A New Methodology in Study of Effective Parameters in Network-on-Chip Interconnection's (Wire/Wireless) Performance

Mostafa Haghi
Center for life science automation (Celisca)
Rostock University
Rostock, Germany

Norbert Stoll
Institute of Automation
Rostock University
Rostock, Germany

Kerstin Thurow
Center for life science automation
Rostock University
Rostock, Germany

Saed Moradi
Department of Electrical engineering
Isfahan University
Isfahan, Iran

***Abstract*—Network-on-Chip (NoC) paradigm has been proposed as an alternative bus-based schemes to achieve high performance and scalability in System-on-Chip (SoC) design. Performance analysis and evaluation of on-chip interconnect architectures are widely considered. Time latency and throughput are two very critical parameters which play vital role to improve the system performance. In this work, these two elements are evaluated in both wire and wireless approaches under different conditions for networks contain 64,512 and 1024 number of cores. There are number of parameters those have direct and indirect effects on the delay and throughput, among all, these four are chosen: routing algorithm, buffer size, virtual channel and subnet. Thus this work is clustered into two general parts, in the first section the effect of algorithms and buffer size are calculated and then later on in second part when switching from wire approach to wireless, it's shown that, virtual channel and subnet are able to influence the performance of a network on chip positively under some circumstances. We don't concentrate on approach and techniques here. Our target in this paper is to determine the critical points, trade-off and study the effect of mentioned parameters on entire system. Evaluation is done by means of Booksim and Noxim simulators which are based on system C.**

***Keywords*—network on chip; on-chip interconnection; buffer size; virtual channel; subnet**

I. INTRODUCTION

System-On-chip (SoC) has recently emerged as a key technology behind most embedded and smart miniaturized systems to provide high flexibility and better performance. These systems can find an appropriate location in industry and market while in addition to providing high-performance, meet very important system requirements, such as a low power consumption, low time latency and small occupied area. Therefore, the design of these systems should be highly flexible, adaptable, and fulfil stringent time-to-market constraints. A key element in the performance and energy consumption in SoCs is the On-Chip Interconnect (OCI), which allows different SoC components to communicate efficiently. In some aspect we can claim that the majority of power is consumed by OCI. Network-on-chip has been proposed as an alternative to bus-based schemes to achieve high performance and scalability in SoC design [1], [18]. Different OCI-based architectures using packet-switching have been recently studied and adapted for SoCs. Examples of these architectures and topology are Fat-Tree (FT)[2], 2D mesh[3], Ring[4], Butterfly-Fat Tree (BFT)[5], Torus[6], Spidergon [7], Octagon[8], WK-Recursive[9]. However, as the architecture is improved to enhance the performance of the system, their increasing complexity makes their design extremely challenging. Furthermore, another parameter which is important to be studied is traffic generated between components and traverse the OCI [10]. Therefore, it is very helpful to perform a traffic analysis and predict the process in early stages of the design to determine an appropriate traffic model, such that the designer can select suitable parameters for the on-chip interconnect architecture. Indeed, the selection of the on-chip interconnect architecture, based on traffic patterns as an application, which SoC generates, allows designers to detect and locate network contentions and bottlenecks. Here in this paper as is seen in majority of research works, traffic is assumed to be Random. Generally, the simulation is extremely slow for large systems and provides little insight, on how various parameters affect the actual NoC performance [11].

Two simulators based on system C are applied to evaluate the studied parameters, Noxim and Booksim for wire and wireless sections respectively. However, analytical models, allow fast evaluation of performance metrics in early stages of the design process. This paper extends the work by evaluation of the performance (e.g., latency, throughput) of two on-chip interconnect methodologies under three on-chip routing algorithms architectures for wire-based and wireless using Mesh and Flattend Butter-fly. It is shown how hybrid interconnects can be used to improve the performance and design tradeoffs. The main objective is to illustrate the effectiveness of 4 elements (Buffer Size, Routing Algorithm, Subnet, Virtual channel) in evaluation of on-chip interconnect architectures. Work is extended by calculation of effect of first two parameters on wire-based section, and next two in wireless part. Furthermore, variation of Packet Injection Rate (PIR), is carefully investigated which always is involved in evaluation of network performance.

