

Improving Routing Efficiency for Network-on-Chip through Contention-Aware Input Selection

Dong Wu, Bashir M. Al-Hashimi, Marcus T. Schmitz

School of Electronics and Computer Science

University of Southampton

Southampton, SO17 1BJ, UK

e-mail: {dw, bmah, ms4}@ecs.soton.ac.uk

Abstract - The performance of Network-on-Chip (NoC) largely depends on the underlying routing techniques, which have two constituencies: *output selection* and *input selection*. Previous research on routing techniques for NoC has focused on the improvement of output selection. This paper investigates the impact of input selection, and presents a novel *contention-aware input selection (CAIS)* technique for NoC that improves the routing efficiency. When there are contentions of multiple input channels competing for the same output channel, CAIS decides which input channel obtains the access depending on the *contention level* of the upstream switches, which in turn removes possible network congestion. Simulation results with different synthetic and real-life traffic patterns show that, when combined with either deterministic or adaptive output selection, CAIS achieves significant better performance than the traditional first-come-first-served (FCFS) input selection, with low hardware overhead (<3%).

I Introduction

As technology scales and chip integrity grows, on-chip communication is playing an increasingly dominant role in System-on-Chip (SoC) design. To meet the performance and design productivity requirements, Network-on-Chip (NoC) [1-4] has been proposed as a solution to provide better modularity, scalability, reliability and higher bandwidth compared to bus-based communication infrastructures. Fig. 1(a) shows a mesh-based NoC, which consists of a grid of 16 cores. Each core is connected to a switch by a network interface. Cores communicate with each other by sending packets via a path consisting of a series of switches and inter-switch links. For each packet, there are several possible paths, which directly influence the time needed for delivery. Therefore, the performance of NoC largely depends on the underlying routing technique, which chooses a path for a packet and decides the routing behaviour of the switches.

Fig. 1(b) shows a block diagram of a switch with $n+1$ input channels and output channels interconnected by a crossbar. In order to route packets through the network, the switch needs to implement a routing technique. A routing technique has two constituencies: output selection and input selection. A packet coming from an input channel may have a choice of multiple output channels, e.g., a packet p_0 of input_0 can be forwarded via output_0, output_1 and so on. The *output selection* chooses one of the multiple output channels to deliver the packet. Similarly, multiple input

channels may request simultaneously the access of the same output channel, e.g., packets p_0 of input_0 and p_1 of input_1 can request output_0 at the same time. The *input selection* chooses one of the multiple input channels to get the access.

Whilst the impact of the output selection on routing efficiency has been investigated [5-7], no explicit work has been reported on the impact of the input selection, which is the aim of this paper. The main contribution of this paper is a novel *contention-aware input selection (CAIS)*, as part of the routing techniques implemented in switches. With CAIS, each output channel within a switch observes the *contention level* (i.e., the number of request from the input channels, Section III), and transmits this contention level to the input channel of the downstream switch. During the input selection within the downstream switch, CAIS chooses an input channel depending on the contention levels. Input channels with higher contention levels get higher priority. CAIS tries to remove possible network congestion by keeping the traffic flowing even in the paths with heavy traffic load, which in turn improves routing efficiency.

Experimental results with synthetic and real-life examples show that CAIS can be combined with either deterministic or adaptive output selection, and it is capable of decreasing

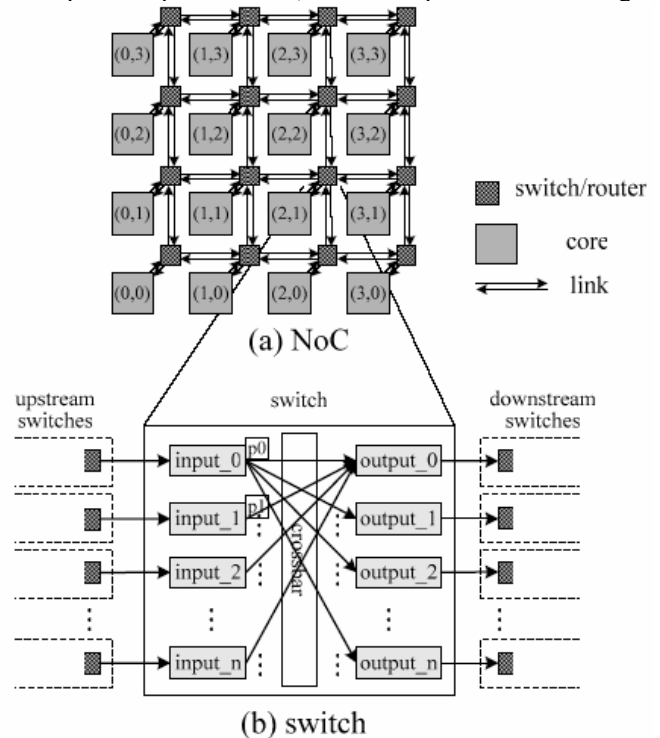


Fig. 1. Block diagram of NoC and switch

* This work is supported in part by the EPSRC, UK, under grant EP/C512804, GR/S95770.

