RR-SFVP: Design of arbitration unit based on Round Robin method with Strong Fairness and Variable Priority for NoC Router

Elnaz Shafigh fard
Islamic Azad University
Mohammad Ali Jabraeil Jamali (m_jamali@itrc.ac.ir)
Islamic Azad University
Mohammad Masdari
Islamic Azad University
Kambiz Majidzadeh
Islamic Azad University

Research Article

Keywords: buffer, Network on Chip, router, Round Robin Arbiter

Posted Date: November 18th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2270678/v1

Abstract

Network on chip (NoC) is one of the communicative structures for multiple cores that has scalability. In designing the NoC micro-router architecture, the arbitration unit is very important due to its significant impact on performance, chip occupation level and NoC power consumption. In this paper, a router arbitration architecture is proposed with a combination of variable priority arbitration and Round Robin. In this architecture, arbitration examines the requests of other channels based on the Round Robin index after requesting the flit to exit the virtual channel in addition to checking the availability of the relevant virtual channel.The simulation results show that the architecture of the RR-SFVP arbitration unit compared to the standard RR method, is 13.7% smaller in area and has 5.7% less power consumption and 53.7% less critical path delay.

1. Introduction

Network on Chip (NoC) is an architecture that has been proposed to solve the problems caused by common bus. In this network, using modular and scalable structure instead of using a common bus, IP blocks are mapped on the network as tiles [1][2][3][4]. Packed data is transmitted among links through a built-in router, and in contrary to the bass-based approaches, NoC can support higher bandwidth and better scalability inside each tile [5] [6][7].

The data path of a router consists of input port buffer, crossbar of the switch structure, and a very important unit called the arbitration unit. The arbitration unit has a controlling role in traffic management and in addition it determines which virtual channel has the highest priority in transmitting its data in competition conditions [8] [9] [10].

The structure of the arbitration unit can be complex relative to the arbitration priorities and type of control.

In a NoC router design, critical path delay is usually observed in input ports, switches, and arbitrations. Compared to other units, this delay is high due to the complexity of the arbitration unit architecture. Hence, the arbitration unit circuit determines the maximum system speed. Therefore, the type of arbitration unit architecture is very important in the performance of the NoC system and its characteristics and its effect on speed and power consumption. [11][12][13][14].

Figure 1 shows two types of arbitration units with four input ports which both of them can arbitrate between n requests *(r_0, r_1… r_n)* using available resources based on a set of criteria and indicators, and each line that wins the arbitration is awarded *(Grant) Gi*. In terms of priority, arbitration architecture can be divided into two groups of fixed and variable architecture. For an arbiter with fixed priority, the priority of requests is determined linearly. Figure 1(b) shows an arbiter with a fixed priority in which *r0* has the highest priority and *r3* has the lowest priority [15][16].

In contrast to fixed priority arbitration, there is variable priority arbitration that in addition to being a sequential criterion, it considers several other indicators in offering a grant. Arbitration can be classified into three groups (weak, strong or FIFO) [17] in terms of fairness and type of arbitration. In a case of weak fair arbitration, any request is ultimately accepted regardless of a set of criteria and priorities. In strong fair arbitration, requests are replied based on a set of priorities and criteria depending on the specific circumstances. Fair requests are provided using the FIFO technique based on the first arrival of the first service. In addition, in order to have a fair arbitration, a variable priority arbitration unit can be used as shown in Fig. 1 (a). If this type of arbitration is changed in such a way that the priority is transferred from one cell to another in a time cycle, this type of arbitration is called round robin arbitration, which is shown in Fig. 2.

In this type of arbitration, for example, if the priority of g1 is high in the current cycle, it causes that P1 to be set in a high priority in the next clock, which leads that r2 to reach the highest priority in the next clock and r1 have the lowest priority.

Round Robin arbitration model [18] is simple and easy to execute and has no hunger. When the number of input requests is high, the round robin arbitrator structure grows and leads to more area, higher power consumption, and more critical path delays on large chips.

Based on the RR method, some other important factors that make arbitration more fair have been considered in this paper. In this method, in addition to exit request cycles, it also responds to cycles in which no request is issued from any of the channels (strong arbitration). Besides this advantage, a circuit has been designed that has a much lower critical path delay than previous methods such as RR.

The following can be considered as contributions to this article:

- By applying variable and fair priority to RR arbitration, the critical path delay and power consumption of NoC are reduced.

- Applying effective factors in selecting a strong arbitration and offering a grant to the relevant channel when there are competition conditions between several virtual channels to select an output port.

- The proposed architecture requires minimal hardware compared to other arbitration architectures.

In the continuation of the article, in section 2, the previous works and in section 3, the **RR-SFVP** arbitration architecture are presented. Section 4 reviews the evaluation and performance of the **RR-SFVP** arbitration architecture and concludes.

2. Previous Research

The Matrix round robin arbitration unit architecture was proposed by Dally and Towles. This architecture examines the input requests in order and it is one of the methods of arbitration with strong fairness while maintaining the previous grant. This type of arbitration is useful for a small number of inputs. However,

due to the high complexity of the hardware and the high use of resources, it causes the researchers to choose a better arbitration unit [19]. This arbitration unit is used in a router whose input port is a link-list DAMQ (LLD) that is called LLD-Matrix router. In this method, the input port uses the links of lists and tables to write and read the flits in the buffer. The router with the LLD input port is an approach with relatively low hardware cost but low performance. This approach leads to simultaneous communication which is not useful for NoC routers. Link-list approach for updating five tables after each write and read causes long delays. Many updates to the mentioned tables also lead to long delays. This delay also increases with increasing input rate [20].

Matrix round robin arbitration method is also used with a router whose input port is ViChaR. In this method, unlike the LINK-LIST DAMQ model, LINK LIST is not used. ViChar controller circuit is very expensive in terms of hardware. This method can support virtual channels with the largest buffer slot size, which requires arbitration to allocate slot buffers to virtual channels and switches. Larger size buffer slots can convert to a bottleneck in critical paths that is a kind of limitation for NoC. Complexity, limitation of adjustment and longer pipeline in the entry and exit of flits are other disadvantages of this method [21].