The rest of this paper is structured as following: In following section, we summarize the existing research works on performance analysis. Section 2 provides a brief overview of Flattened Butter Fly and methodology in hybrid approach and features. In section 3, work is clustered in 5 subsections, in first and second sections, It will be observed, how buffer size and routing algorithm are able to improve the performance in wire-based approach.

In subsection 3 we compare wire and wireless on delay aspect. In subsection 4 and 5, is proved that virtual channel and subnet can have a direct role in hybrid approach to enhance the different parameters. It's remarkable that, in each subsection a conclusion is provided independently, and finally, in section 5, the conclusion is presented.

II. Related Research Works

OCI architecture has some important specifications and features so called: latency, throughput, traffic load, energy consumption, and occupied silicon area requirements. These are adopted in SoCs to characterize the performance. ***K. Lahiri and et al***. in [13] have pointed out that the current design tools and methodologies are not suitable for NoC evaluation, and simulation methods, despite their accuracy, are very expensive and time consuming. Therefore, updated techniques and tools are required to extract new communication characteristics, also to estimate the network performance and energy consumption efficiently, and of course area requirements for candidate communication architectures, is always a serious concern. Approaches proposed in the literature can be classified in four main categories: deterministic approaches, probabilistic approaches, physics based approaches, and system theory based approaches. First category is mainly based on graph theory that has been used successfully in many software and computer engineering domains. An example is the work has been carried out by ***A. Hansson*** in [14], a model using a cycle-static dataflow graph was used for buffer dimensioning in NoC applications. Deterministic approaches assume, the designer has thorough understanding of the communication pattern among cores and switches. Nowadays researchers mostly use probabilistic approaches which are based on queuing theory. For example, an analytical model using queuing theory was introduced in [15] to evaluate the traffic behavior in Spidergon NoC. Simulation results have been reported to verify the model for message latency under different traffic rates and variable message lengths. Most queuing approaches consider incoming and outgoing traffic as probability distributions (e.g., Poisson traffic) and allow designers to perform a statistical analysis on the whole system in order to evaluate certain network metrics, (average buffer occupancy and average buffer delay in an equilibrium state). In[16], Unlike stochastic approaches that make Markovian assumptions on the network behavior, statistical physics model the interactions among various components while considering the long term memory effects. The main concept in this model is that packets in the network move from one node to another in a manner that is similar to particles moving in a Bose gas and migrating between various energy levels as a consequence of temperature variations. Network Calculus features [17] are derived from system theory which is placed in the fourth category, this strategy has been frequently applied to electronic circuits successfully. Performance bounds (e.g., end-to-end delay) in networks such as the internet is modeled and evaluated. According to shapes of the traffic flows (by analogy, signals in system theory), designers are able to capture some dynamic features of the network [18], [19].

III. Simulators

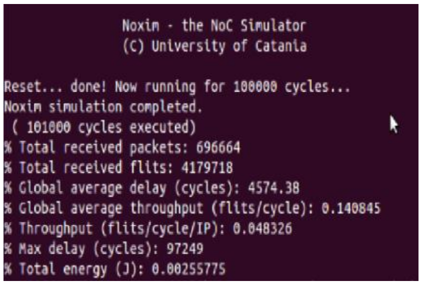
In this section ***BookSim*** interconnection network simulator is described which has is widely used in entire work. The simulator is installed on LINUX OS and is open source. To obtain data and simulate a candidate network, there are some parameters to be set. By changing or resetting input parameters, user can simulate the performance of different NOC systems. It's remarkable that the Noxim simulator setting (Fig.1) is the same, but is applied for wire-based only.