packet latency significantly compared to the traditional first-come-first-served (FCFS) input selection. Furthermore, the synthesis of prototype switches with CAIS shows low hardware overhead compared to FCFS.

The rest of paper is organised as follows. Related work is reviewed in Section II. Section III describes the proposed contention-aware input selection (CAIS) in detail. The experiment results and switch implementation are presented in Section IV. Finally Section V gives the conclusion.

II. Related Work

The idea of NoC is derived from large-scale computer networks and distributed computing [8, 9]. However, the routing techniques for NoC have some unique design considerations besides low latency and high throughput. Due to tight constraints on memory and computing resources, the routing techniques for NoC should be reasonably simple [7]. Several switch architectures have been developed for NoC [10-12], employing XY output selection and wormhole routing (Section III). In [6], a deflective routing technique is proposed to avoid network congestion by spreading the traffic over a larger area. It performs output selection based on the number of packets being handled in the neighbouring switches. Packets are forwarded to switches with less traffic load. The routing technique proposed in [7] is similar to [6] in terms of acquiring information from the neighbouring switches to avoid network congestion, but uses the buffer levels of the downstream switches to perform the output selection. A routing scheme which combines deterministic and adaptive routing is proposed in [5], where the switch works in deterministic mode when the network is not congested, and switches to adaptive mode when the network becomes congested. All the routing techniques [5-7] focused on the output selection. The motivation of this paper is to investigate the impact of input selection and develop a simple yet effective input selection, aiming to improve the routing efficiency with low hardware cost.

III. Proposed Technique

Two input selections have been used in NoC, first-come-first-served (FCFS) input selection [5] and round-robin input selection [10, 11]. In FCFS, the priority of accessing the output channel is granted to the input channel which requested the earliest. Round-robin assigns priority to each input channel in equal portions on a rotating basis. FCFS and round-robin are fair to all channels but do not consider the actual traffic condition. This section presents a contention-aware input selection (CAIS) as part of the routing techniques implemented in switches. CAIS performs more intelligent input selection by considering the actual traffic condition, leading to higher routing efficiency.

A. Preliminaries

In this paper we consider NoCs with 2D mesh topology (Fig. 1(a)). Wormhole switching [9, 13] is employed because of its low latency and low buffer requirement. In wormhole switching, a packet is divided into flits for transmission. The header flit contains the routing information, which is used by

the switches to establish the routing path. The remaining flits simply follow the path in a pipeline fashion. A flit is passed to the next switch as soon as enough buffer space is available to store it, even though there is not enough space to store the whole packet. If the header flit encounters a channel already in use, the subsequent flits have to wait at their current locations and are spread over multiple switches, thus blocking the intermediate links.

The proposed contention-aware input selection (CAIS, Section III.B) has been combined with an output selection, either deterministic or adaptive [5], to complete the routing function. In this paper, the XY routing [9] is used as a representative of deterministic output selection for its simplicity and popularity in NoC. In the XY output selection, packets are sent first along the X dimension then along the Y dimension. For example, considering the NoC of Fig. 1(a), a packet from (0, 3) to (2, 2) will take a path as follows: (0, 3) \rightarrow (1, 3) \rightarrow (2, 3) \rightarrow (2, 2). The wormhole switching is sensitive to deadlock [9, 13]. To avoid deadlock, the minimal odd-even (OE) routing [8] is used as a representative of adaptive output selection. In the OE output selection, a packet chooses a path from multiple alternatives, but paths with certain turns are prohibited to avoid deadlock. Considering the previous example again, a packet from (0, 3) to (2, 2) has two alternative paths: (0, 3) \rightarrow (0, 2) \rightarrow (1, 2) \rightarrow (2, 2) and (0, 3) \rightarrow (1, 3) \rightarrow (1, 2) \rightarrow (2, 2). Note the path (0, 3) \rightarrow (1, 3) \rightarrow (2, 3) \rightarrow (2, 2) is invalid because an east-south turn is not allowed in the switch positioned at (2, 3) to avoid deadlock.