Fu and Ling compared the two methods RoR [17] and Matrix [19] in terms of resource consumption, performance and power consumption for an FPGA platform [22]. They concluded that Matrix arbitration consumes more resources. The two methods have equal power consumption, the Matrix method can process data faster than the RoR method.

Zheng and Yang proposed a round robin arbiter method in which inputs are arbitrated in parallel. PRRA is based on a simple binary search algorithm to improve latency consisting of four inputs. An improved design (IPRRA) has been proposed to reduce the delay of the critical path of PRRA architecture [23]. The IPRRA method dramatically reduces the delay from the PRRA critical route.

The Round Robin Arbiter (HDRA) method was proposed by Lee et al. [24]. In this method, there is a filter circuit for each input, based on the indicators in that circuit and the order of request, the grant is assigned to a specific input. In this method as well as PRRA methods, PRRA has weak fairness.

Another arbitration that has a better architecture in terms of power consumption and delay compared to similar methods is IIR arbitration. In this architecture, requests r0, r1,… rn are reported to the arbitration unit. This unit examines them on their order basis, and if eligible, a grant will be assigned to the approved request. The IRR_WF method, which is similar to the IRR method, does not store the previous grant clock value. This method is called arbitration with weak fairness. The difference between IRR and IRR_WF is that if there is no exit request, the IRR method records the previous request due to the presence of REG, and this is considered as a fair arbitration for the next cycle. However, in the IRR-WF method, due to the lack of a register, if all requests are equal to zero, the previous request will also be lost [25].

In the EDVC-IRR method [26], in the input port of the router, the EDVC method and in its arbitration, the IRR method have been used. EDVC is another input port design method that uses a common buffer with

dynamic virtual channels to allocate flits, which is more efficient and less delayed than previous methods. In this method, the circuit hardware designed for the flits read and write mechanism is large, which can increase the cross-sectional area of the chip using power by increasing the size of the buffers.

Table 1 summarizes all the methods, their functions, advantages and disadvantages.

Table 1
Comparisons of the implemented methods.

Method	Performance	Advantages	Disadvantages	
IRR[25]		All requests enter the general multiplexer, which by selecting a request, it is registered and leads to a strong arbitration (without request) and goes to the next request with the counter (in case of request)	Less delay and cross-section compared to other methods -Simplicity -Less overhead	Fixed priority arbitration
ROR[17]		It acts as a variable priority and enters the arbitration process in a chained manner with priority of p0 and i0 and proceeds in order	-Capability to save current priority in the cycle without request	Processing speed slower than other methods
Matrix[19]		Resets the bits of ith row and sets the bits of ith column as matrix	-Faster data processing speed - Fair arbitration	Higher resource consumption
IRF_WF[25]		Like the IRR method minus the SF multiplexer	-Faster arbitration than IRR -Less consumption area than IRR	Weak arbitration (failure to save the current grant) in the absence of a request
HDRA[24]		Requests are of the fixed priority type, each request consisting of a flip-flop and multiplexer, where r0 initially has the highest priority, and reaches the other rs without requesting r0 in chains.	Less delay on critical routes	Weak arbitration (failure to save the current grant) in the absence of a request
PRRA[23]		Round robin method based on binary search algorithm (parallel)	Reduction of the critical path delay of the RR method	The longest delay among all arbitration methods Weak arbitration (no current grant record) High power consumption

Method	Performance	Advantages	Disadvantages	
IPRRA[23]		Improved model of PRRA method	Reduction of the implementation schedule of the PRRA method	High delay The highest cross section compared to other methods

3. Proposed Method

Figure 3 shows the logic circuit of the router arbiter logic block diagram.

As can be seen from the circuit in Fig. 3, if in the first class, r_i (Request) is active by checking the signal value of flip_flop (i) and 0), even if other channels have requested, priority is given to r_i. However, If the signal r_i is active and equal to one, it is time to check flip_flop(i). If the value of f_i has a signal of 1 thus G_i is already 1 and the other input signals are checked and at least one of the other virtual channel signals requests g0 equals 0.

One of the features of this arbitration method is its strong arbitration that if no request is issued through virtual channels, it can save the final request.

In the **RR-SFVP** method, all the positive cases in the design of the arbitration unit circuit, including variable priority, arbitration with strong fairness (preservation of the previous cycle grant if not requested in the current cycle) are used to reduce the critical path delay and the area of the arbitration circuit. In this method, a combination of the constant and variable priority methods with round robin technique has been used.

As shown in Fig. 3, when there are competition conditions among multiple virtual channels for choosing an output port, several factors are involved in selecting a strong arbiter and offering the grant to the respective channel, which are: 1. Request to exit r(i) 2. Value of flip_flop(i) for awarding grant 3. Being influenced by requests from other channels .

According to the algorithm of Fig. 4, the factors related to each channel are determined. After receiving the request signal to exit channel i, the request time operation, the number of updates of each channel and the request of other channels are checked respectively. If a packet requests to exit the virtual channel, the amount of the last access will be checked, and if it received a grant in the previous arbitration, it will not be able to receive a grant again in this clock unless no request is received from other virtual channels, if in the previous methods, the inputs had to be checked in order to be informed about lack of request and it was not done in parallel. In order to save the amount of previously saved grants, it can use a flip-flop to save both the previous amount related to receiving or not receiving a grant, and that if a request is not issued from any of the channels, the value of r is equal to 0 priority should be given to output.

At first all variable be set to 0. In this algorithm ack_1, ack_2 are calculated, which this operation is performed for all virtual channels, respectively.The meaning of other requests is the result of the output port of the or gate as shown in the Fig. 3.

According to the algorithm of **RR-SFVP** method, other requests must be checked in clocks where the request is zero for a fair and robust arbitration. If no request is asked from other channels, the previously stored value in the flip-flop is transferred to the output. For example, if in the previous cycle, the value of the grant related to this virtual channel was zero, it would be zero at the output, and if it was one, number 1 is transferred to the output. However, in the clocks where the request signal of a virtual channel is one, according to the algorithm proposed in Fig. 4 (mentioned parameters) as in the request of other channels, the previous value of the flip-flop from the previous cycle is examined. Given that for leaving the relevant flit from the virtual channel, the current value of its flip-flop in the relevant clock must be zero, so according to the scenario in Fig. 5, if the number of flip_flop is 0 and the request signal is 1, the corresponding grant (G) value is 1. Therefore, if the signal related to flop_flop is 1, this shows that in the previous cycle the grant related to this channel was 1, if no request is issued from other channels in the same clock, the relevant grant will be 1, otherwise, it will be 0.

