD*, 2007.

[3] A. Kumar, L.-S. Peh, and N. K. Jha, "Token flow control," in *MICRO*, pp. 342–353, 2008.

[4] T. Krishna, J. Postman, C. Edmonds, *et al.*, "SWIFT: A SWing-reduced Interconnect For a Token-based network-on-chip in 90nm CMOS," in *ICCD*, 2010.

[5] N. E. Jerger, T. Krishna, and L.-S. Peh, *On-Chip Networks, Second Edition*, vol. 12. 2017.

[6] D. U. Becker, N. Jiang, G. Michelogiannakis, and W. J. Dally, "Adaptive backpressure: Efficient buffer management for on-chip networks," in *ICCD*, 2012.

[7] A. Kumar, "The new Intel Xeon processor scalable family (formerly Skylake-SP)," in *Hot Chips*, 2017.

[8] S. Davidson, S. Xie, C. Torng, *et al.*, "The Celerity open-source 511-core RISC-V tiered accelerator fabric: Fast architectures and design methodologies for fast chips," *IEEE Micro*, pp. 30–41, Mar 2018.

[9] S. Ma, N. E. Jerger, and Z. Wang, "Whole Packet Forwarding: Efficient design of fully adaptive routing algorithms for networks-on-chip," in *HPCA*, 2012.

[10] Y. Tamir and G. L. Frazier, "Dynamically-allocated multi-queue buffers for vlsi communication switches," *TC*, vol. 41, pp. 725–737, Jun 1992.

[11] C. A. Nicopoulos, D. Park, J. Kim, N. Vijaykrishnan, *et al.*, "ViChaR: A dynamic virtual channel regulator for network-on-chip routers," in *MICRO*, 2006.

[12] N. Jiang, J. Balfour, D. U. Becker, *et al.*, "A detailed and flexible cycle-accurate network-on-chip simulator," in *ISPASS*, pp. 86–96, 2013.

[13] A. Psarras, I. Seitanidis, C. Nicopoulos, and G. Dimitrakopoulos, "Short-path: A network-on-chip router with fine-grained pipeline bypassing," *TC*, vol. 65, pp. 3136–3147, Oct 2016.

[14] C. Sun, C.-H. O. Chen, G. Kurian, *et al.*, "DSENT-a tool connecting emerging photonics with electronics for opto-electronic networks-on-chip modeling," in *NoCS*, 2012.

[15] S. A. R. Jafri, Y. J. Hong, M. Thottethodi, and T. N. Vijaykumar, "Adaptive flow control for robust performance and energy," in *MICRO*, pp. 433–444, 2010.

[16] C. H. O. Chen, S. Park, T. Krishna, *et al.*, "SMART: A single-cycle reconfigurable NoC for SoC applications," in *DATE*, pp. 338–343, March 2013.