B. Contention-Aware Input Selection

To show the influence of input selection and output selection on the routing efficiency, consider the example of Fig. 2, which shows a network of switches (cores are ignored for simplicity). Note the grey scale of the switches indicates the number of packets waiting at the switches. The white colour switches have low number of waiting packets, whilst the grey colour switches have higher number of waiting packets, and the black colour switch at (2, 2) has the highest number of waiting packets. To demonstrate the influence of output selection, consider a packet p0 traveling from (3, 0) to (0, 2), which has a choice of multiple paths. A good path would be to avoid the congested area (i.e., the grey and black switches), as indicated by the dashed line. This shows a suitable output selection can avoid network

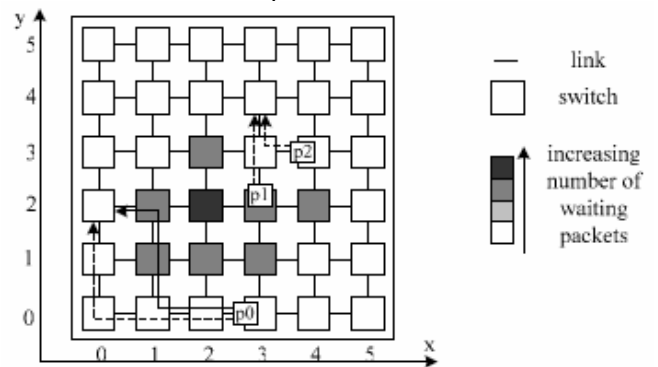


Fig. 2. Motivation of CAIS

congestion. Now consider the input selection. Packets p_1 at (3, 2) and p_2 at (4, 3) both want to travel through (3, 3). In this case, a good choice would be let p_1 take the priority to access (3, 3), because the switch at (3, 2) has more waiting packets than the switch at (4, 3). Such an input selection helps reduce the number of waiting packets in congested areas. This removes possible network congestions and leads to better NoC performance. Based on this observation, a contention-aware input selection (CAIS) is developed.

The basic idea of CAIS is to give the input channels different priorities of accessing the output channels. The priorities are decided dynamically at run-time, based on the actual traffic condition of the upstream switches. More precisely, each output channel within a switch observes the contention level (the number of requests from the input channels) and sends this contention level to the input channel of the downstream switch, where the contention level is then used in the input selection. When multiple input channels request the same output channel, the access is granted to the input channel which has the highest contention level acquired from the upstream switch. This input selection removes possible network congestion by keeping the traffic flowing even in the paths with heavy traffic load, which in turn improves routing performance.

Fig. 3 illustrates the detailed architecture of a switch with CAIS. As can be seen, besides wires for data transmission, CAIS requires additional wires to transmit contention levels (CLs) between neighbouring switches. The switch has $n+1$ input channels and output channels. Input channels contain a buffer to store the incoming flits temporarily before forwarding them to one of the output channels. The output selection (OS) module examines the header flit and decides to which output channel the packet should be passed. The OS sends an access request to the corresponding output channel. In an output channel, once it becomes available, the CAIS module examines the access requests, and sends a selection signal to the MUX module which accordingly

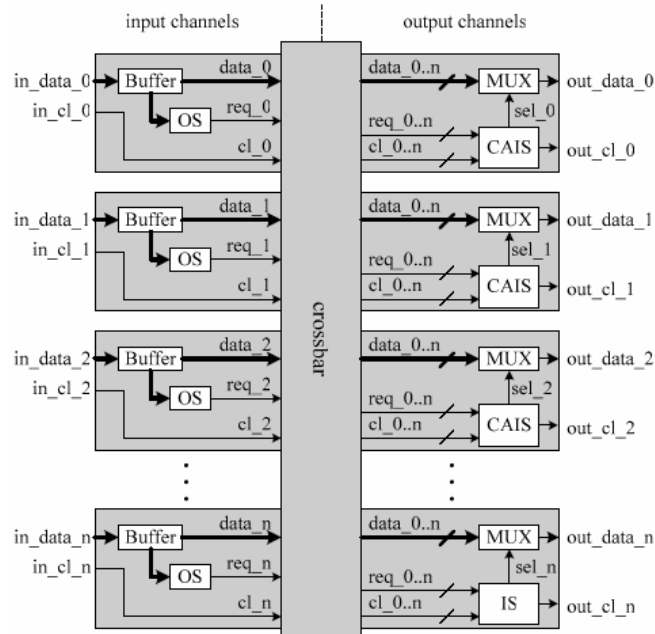


Fig. 3. Switch architecture with CAIS

connects one of the input data signals to the output data signal. If there is only one access request, the request is granted. If there are multiple access requests, a selection of which input channel gets the access has to be made. This selection mechanism is explained next.