According to the scenario in Fig. 5, there are two ways to obtain a grant (G = 1). The first case is when the value of flip_flop equals 1, the value of the Request for exit from the virtual channel equals 1 and other requests equal to 0.

The second case for a virtual channel grant (G) to be 1 is when flip_flop is zero, the request for a virtual channel exit equals 1, and the value of other requests is not important.

Also, according to the scenario in Fig. 5, there are two states for the case that the virtual channel is not allocated grant (G = 0). The first case is when all three values of flip_flop(i), Request(i) and other requests(n-i) are equal. In the other case, the value of flip_flop(i) equals 1 and the Request(i) signals related to exit the virtual channel and other requests(n-i) equal 0.

In the **RR-SFVP** method, in addition to having the feature of arbitration with strong fairness, all the factors involved in an arbitration for a specific channel including the time of request, the number of times to access the service and checking of other requests are also considered.

In the proposed model, concurrent processing of all channels in parallel based on the proposed algorithm, considering other requests and examining the request history of one channel can lead to a fairer and robust arbitration.

For the input port of this type of arbitration, the input port of the method mentioned in [27] has been used. So that each flit is stored in one of the virtual channels when entering after checking the buffer slot table and the emptiness of each section based on the write pointer and header flit. The id related to the header flit is assigned to the corresponding flit id.

At the time of accepting a new flit, the empty slots in the shared buffer must be identified in the second phase after specifying one of the virtual channels.

In order to evaluate the mentioned arbitration scheme, the input port in the router is needed, which [27] has been used to read and write part of the input port from/to the buffer. One of the advantages of this method is the use of a table and two simple read and write circuits. The input port of this method has a hardware architecture that leads to parallel processing and at the same time finds the first empty slot in the buffer to write and simple and parallel hardware has been used to read from the mentioned address in the buffer.

4. Evaluation Of The Proposed Method

In this section, the **RR-SFVP** router is simulated and compared with similar cases based on structure and architecture.

The main efficiency criterion (figures of merit) of the designed circuit is its speed, power consumption and area, which is the usual measurement for the speed of the arbitration circuit, its delay or maximum frequency (Fmax). The frequency of the arbitration circuit depends on the longest delay (critical path) between two registers at a time.

The electrical parameters of the logic gates are taken from the standard Synopsys 90nm Digital library listed in Table 2.

Hardware requirements and architectural features of the arbitration unit of the **RR-SFVP** and other methods have been mentioned using the Synopsys design compiler for general technology 90 nm and using the ise Xilinx simulator for Virtex 7 FPGA.

The complete characteristics of the modified NoC framework are shown in Table 2.

Table 2
Structure of simulation parameters

Network Size	**8 x 8**
Packet/Flit/Data	16-bits
VC and I/O ports	4 VCs for each 5 ports
Switching mode	Wormhole
Topology	Mesh
Routing Algorithm	XY routing
Traffic Patterns	Tornado, Complement,tornado

Microarchitecture and simulation using verilog have been used for measuring and evaluating the performance of various types of parameters, including estimating latency and throughput.

Critical path, power consumption, arbitration circuit area based on Table 3 are given in Table 4.

Table 3
Characteristic of gates

Gate name St.	Propagation Delay (ps)	Power (nW) Dy.	Power (nW/MHz	Area (µm2)
INVX1	38	88	12	6.5
AND2X1	85	298	19	7.4
AND3X1	119	297	34	8.3
NAND2X1	51	336	15	5.5
OR2X1	85	226	23	7.4
OR3X1	114	250	39	9.2
OR4X1	137	261	56	10.1
NOR2X1	64	170	15	6.5
MUX21X1	107	815	43	11.1
MUX41X1	168	827	58	23.0
DEC24X1	119	1238	66	29.5
XOR2X1	133	454	26	13.8
DFFARX1	217	620	100	32.2

Table 4
Characteristics Of 4-Input Arbiters Based On Table 3

Type of 4-input arbiters	**Area (μm2)**	**Power (μW)**	**Critical Path Delay (ps)**
IRR	294	296(282d)	625 (217 + 133 + 168 + 107)
RoR	328	298(289d)	1242(217 + 5*(85 + 85) + 137 + 38)
Matrix	556	479(465d)	747 (217 + 2*38 + 3*85 + 114 + 85)
IRR_WF	280	274(262 d)	518 (217 + 133 + 168)
HDRA	431	360(348d)	609 (217 + 64 + 51 + 85 + 85 + 107)
PRRA	510	493(479d)	861(217 + 2*38 + 3*85 + 85 + 2*114)
IPRRA	528	488(473d)	747 (217 + 2*38 + 3*85 + 85 + 114)
RR-SFVP	**283**	**281(271d)**	**574(217 + 107 + 51 + 85 + 114)**
Arbiters	Saving	Saving	Faster
RR-SFVP /IRR	3.7%(better)	5%(better)	8.16%(better)
RR-SFVP /RoR	13.7%(better)	5.7%(better)	53.7%(better)
RR-SFVP /Matrix	49%(better)	41%(better)	23%(better)
RR-SFVP /IRR_WF	1%(worse)	3%(worse)	5%(worse)
RR-SFVP /HDRA	34%(better)	21%(better)	5.7%(better)
RR-SFVP /PRRA	44%(better)	43%(better)	33%(better)
RR-SFVP /IPRRA	46%(better)	42%(better)	23%(better)

As shown in Table 3, based on the hardware design, the **RR-SFVP** has less power consumption, area and critical path compared to other methods, except for the IRR_WF method, which is one of the methods with weak fairness, so that if no request is asked from any data channel, the last grant given will not be preserved, so this design has a smaller area, resulting in less power consumption and a shorter critical path compared to the **RR-SFVP** Three main factors in designing circuit have been referred in this table that the first case is occupied area by arbiters unit based on the used gate and the second case is consumed power and the last column is about critical path delay.