Number of virtual channel, depth of buffer size, topology, traffic model, number of IPs, packet injection rate are the parameters to be adjusted before simulator is ran. The parameters like topology, traffic and virtual channel are configured only once, but for the other parameters are changed depending on the delegated task. Here, some important parameters definition are presented:

Virtual channel is the number of the virtual channels per router;

Buffer size means the depth of buffer base on bit;

Traffic model is defined uniform, means each source sends an equal amount of data to destination. The most two important parameters during this work which are frequently varied are PIR and Subnet. PIR unit is either flit/cycle/node or bit/cycle/node. Each flit in this work, contains 6 bits. For example, if the ***injection rate*** is 0.15, it means that during each cycle, 0.15 bit is injected in every node. Once the parameters were set, simulation is begun. In order to start, type command \*:=booksim* [*configfile*]" in the \\=*src* directory. At the output, following parameters are seen \average latency", \average accepted rate", which means the average throughput, \min accepted rate" and \average hops". The unit of *latency* is cycle; the unit of *throughput* is it/cycle/node. Then, by running the simulation for various \injection rate", an average latency graph can be obtained. In throughout evaluation, simulator is fed with different number of IPs, subnet and PIR rate which uniquely are able to determine the performance [20]. For large (network containing more than 1024 IPs) networks, during each simulation, under high rate of PIR, simulation takes 20 minutes approximately.



```
                Noxim - the NoC Simulator
               (C) University of Catania

Reset... done! Now running for 100000 cycles...
Noxim simulation completed.
 ( 101000 cycles executed)
% Total received packets: 696664
% Total received flits: 4179718
% Global average delay (cycles): 4574.38
% Global average throughput (flits/cycle): 0.140845
% Throughput (flits/cycle/IP): 0.048326
% Max delay (cycles): 97249
% Total energy (J): 0.00255775
```

Fig. 1. Noxim Simulator

IV. Methodology and Approach

A **hybrid** wired/wireless NOC architecture (Fig.2) is used. Hubs are interconnected by both wired and wireless links while the subnets internally are connected via wires only. The hubs those are linked via wireless, have to be equipped with Wireless Base stations (WBs), WBs are responsible to transmit and receive data packets over wireless channels. When a data needs to be sent to a core in a different subnet, it travels from the source to respective hub and reaches to the destination subnet's hub via the ***small-world network*** consisting of both wired and wireless links, then it is routed to the final destination core. For intra subnet and inter subnet data transferring the ***wormhole*** routing is adopted [20]. It is remarkable that data packet is broken down to smaller units called flits.

- ***Adopted Routing***

In proposed hierarchical NoC [20], two topologies are used. In this work, for inter-subnet we consider mesh topology. For an intra-subnet communication if the destination is more than two hopes away, then the flit goes through the central hub to its destination. Thus, within the star-ring subnet, each core is at a distance of at most two hops from any other cores (Fig. 3). To avoid deadlock, the virtual channel management scheme from Red Rover algorithm is adopted, in which the ring is divided into two equal sets of contiguous nodes. Messages originating from each group of nodes use dedicated virtual channels. This scheme breaks cyclic dependencies and prevents deadlock. However, intra-subnet data routing, requires flits to use the upper-level network consisting of wired and wireless links. By using the wireless shortcuts between the hubs with WIs, flits can be transferred in a single hop between them. If the source hub is not equipped with a WI, the flits are routed to the nearest hub with a WI via the wired links and are then transmitted through the wireless channel. Likewise, if for destination hub also a WI is not accessible, then the nearest hub to it with a WI, receives the data and passes it to the destination through wired links. Between a pair of source and destination hubs without WIs, the routing path involving the wireless medium is chosen if it reduces the total path length compared to the wired path. A token flow control (***Kumar***) along with a distributed routing strategy is adopted to alleviate this problem.