CAIS performs input selection based on the contention level (CL). The contention level of an output channel is the number of access request received at a certain time. The CL of an input channel is acquired from the output channel of the upstream switch through signal wires. Fig. 4 shows the algorithm of CAIS, which consists of two processes working in parallel. Process *observe_cl* is activated when the status of $req_{0..n}$ changes. It observes the number of request to this output channel (i.e., CL) and puts the CL value at out_cl_i . Then the CL is transmitted to the input channel of the downstream switch, where the CL will be used to perform the input selection. For example, considering the switch of Fig. 3, if input channels 0 and 1 are requesting output channel 0, then the CL of output channel 0 (out_cl_0) is 2, and the CL of the input channel of the downstream switch is also 2. Note that, to avoid high complexity and hardware cost, CL is only sent to the immediate downstream switches and is not spread any further. For example, considering the previous example again, if the CLs of the input channels 0 and 1 (in_cl_0 and in_cl_1) are 3 and 4 respectively, the CL of output channel 0 is NOT 3+4, but 2. Process *select_input* is activated when the output channel is available and there are requests. It examines all requests and the CLs of the corresponding input channels, and grants the request with the highest CL.

For the input channels connected to the cores, there are no upstream switches transmitting CL to them. The CL value is set to 0 for these input channels. Therefore, the packets already in the network have higher priority than the packets waiting to be injected into the network.

Contention-Aware Input Selection (CAIS)	
$req_{0..n}$	request signals from the input channels
out_cl_i	CL of the i^{th} output channel
in_cl_j	CL of the j^{th} input channel acquired from the upstream switch
max_cl	maximum contention level
sel_i	selection signal of the i^{th} output channel
$i = 0, 1, 2, \dots, n; \quad j = 0, 1, 2, \dots, n;$	
01	process <i>observe_cl</i> ($req_{0..n}$)
02	begin
03	$out_cl_i \leq$ number of request to the i^{th} output channel;
04	end process <i>observe_cl</i>
05	
06	process <i>select_input</i>
07	begin
08	$max_cl := 0;$
09	for all requests loop
10	if $in_cl_j \geq max_cl$ then
11	$max_cl := in_cl_j;$
12	$sel_i \leq j;$
13	end if
14	end loop
15	end process <i>select_input</i>

Fig. 4. Pseudo VHDL code of the CAIS algorithm

IV. Experimental Results

Experiments are conducted to evaluate the performance of the contention-aware input selection (CAIS) and give a comparison between CAIS and traditional input selections. Two traditional input selections have been used in NoC, first-come-first-served (FCFS) input selection [5] and round-robin input selection [10, 11]. Due to the advancement of FCFS over round-robin, FCFS is selected to compare with CAIS. Both CAIS and FCFS are combined with a deterministic output selection (XY routing [9]) and an adaptive output selection (OE routing [8]). Four switch models are developed using VHDL to implement the four routing schemes: XY+FCFS, XY+CAIS, OE+FCFS and OE+CAIS. Simulations are carried out on a 6×6 mesh NoC using these four switch models. As in previous work [5, 8, 14], the performance of the routing scheme is evaluated through latency-throughput curves. For a given packet injection rate (i.e., the number of packets injected to the network per cycle), a simulation is conducted to evaluate the average packet latency. It is assumed that the packet latency is the duration from the time when the first flit is created at the source core, to the time when the last flit is delivered to the destination core. For each simulation, the packet latencies are averaged over 50,000 packets. Latencies are not collected for the first 5,000 cycles to allow the network to stabilise. It is assumed that the packets have a fixed length of 5 flits and the buffer size of input channels is 5 flits. Since the network performance is greatly influenced by the traffic pattern, we applied four different traffic patterns, including three synthetic traffic patterns (uniform, transpose and hot spot) and a real-life traffic pattern (GSM voice CODEC).