Evaluation of the **RR-SFVP** in terms of latency and throughput

We measured the throughput and latency, while the throughput is measured at the rate of received packets to the maximum number of packets injected at a given time, which can be expressed (1) as

follows:

$$throughput = \frac{Equation Number of recieved packets * size of 1 packet}{Equation Number of nodes * Equation Number of cycles}$$

1

The average delay with the average latency (in each clock cycle) related to the exit and entry of a certain number of packets in NoC is measured by the following formula (2).

Latency $= |departure\,(time) - arrival\,(time)|$(2)

Figures 6, 7 and 8 show the average latency and throughput criteria for the mesh topology (8.8) for Complement and Random Tornado traffic patterns [28] [29][30] respectively. For the mesh topography m × m, source address (Sx, Sy) and destination address (Dx, Dy) where 0 ≤ x, y ≤ m-1, with relations (3), (4) and (5) are determined for Tornado, Complement and random traffic patterns.

For Tornado:

Dx = Sx + (m/2) -1, Dy = Sy + (m/2)-1 (3)

For Complement:

Dx = m-Sx-1, Dy = m-Sy-1 (4)

For Random:

Dx = 1/m, Dy = 1/m (5)

In XY routing and tornado traffic, all routers are uniformly crowded, while in complement traffic, side routers are more crowded than middle routers, and in random type, with equal probability, the packet can be transferred to other nodes. It is done for 8.8 network in this experiment and a packet consists of 16 flits and each input port includes a central buffer with 8 slots.

Compared to other arbiters, it is efficient for arbitration structures with different inputs. On average, the chip area is smaller, the power consumption is lower, and it has a smaller value in the critical path. In general, this method consumes the least amount of power among all judges due to the use of fewer gates. The number of gates used also makes the design and layout of the chip simpler.

As can be seen from Figs. 6, 7 and 8, the test results of the **RR-SFVP** are much better compared to other methods, especially at high rates. In higher injections, the flit population increases and the competition between the flits enhances.

Virtual channels reduce the density of the input port. Packets without compaction take up more buffer space.

Compared to Link-List [20], EDVC-IRR [26] and ViChaR [21] for 4VC and implementation of 8-slots show much higher output NoC performance and lower average latency than Link-List and ViChaR NoC for different traffic models. The reason for this acceleration is the performance of the input port, which has a direct effect on reducing the number of executable cycles. In addition to the type of input port, due to the designed circuit all channels are checked in parallel and concurrent according to arbitration unit in some part of the circuit and based on the structure of the circuit and its design, have a higher processing speed. Thus, in other methods, the operation of selecting the desired port is performed sequentially, the requested port may not be done in one clock, which in turn wastes time, but in the **RR-SFVP**, based on what mentioned in section 3, the arbitration operations are performed in almost all channels in parallel and simultaneously so that parallel processing causes the general consideration of requests and makes the best choice based on the criteria stated in the **RR-SFVP** section.

5. Conclusion

In this paper, the proposed RR-SFVP microarchitecture, which is a modified RR method, has been presented. The buffer uses an arbitration unit among several VCs that has strong priorities for selecting a port in the competition between several ports so that no port is starved and no port has a higher priority than other ports. Evaluations show that the RR-SFVP method has less area and power consumption compared to other methods. Compared to the RR arbitration unit, which is one of the standard methods, it has reduced the critical path delay by 53.7% and decreased it by 13.7% compared to the area with enhanced power consumption of 5.7%. On the other hand, the simulation results indicate that the method **RR-SFVP** compared to the IRR method, which is one of the newest arbitration methods, has 3.7% less area and has a power consumption of 5% and less critical path delay of 8.16%. Making changes in the arbitration unit architecture, including the use of shared comparison gates that reduce critical path delays can be suggested as future work.

Declarations

Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Authors' contributions

elnaz shafigh fard Conceived and designed the analysis and Contributed data or analysis tools.

Mohammad Ali Jabraeil Jamali Performed the analysis.

Mohammad Masdari Collected the data.

Kambiz Majidzadeh contributed to the interpretation of the results.

All authors reviewed the final manuscript.

Funding

The authors of the article have not received any financial support from any organization in connection with this research.

Availability of data and materials

The data that support the findings of this study are available on request from the authors. The data are not publicly available due to privacy and ethical restrictions.

References

1. Benini, L. *and G. D. Micheli, Networks on Chips: Technology and Tools. Morgan Kaufmann, 2006.*
2. Kumar, S., Jantsch, A., Soininen, J. P., Millberg, M., & Tiensyrja, K. (2002). Forsell, M., Millberg, M., Oberg, J.,Tiensyrja, K.,*and* Hemani, A., *"A network on chip architecture and design methodology," in IEEE Symposium on VLSI*, pp. 105–112.
3. *W. J. Dally and B. Towles, Principles and Practices of Interconnection Networks. Morgan Kaufmann.*
4. *F.Rad1, M.Reshadi1, A.Khademzadeh A novel arbitration mechanism for crossbar switch in wireless networkon-chip Springer Science + Business Media, LLC, part of Springer Nature 2020*
5. Dally, W. J. *and B. Towles, "Route packets, not wires: On-chip interconnection networks," in Proceedings of the Design Automation Conference (DAC), June*
6. Nirmala, M.,*V. Shylaja,fpga implementation of priorityarbiter based router design for noc systems, International Journal of Advanced Research in Engineering and Technology (IJARET), Volume 10, Issue 2, March - April* pp. 509–516.
7. *B.Aghaei,M.Reshadi,M.Masdari,S.H.Sajadi,M.Hosseinzadeh,A.Darwesh, Network adapter architectures in network on chip: comprehensive Literature review,Springer,Science + Business Media, LLC, part of Springer Nature 2019.*
8. Mamaghani, M., & Jabraeil Jamali, S., *M.A. A load-balanced congestion-aware routing algorithm based on time interval in wireless network-on-chip. J Ambient Intell Human Comput 10, 2869–2882 (2019*). i>https://bbibliograficas.ucc.edu.co:2160/10.1007/s12652-018-1020-z.
9. *R.Akbar, F.Safaei, A novel heterogeneous congestion criterion for mesh-based networks-on-chip, 0141–9331© 2021 Elsevier B.V. All rights reserved.*
10. A.T S, B.M, U.Sankar S M, R. Thiagarajan, *A.D.G, P.Raja Rajeshwari, A.Sai Kumar, S.A, Evaluation of low power consumption network on chip routing architecture, 0141–9331/© 2021 Elsevier B.V. All rights reserved*.
11. *A. Jantsch and H. Tenhunen, Networks-on-Chip. Norwell, MA: Kluwer, 2003.*
12. Ogras, U. Y., Hu, J., *and R. Marculescu, "Key research problems in NoC design: A holistic perspective," in Proceedings of the 3rd IEEE/ACM/IFIP International Conference on Hardware/Software Codesign*