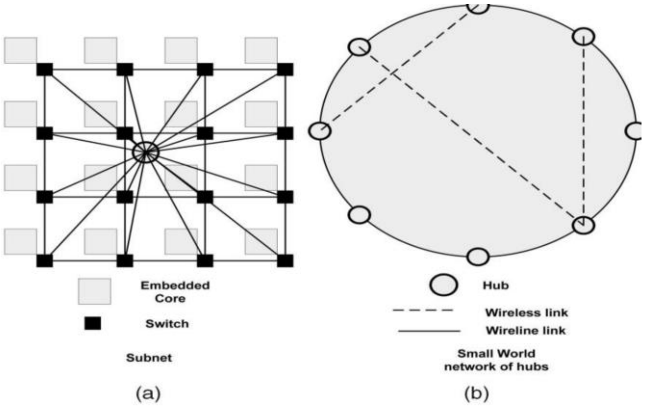


Fig. 2. hybrid structure

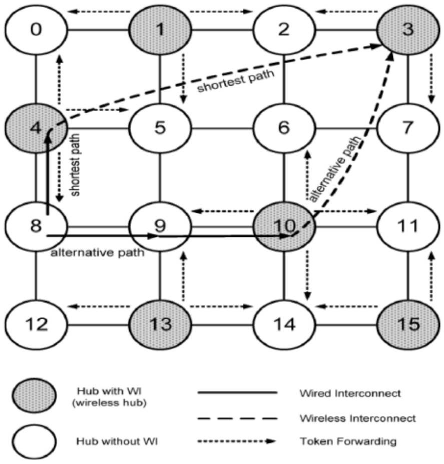


Fig. 3. Adopted Routing in Wireless Communication

- ***Flattened butterfly***

Selecting an appropriate topology in the design of an interconnection network is an important task, because it has a direct effect on critical performance metrics such as network's zero-load delay and its capacity, also directly influences the implementation expense, both in terms of on-chip resources and implementation complexity. In this work wire-based and wireless networks are running under mesh and flattened butterfly topology respectively. Here to clarify how Flattened Butterfly topology impresses on network performance we briefly discuss on it.

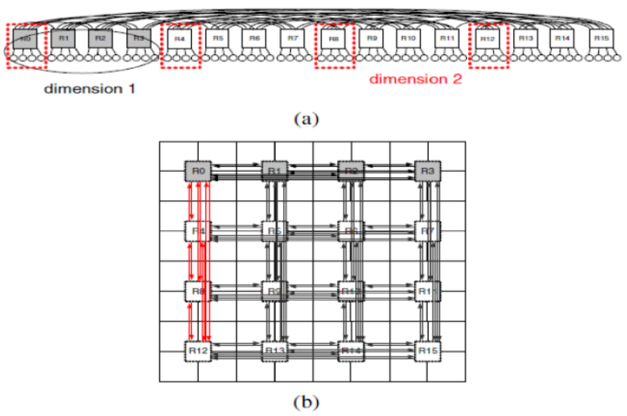
The flattened butterfly topology is a cost-efficient topology in particular to use with high-radix routers [21]. Flattened butterfly is derived by combining (or *flattening*) routers in each row of a conventional butterfly topology while preserving the inter-router connections. In this design concentration is on the routers, therefore flattened butterfly reduces the wiring complexity of the topology significantly, this is resulted in scaling more efficiently.

Now to make it more understandable, readers are referred to a case study and observe how to map a network with 64-node onto the flattened butterfly topology, we collapse a 3-stage radix-4 butterfly network (4-ary 3-fly) to produce the flattened butterfly shown in Fig. 4(a). The presented flattened butterfly has 2 dimensions and uses radix-10 routers [22]. Each router has a concentration factor of 4. It means, four processor IPs are attached to each router. The remaining 6 router ports are used for inter-router connections: 3 ports are used for the dimension 1 connections, and 3 ports are used for the dimension 2 connections. Routers are placed as shown in Fig. 4(b) to embed the topology in a planar VLSI layout with each router placed in the middle of the 4 processing nodes [22]. Routers connected in dimension 1 are aligned horizontally, while routers connected in dimension 2 are aligned vertically; thus, the routers within a row are fully connected, as are the routers within a column. The wire delay associated with the ***Manhattan distance*** between a packet's source and its destination provides a lower bound on latency required to traverse an on-chip network. When minimal routing is used, processors in this flattened butterfly network are separated by only 2 hops, which is a significant improvement over the hop count of a 2-D mesh.