A. Synthetic Traffic

In the first set of experiments we consider three synthetic traffic patterns: uniform, transpose, and hot spot [8]. In the uniform traffic pattern, a core sends a packet to any other cores with equal probability. In the transpose traffic pattern, a core at (i, j) only send packets to the core at (5-j, 5-i). In the hot spot traffic pattern, the core at (3, 3) is designated as the hot spot, which receives 10% more traffic in addition to the regular uniform traffic.

Fig. 5 shows the performance of the four routing schemes under uniform traffic. The X-axis represents the packet injection rate per node (the packet injection rate for the whole NoC is 36 times higher), and the Y-axis represents the average packet latency. As can be seen from the figure, the four schemes have almost the same performance at low traffic load (<0.038 packets/cycle). As the traffic load increases, the packet latency rises dramatically due to the network congestion. Comparing the curves of OE+FCFS and OE+CAIS, it can be seen that, using the OE output selection, CAIS performs significantly better than FCFS. Similarly, the curves of XY+FCFS and XY+CAIS show that CAIS also outperforms FCFS when using XY output selection. As reported in [5, 8], the XY output selection has better performance than the OE output selection. This is because the XY output selection incorporates global, long-term information about the uniform traffic, leading to

even distribution of traffic. On the other hand, the OE routing is based on local, short-term information, which only benefits the immediate future packets while loses the evenness of uniform traffic in the long run.

Fig. 6 shows the performance of the four routing schemes under transpose traffic. It can be seen that FCFS and CAIS have the same performance when using the XY output

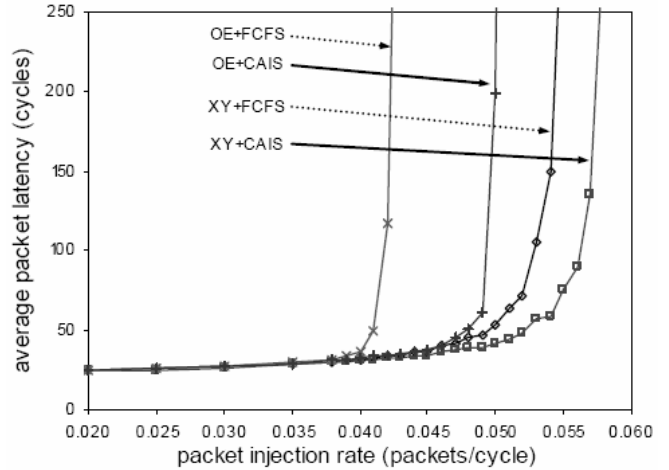


Fig. 5. Routing performance under uniform traffic

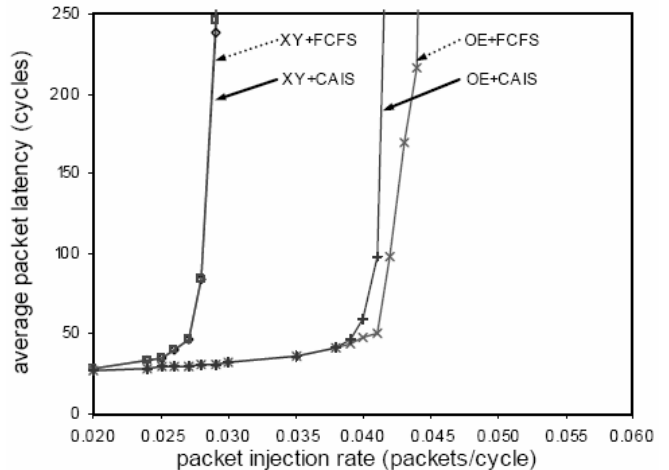


Fig. 6. Routing performance under transpose traffic

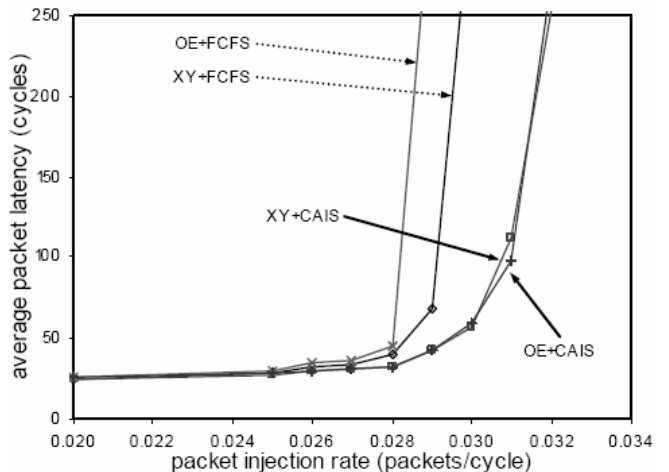


Fig. 7. Routing performance under hot spot traffic

selection; FCFS works slightly better than CAIS when using the OE output selection. This is because with transpose traffic, it is rarely the case that more than one input channels compete for the same output channel. Therefore, the input selection policy has little impact on the routing performance.