and System Synthesis (CODES + ISSS), Jersey City, NJ, USA, September 19–21

13. Uma, R., Sarojadevi, H., Sanju, V., *"Network-On-Chip (NoC) - Routing Techniques: A Study and Analysis" Global Conference for Advancement in Technology ©2019 IEEE*
14. *S.Sreekumar, Literature survey on Network on Chips, © 2021 JETIR March 2021, Volume 8, Issue 3.*
15. Dally, W. J. *and B. Towles. "Arbitration," In: Principles and Practices of Interconnection Networks. Morgan Kaufmann Publishers*, pp. 349–362.
16. *Dr. Jayanthi K Murthy, Nishanth.B.N, Design and Analysis of Arbiters for the NoC's Routers, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (An UGC Approved Journal) Website: www.ijareeie.com Vol. 6, Issue 9, September*
17. Dally, W. J. *and B. Towles. "Arbitration," In: Principles and Practices of Interconnection Networks. Morgan Kaufmann Publishers*, pp. 349–362.
18. *Dr. Sohail A. Dianat, Professor, Design and Verification of a Round-Robin Arbiter, Theses 8-*
19. Nadu, T., *J. Arjana, Low Latency Noc with Dynamic Priority based Matrix Arbiter*,Indian Journal of Science and Technology, *Vol 9*(*29), DOI*:i>10.17485/ijst/2016/v9i29/91707, August
20. Evripidou, M., Nicopoulos, C., Soteriou, V. (2012). *and J. Kim,"Virtualizing Virtual Channels for Increased Network-onChip Robustness and Upgradeability," in Proc. IEEE Symp. on VLSI, pp. 21–26, Amherst, MA, Aug.*
21. Nicopoulos, C. A., Dongkook, P., Jongman, K. (2006). Vijaykrishnan, N., *M. S. Yousif and C.R. Das, "ViChaR: A Dynamic Virtual Channel Regulator for Network-on-Chip Routers," in Proc. IEEE/ACM Int. Symp. on Microarchitecture, pp. 333–346, Orlando, Dec.*
22. *Z. Fu and Xiang Ling, "The design and implementation of arbiters for Network-on-chips," Proc. 2nd Int. Conf. Industrial and Information Systems, Dalian*, 292–295.
23. *S. Q. Zheng and Mei Yang, "Algorithm-Hardware Codesign of Fast Parallel Round-Robin Arbiters", IEEE Trans Parallel and Distributed Systems, vol. 18, January* pp. 84–95.
24. Lee, Y.-L., *Jer Min Jou, and Yen-Yu Chen, "A High- Speed and Decentralized Arbiter Design for NoC," Proc. IEEE/ACS Int. Conf. on Computer Systems and Applications, Rabat*, 350–353.
25. *M. Oveis-Gharan and G. N. Khan, "Index-Based Round- Robin Arbiter for NoC Routers," in Proc. IEEE Computer Society Annual Symposium on VLSI, pp. 62–67, Montpellier, July*
26. *Oveis-Gharan, Masoud, and Gul N. Khan, "Efficient dynamic virtual channel organization and architecture for NoC systems, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, pp. 465–478*,
27. Shafigh Fard, E., Jabraeil Jamali, M. A., Masdari, M., *K. Majidzadeh, An efficient NoC router by optimal management of buffer read and write mechanism*, i>https://doi.org/10.1016/j.micpro.2022.104440
28. Das S, Doppa JR, Pande *PP, et al.) Energy-efficient and reliable 3D network-on-chip (NoC): architectures and optimization algorithms. In: International Conference on Computer-Aided Design.*

29. Alfaraj, N., Zhang, J., Xu, Y. (2011)., *and H.J. Chao, “HOPE: Hotspot Congestion Control for Clos Network On Chip,” in Proc. IEEE/ACM International Symposium on Networks on Chip, pp. 17–24, Pittsburgh, PA, May*
30. Xu, Y., Zhao, B., Zhang, Y. (2009)., *and* Yang, J., *"Simple virtual channel allocation for high throughput and high frequency on-chip routers," in Proc. International Symposium on High Performance Computer Architecture, pp. 1–11, Bangalore, India, January*