V. Simulation Results

- ***Evaluation of effect of Routing Algorithm and PIR on NOC performance***

At the first step to have a transparent idea about range of delay and throughput in NOC, we start with wire-based methodology. Two parameters which NOC performance is extremely influenced by, are routing algorithm and rate (PIR). Therefore fully- adaptive [23], XY [23] and West [23] are picked as samples to feed the simulator. Once the simulator is modelled and adjusted by any of routing algorithm, it shall be kept fixed and another factor (here in this study PIR) is the only variable studied factor to be fed into simulator. The used traffic model is uniform. For each rate of PIR, simulator with all other fixed parameters is executed once. To start simulation, machine is configured under ***Fully Adaptive*** routing algorithm.



(a)

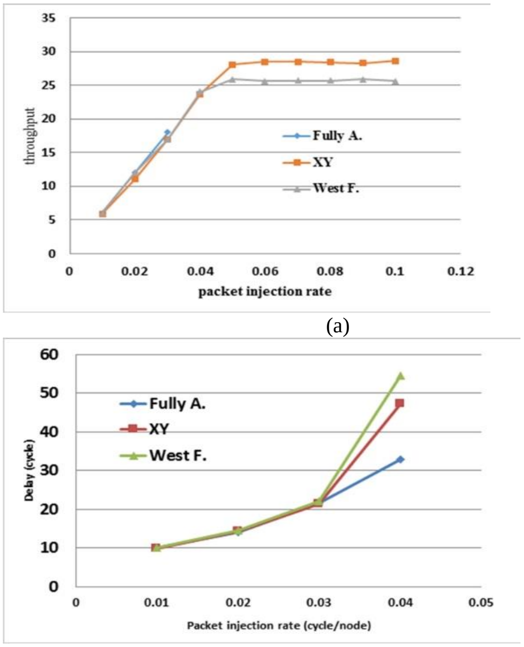
(b)

Fig. 4. (a,b)Flattened Butterfly topology with radix-10

As in Fig. 5(a) is seen when PIR is varied between 0.01 and 0.03, in fact rate of data transferring is increased, thus it's expected total received packets become increased and consequently when majority of network's IPs are involved in communication, thus majority of network is active, this causes an IP sends and receives bits with higher potential capacity, the throughput is improved expectedly. On the other, data traffic in directions is increased due to higher rate of bit communication between IPs, which is led to delay, this latency is not desirable but still is controllable.

This policy is same almost for all three routing algorithms with slightly changes. In first case study, when PIR is greater than 0.03 and less than 0.04, an unusual behavior is observed. Throughput is degraded rapidly, on the other side, the slop of figure for delay is getting sharper. For simulation under PIR>0.05, an interesting result is raised up, the simulator fails and no result is obtained for PIR>0.05, this occurs under fully adaptive routing algorithm (see Fig.5(a)). It justifiable that buffer size which is limited to 8bits in this work has no more capacity to reserve the bits. In fully adaptive routing a packet always is transmitted through a not congested route, with increasing the number of received packets, buffer is filled and there is no more space to receive new packets, this causes a block direction. Throughput is zero when no packet is received and the NOC paradigm under fully adaptive routing algorithm is not applicable for ***PIR>0.04***.

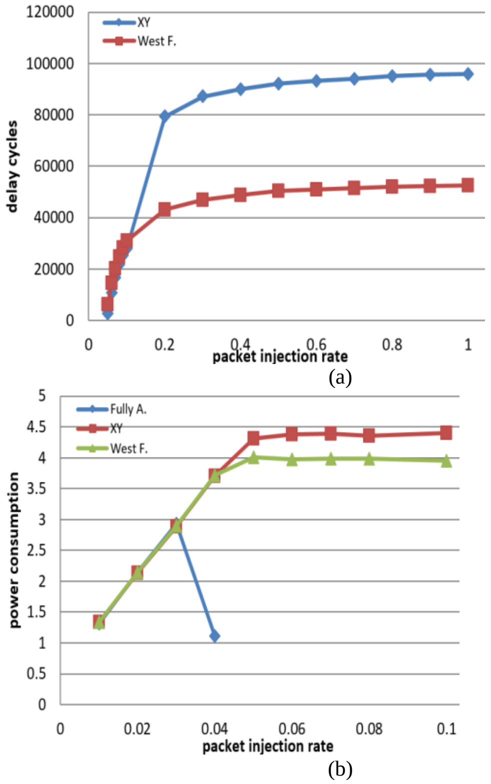
In contrast with fully adaptive algorithm, in XY routing algorithm situation is different. In this routing algorithm when the PIR exceeds a certain value, instead of network failure, the saturation state is occurred. For 0.01< PIR <0.04 the same policy with previous routing algorithm is experienced. At PIR=0.04, all Parameters are in MAX level, but the slop of figure for time latency has become sharper. At PIR=0.05, as is shown in Fig.5(a) still all parameters are increasing but two points clearly are observable. First the rate of throughput enhancement has been reduced. Second point is that the number of delay cycles are increased rapidly and is not under control any