Fig. 7 shows the routing performance under hot-spot traffic. Once again, it can be seen that CAIS significantly outperforms FCFS, either using XY or OE output selection. Furthermore, although the OE output selection performs worse than the XY output selection when using the FCFS input selection, it achieves similar performance as XY when using the CAIS input selection.

B. Voice CODEC Traffic

To evaluate the performance of CAIS under more realistic traffic loads, we have conducted simulations using a GSM voice CODEC [15]. The GSM voice CODEC is partitioned into 9 cores. The communication trace between the cores is recorded for an input voice stream of 500 frames (10 seconds of voice). The cores are mapped manually to a 6x6 mesh NoC. Fig. 8 shows the partition and communication trace of the GSM voice CODEC. As can be seen, the encoder and decoder are partitioned into core_0 – core_4 and core_5 – core_8 respectively. The directed edges between the cores represent communications. For example, the edge between core_0 and core_1 means there is a communication from core_0 to core_1, the communication has a size of 320 bytes and occurs at cycle 0. Due to the space limitation, only communications in the first 1000 cycles are shown. Fig. 8 also shows the mapping of the cores to the NoC using the dashed lines. The CODEC cores generate packets according to the recorded communication trace. The other cores in the NoC generate uniform traffic, with the same average packet injection rate as the CODEC cores. The packet injection rate is increased incrementally to get the latency-throughput curve, which is shown in Fig. 9. As can be seen, in both cases of using XY and OE output selection, CAIS achieves better performance than FCFS. Furthermore, although XY has worse performance than OE when using FAFS, it achieves similar performance as OE when the CAIS input selection is used. This observation and the one obtained from Fig. 7 show that the employment of

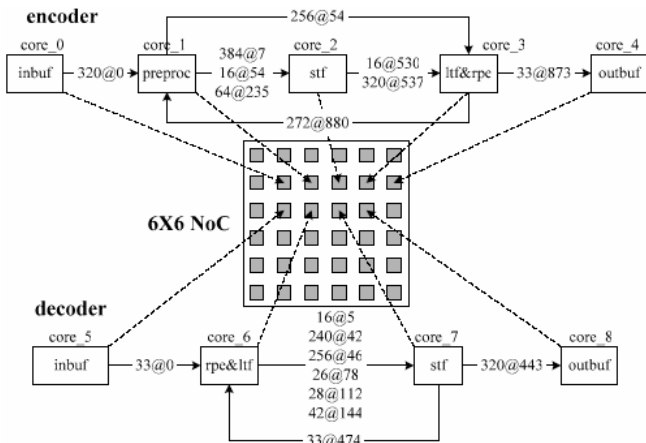


Fig. 8. Partition, communication trace and core mapping of a GSM voice CODEC

CAIS can help the otherwise less effective output selection when using FCFS catch up with the more effective output selection. This demonstrates further the importance of input selection in routing efficiency and the effectiveness of CAIS.

One consideration of CAIS is that some packets may experience indefinite waiting. Although theoretically possible, it does not happen in the experiments. To give an insight, Fig. 10 shows the maximum packet latency of the routing schemes under the GSM voice CODEC traffic. Note the curves in Fig. 10 are NOT average latency, but the maximum latency experienced by the packets, thus the curves show some dips and jumps. As can be seen, when the network load is low, CAIS has similar maximum packet latency as FCFS; when the network load is high, CAIS has shorter maximum packet latency than FCFS. CAIS does not cause indefinite waiting. The reason is that, due to the latency introduced by the processing of the header flit, there is a gap between the access requests of two consecutive packets. During this gap, one of the packets waiting in input channels with low contention levels can get the access to the output channel, thus indefinite waiting is avoided.