Figures

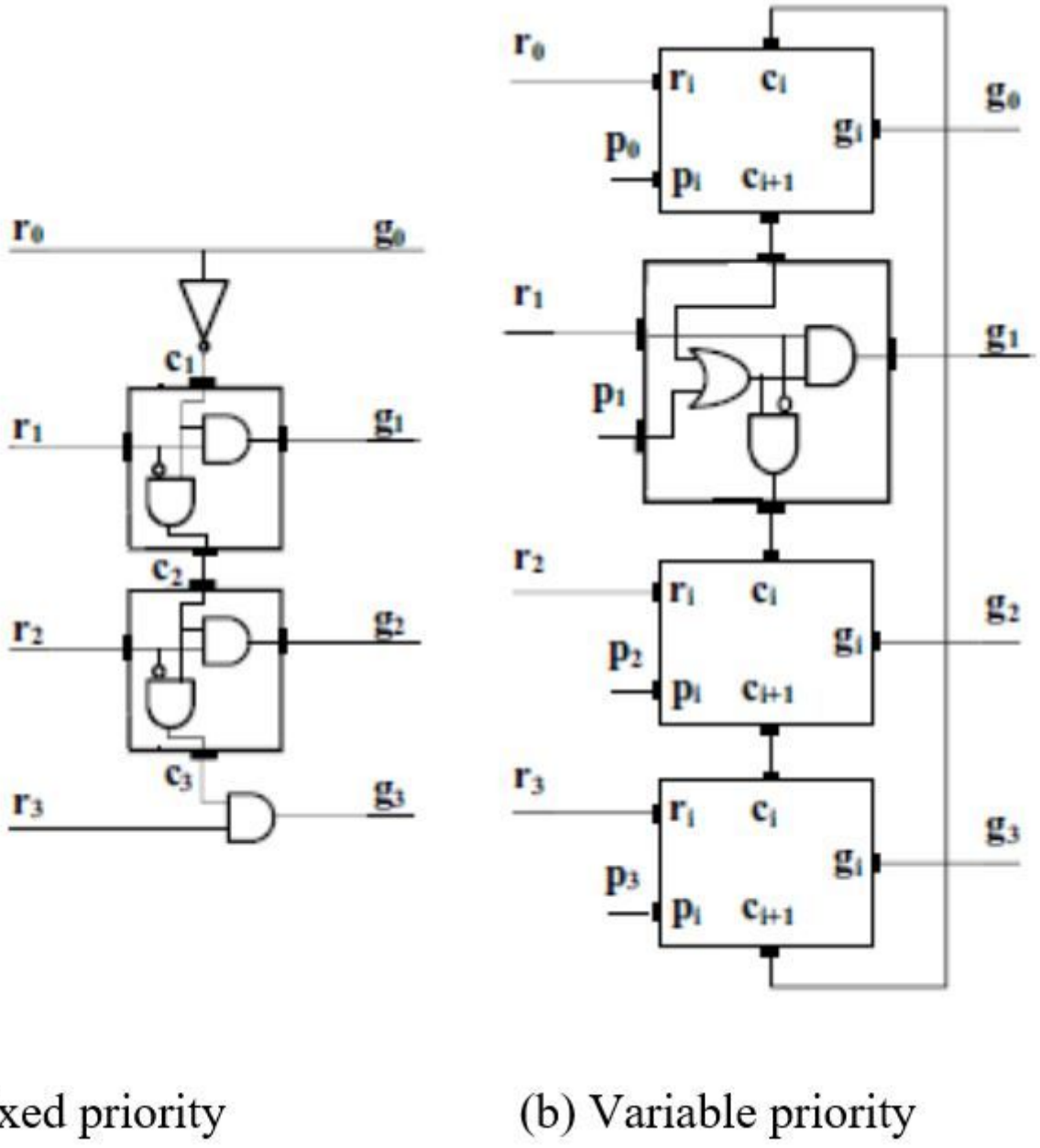

Figure 1

Arbitration architecture with 4 input ports

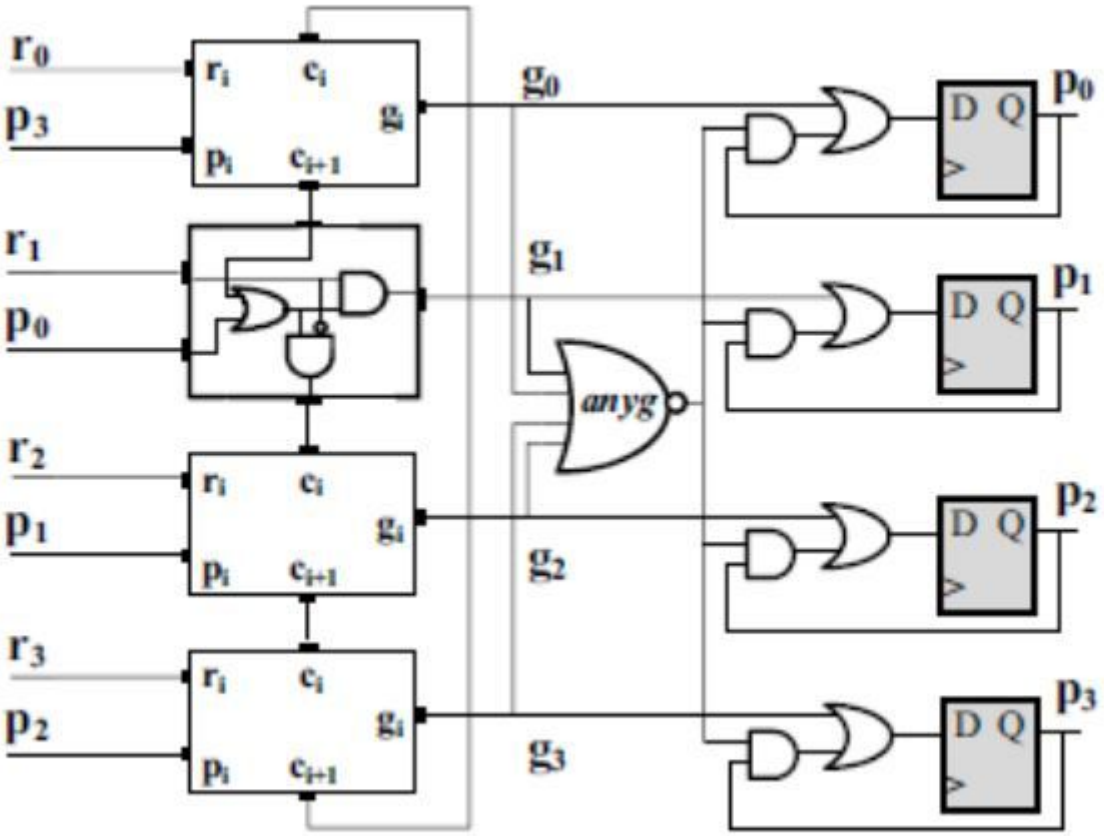

Figure 2

Round robin arbitrer architecture with variable priority

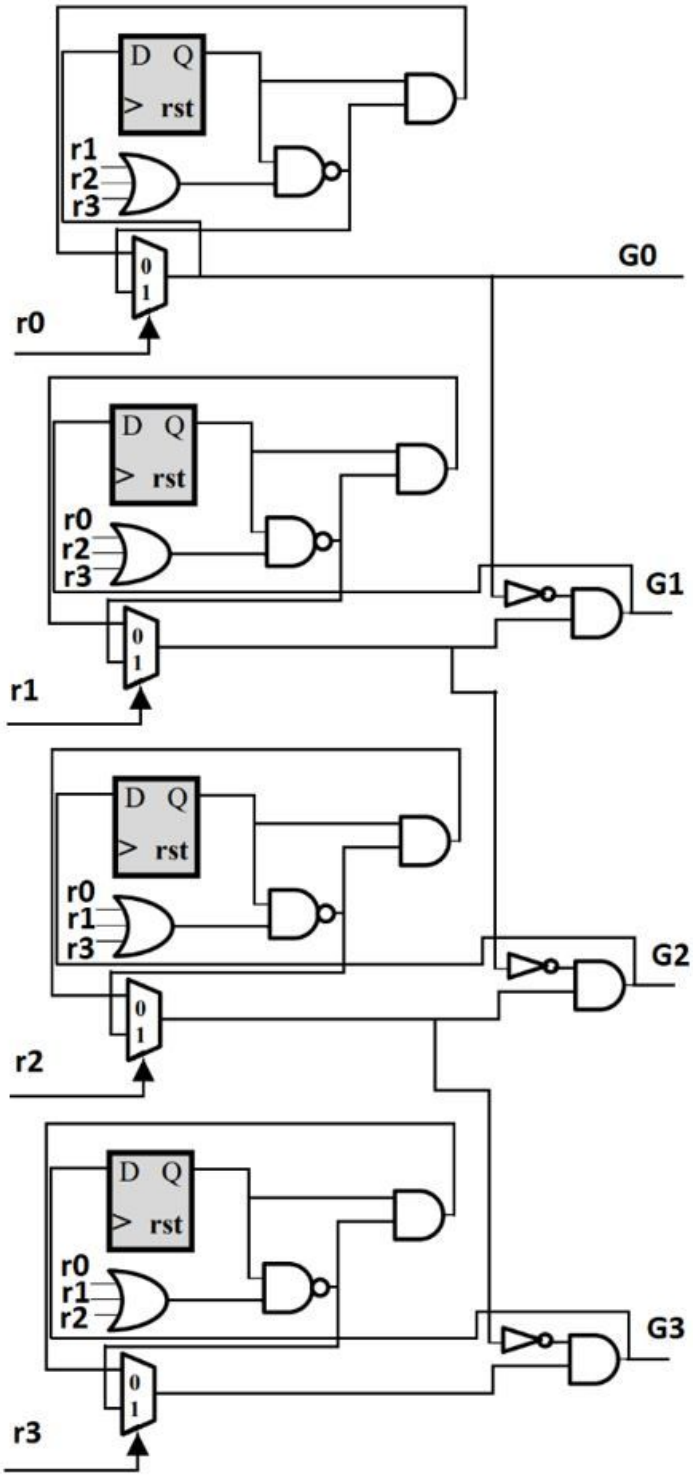

Figure 3

RR-SFVP method arbiter circuit

Step 1: Start
Step 2: Declare variables flip_flop[N],
request [N], other request [N],
Step 3: Initialize variables
Step 4: Repeat the steps until i=N
4.1: If i > N
i <=0
Else
read the value of flip_flop [i]
read the value of request [i]
read value of other requests [N-i]
if request [i] = 0
ack_1 [i] = flip_flop [i] ***nand*** other_requests
[N-i]
ack_2 [i] = flip_flop [i] ***and*** ack_1 [i]
Else
ack_1 [i] = flip_flop [i] ***nand*** other_requests
[N-i]
if ack_1 [i] = 1
G(i)=1