(a)

(b)

Fig. 5. (a) Throughput under three routing algorithm (b) Delay under Fully A., XY, West F. routing algorithms under xy longer. For instance at PIR=0.05 the latency = 2645, and when PIR=0.06, latency = 107770, this time latency is not acceptable. If packet injection rate is kept increasing, the according to Fig.6(a), for PIR>0.05 the saturation is occurred. In saturation state throughput will be varied between 28.1 and 28.7. Delay in saturation region is very sensitive to PIR, the time latency is rapidly increased, but for PIR>0.3 the rate of delay is decreased and sensitivity degree is reduced, this number of delay cycle kills performance. West first routing algorithm follows the very similar policy to XY algorithm, but obtained information from machine are different. For 0.01<PIR<0.04 above under studied parameters are increased, with an almost fixed rate, as is observed in Fig.5(b), at PIR=0.05 time latency is increased with a sharp rate and is out of control. (54.53 at PIR=0.04 to 6240.3 at PIR=0.05). At PIR=0.05 again saturation state is happened.



(a)

(b)

Fig. 6. (a), Full diagram of delay under West F. and XY (b), Power consumption under three routingalgorithms

Number of received packets and average throughput are almost fixed or with a negligible changes. In this case the throughput in saturation area is swing between 25.2 and 25.7.

- ***Delay***

For 0.01<=PIR<=0.03, under three routing algorithms, simulator delivers the same results (Table.1). For PIR>0.03 WEST FIRST routing has higher rate of latency. XY routing algorithm accomplish the better latency than Fully ADAPTIVE. In Fig. 6(a) It's clearly seen that for PIR>0.04 the rate of latency is decreased and slowly is getting close to saturated surface. One more point is that, the saturation peak for xy routing algorithm is higher than west first routing algorithm, therefore the sensitivity of west first to PIR is lower. These obtained results are due to different definition and design of structures in each routing algorithm.

TABLE I. TIME LATENCY FOR F.A, XY AND W.F UNDER DIFFERENT PIR

PIR	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08
F.A	9.9	14.0	21.4	32.9	-	-	-	-
XY	9.8	14.3	21.35	47.1	2645	10777	16693	21739
W.F	10.11	14.44	22.02	54.53	6244.3	14570	20174	24697

- ***Throughput***

According to Fig.5(a) fully adaptive routing algorithm has the best throughput in range of 0.01 <=PIR<=0.03. When PIR is greater than 0.03 west first stands on higher level, in saturation state the throughput of XY is constant and stays in higher level than west first while swing between 28.1 and 28.6. West first routing algorithm is varying between 25.6 and 25.9(Table.2).

In Fig.6(b), network power consumptions are depicted which are provided to give an idea to reader and are not discussed in this work.

- ***Buffer size***

In this subsection effect of buffer size is studied. Buffer size [24] plays an important role in driving of packets in directions and keep them online when there is a heavy traffic, therefore route congestion depends on capacity of buffer. It means that there is a specific minimum size for buffer size to run the network under an assigned PIR without failure. At the first attempt buffer size is kept fixed, then under variation of PIR, simulator is executed and delay and throughput results are recorded. In second step we apply the worst case of PIR which results under it have been obtained to the machine and this time buffer size is considered as input to be altered. In each round size of buffer is extended and the results are recorded for delay and throughput, therefore by comparing the results in both situations, it could be concluded, by extension of buffer size performance is improved. We take two samples to observe how buffer size influences the network performance.