Overall, the experiments of Sections IV.A and IV.B have demonstrated the importance of input selection, which is in line with that obtained in [14] when applied in distributed computing. The experiments have also shown that CAIS

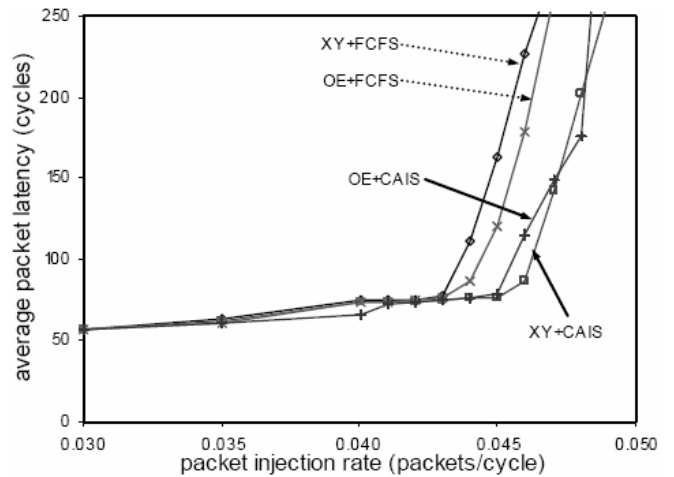


Fig. 9. Routing performance under voice CODEC traffic

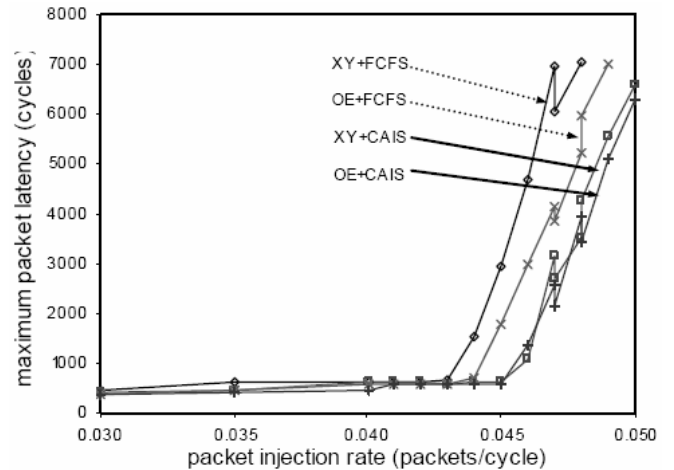


Fig. 10. Maximum packet latency under CODEC traffic

effectively improves the routing efficiency for NoCs.

C. Implementation of Prototype Switch

To evaluate the area overhead of CAIS and show the performance/area trade-off, switches with four different routing schemes have been implemented. The first scheme is XY+FCFS, i.e., XY output selection and FCFS input selection. The other three schemes are XY+CAIS, OE+FCFS and OE+CAIS. The switches were coded in VHDL and synthesized with Synplify ASIC using an ST Microelectronics 0.12 μm standard cell library. For all switches, the data width is set to 32 bits, and each input channel has a buffer size of 5 flits. Fig. 11 shows the area cost of the four switches. As expected, using the deterministic XY output selection and the simple FCFS input selection, XY+FCFS has the lowest area cost of 0.109725mm^2 . Due to the relative complexity of the adaptive OE output selection and the CAIS input selection, XY+CAIS and OE+FCFS have slightly higher area costs of 0.111480mm^2 and 0.110355mm^2 respectively, and OE+CAIS has the highest area cost of 0.113580mm^2 . Comparing the area costs of XY+FCFS and XY+CAIS, CAIS introduces 1.6% additional overhead than FCFS. Similarly, when comparing OE+FCFS and OE+CAIS, CAIS introduces 2.9% additional overhead than FCFS.

CAIS requires additional wires to transmit the contention levels (CLs). In the case of 2D mesh NoC, each switch have at most 5 input channels, receiving packets from the core and the 4 neighbouring switches. Thus at most $3 \lceil \log_2 5 \rceil$ wires are needed to transmit a CL. This is acceptable because NoC have abundant wiring resources [3, 4].

Note although this paper has considered mesh-based NoC, CAIS is flexible enough to support other NoC topologies including irregular topologies. This can be easily done by configuring the value of n (Fig. 3 and Fig. 4), and the number of wires for CL transmission accordingly.