Else

G(i)=0

i<= i+1
Step 5: Stop

Figure 4

The algorithm of **RR-SFVP** method

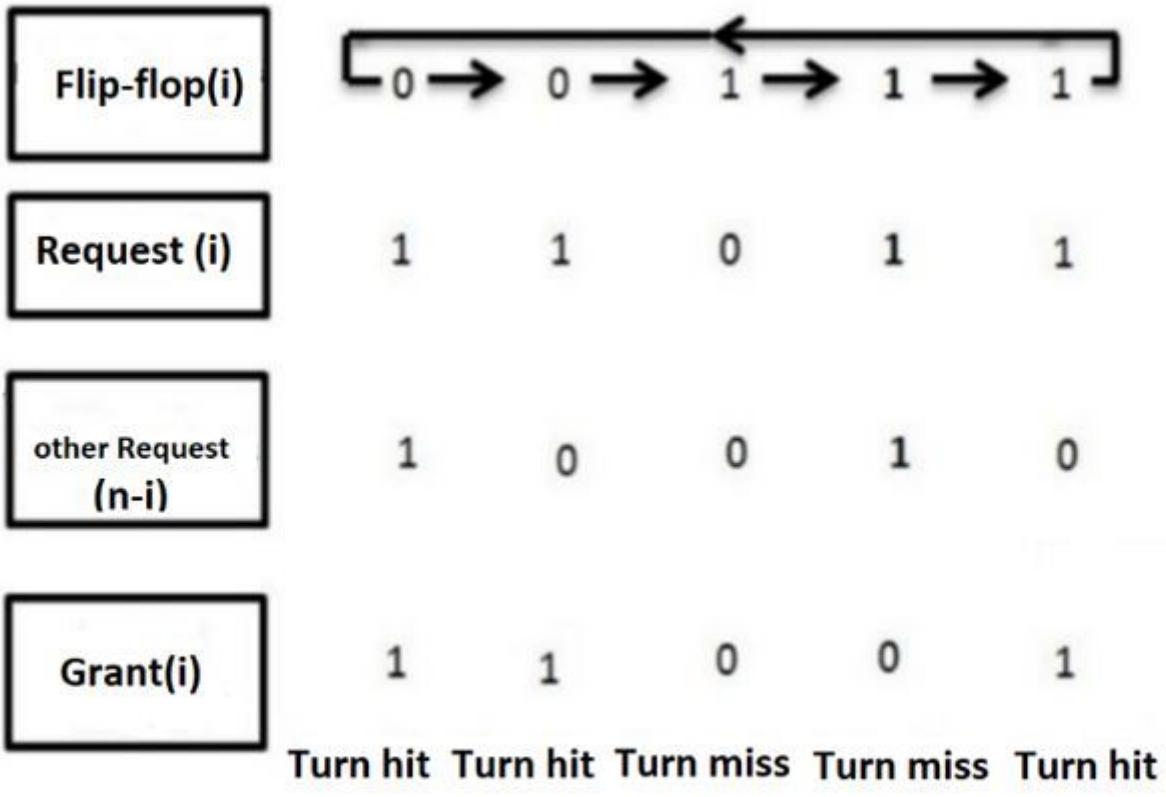

Figure 5

Scenario of RR-SFVP method

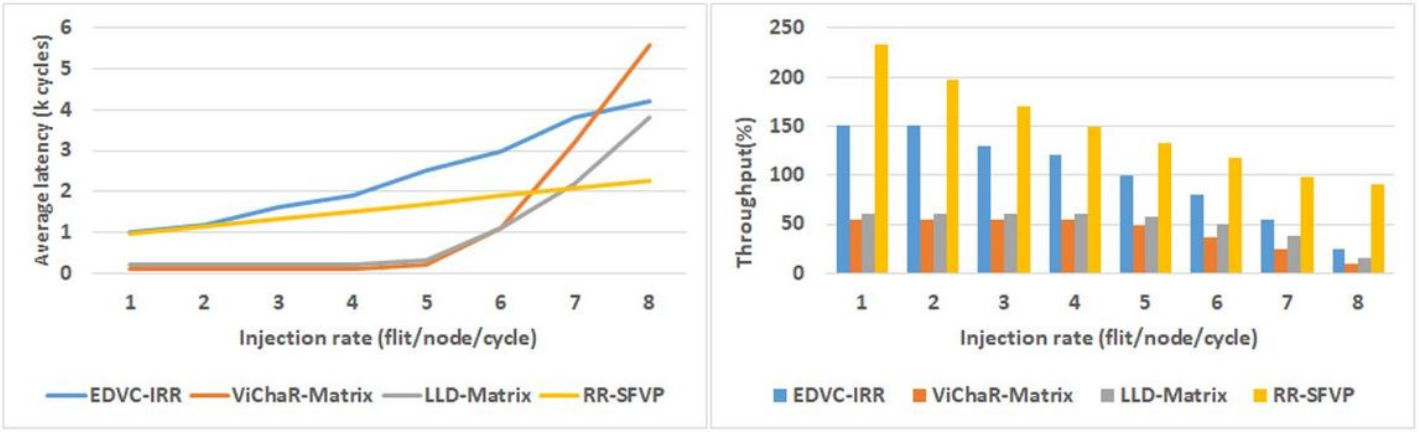

Figure 6

Latency and average throughput for random traffic

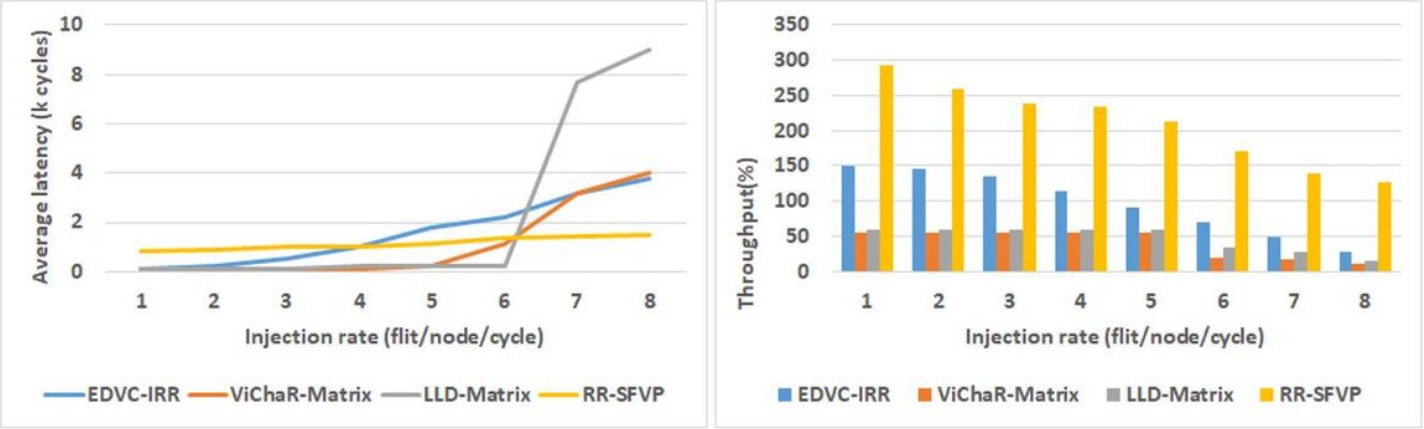

Figure 7

Latency and average throughput for tornado traffic

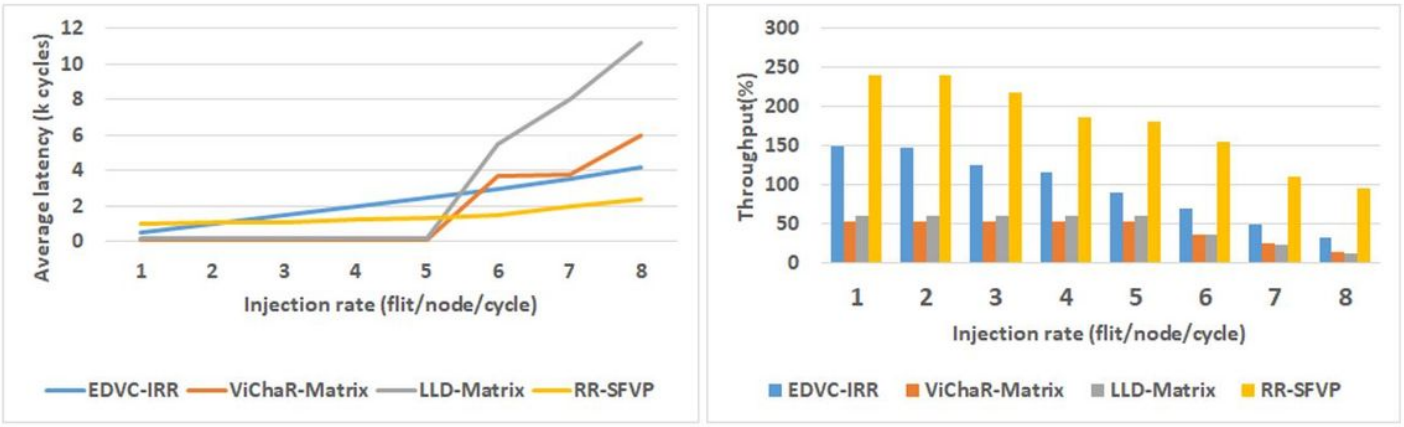

Figure 8

Latency and average throughput for complement traffic