V. Conclusion

This paper has shown the importance of input selection in routing efficiency, and presented a simple yet effective contention-aware input selection (CAIS) as part of the routing techniques implemented in switches. CAIS performs the input selection considering the contention level of the upstream switches. By granting busier input channel higher priority to access the output channel, CAIS keeps the traffic in busy paths flowing, therefore removes possible

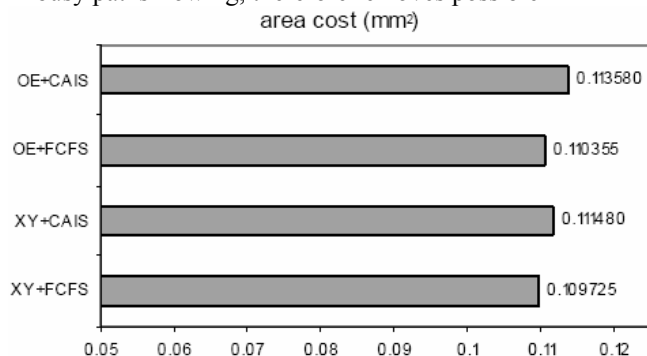


Fig. 11. Area cost of the switches

network congestion. Simulation has been carried out with a number of different traffic patters, including synthetic traffic and realistic voice CODEC traffic. The results shows that, no matter which output selection is used (deterministic XY routing or adaptive odd-even routing), the proposed CAIS achieved better performance than the traditional first-come-first-served (FCFS) input selection for most traffic patters except the transpose traffic, where CAIS has similar performance as FCFS. Furthermore, the employment of CAIS can make the XY routing (low complexity and hardware cost) to achieve similar or even better performance than the higher complexity and hardware cost odd-even routing for some traffic patters, i.e., area saving. The prototype switch with CAIS has been implemented and shows that CAIS has slight hardware overhead compared to FCFS (< 3%). As explained in Section IV.B, although not shown in the experiments, there is a starvation possibility in CAIS, which remains to be addressed in future work.

References

- [1] S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M. Millberg, J. Oberg, et al, "A network on chip architecture and design methodology," ISVLSI, pp. 117-24, USA, 2002.
- [2] L. Benini and G. De Micheli, "Networks on chips: A new SoC paradigm," Computer, vol. 35, pp. 70-78, 2002.
- [3] J. Henkel, W. Wolf, and S. Chakradhar, "On-chip networks: A scalable, communication-centric embedded system design paradigm," VLSI Design, pp. 845-851, India, 2004.
- [4] W. J. Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks," DAC, pp. 684-689, USA, 2001.
- [5] J. Hu and R. Marculescu, "DyAD - Smart routing for networks-on-chip," DAC, pp. 260-263, USA, 2004.
- [6] E. Nilsson, M. Millberg, J. Oberg, and A. Jantsch, "Load distribution with the proximity congestion awareness in a network on chip," DATE, pp. 1126-7, Germany, 2003.
- [7] T. T. Ye, L. Benini, and G. De Micheli, "Packetization and routing analysis of on-chip multiprocessor networks," Journal of Systems Architecture, vol. 50, pp. 81-104, 2004.
- [8] G.-M. Chiu, "The odd-even turn model for adaptive routing," IEEE Transactions on Parallel and Distributed Systems, vol. 11, pp. 729-38, 2000.
- [9] L. M. Ni and P. K. McKinley, "A survey of wormhole routing techniques in direct networks," Computer, vol. 26, pp. 62-76, 1993.
- [10] C. A. Zeferino, M. E. Kreutz, and A. A. Susin, "RASoC: A router soft-core for Networks-on-Chip," Designers Forum - DATE, pp. 198-203, France, 2004.
- [11] N. Kavaldjiev, G. J. M. Smit, and P. G. Jansen, "A virtual channel router for on-chip networks," IEEE International SOC Conference, pp. 289-93, USA, 2004.
- [12] E. Rijpkema, K. Goossens, A. Radulescu, J. Dielissen, J. Van Meerbergen, P. Wielage, and E. Waterlander, "Trade-offs in the design of a router with both guaranteed and best-effort services for networks on chip," IEE Proceedings: Computers and Digital Techniques, vol. 150, pp. 294-302, 2003.
- [13] D. Bertozzi and L. Benini, "Xpipes: A network-on-chip architecture for gigascale systems-on-chip," IEEE Circuits and Systems Magazine, vol. 4, pp. 18-31, 2004.
- [14] C. J. Glass and L. M. Ni, "Adaptive routing in meshconnected networks," ICDCS, pp. 12-9, Japan, 1992.
- [15] M. T. Schmitz, B. M. Al Hashimi, and P. Eles, "Systemlevel design techniques for energy-efficient embedded systems," Kluwer Academic Publishers, 2004.