A)0.028<=PIR<=0.030, B.S=4

Size of buffer is configured to B.S=4, execution is started with PIR=0.028, different PIR rate are fed into machine. Delay and throughput are obtained 21.56 and 14 respectively, as we see in Fig.7(a) PIR is gradually increased, PIR=0.030 is the maximum rate which still machine can run under it and enhance the delay and throughput, if PIR is kept rising, network will be failed due to full routes capacity. We record the ***PIR=0.030 as the worst result for B.S=4***, now we extend the B.S from 4 to 6, obtained results say that throughput is improved dramatically by 30 times (2900%) and delay 22.9%. Buffer size can be extended even more but for B.S>6 saturation state occurs (see Fig.7(b)). To sum up, if B.S is incremented from 6 to 16 gradually, delay is reduced only 3.9%and throughput just -0.5%. It is concluded

TABLE II. THROUGHPUT FOR F.A, XY AND W.F UNDER DIFFERENT PIR

PIR	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1	0.2	0.3	0.4	0.5

F.A	6	12	18	1.2	-	-	-	-	-	-	-	-	-	-
XY	6	11.9	17.7	23.7	28.1	28.5	28.5	28.4	28.3	28.6	28.6	28.3	28.5	28.2
W.F	6	12	17	23	25	25.6	25.7	25.7	25.6	25.2	25.6	25.5	25.4	25.4

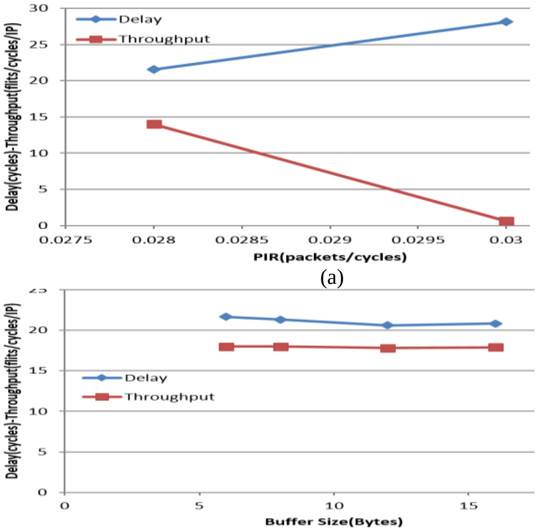
That, the best Buffer Size for 0.028<=PIR=<0.030 is 6. For B.S>6, PIR is not sensitive to B.S extension (Table.3). This can easily be jastified, as the PIR rate and IPs are restricted, therefore the peak of traffic is pre-determinable. It's required to calculate, to match buffer size with maximum traffic to fluent packet travelling. Nevertheless if buffer size would be extended has no mentionable effect on both parameters. Early saturation refers to limited number of IPs in this NOC. Simulator is configured according fully adaptive routing algorithm. b)0.035<=PIR<=0.037, B.S=8,

The policy is same with previous investigation, two steps, PIR is fed into machine under fixed size of buffer, results are recorded for delay and throughput and then the PIR that the worst results are obtained over it, is considered. Now B.S is extended, then the two states results are compared. It's seen in Fig.8(a) that, time latency is intending toward higher cycles and throughput is reducing gradually while PIR is incremented. In this case also results for PIR=0.035, 0.036, 0.037 are as follow:

Delay=27.9, 30.1, 35.3 and *throughput=2, 8, 1*. Maximum PIR rate for B.S=8 is 0.037, in next round B.S is extended to 12(by 50%) in Fig.8(b). The obtained results for B.S=12, 16, 20 are provide below:

Delay = 30, 30.64, 28   Throughput = 22, 22.2, 22.2

As the results show in Table 4 and Table 5, we can come up with this fact that, as the PIR rate is ascended, the rate of buffer size has to be extended in a sharper rate to can compensate the much heavier traffic.



(a)

(b)

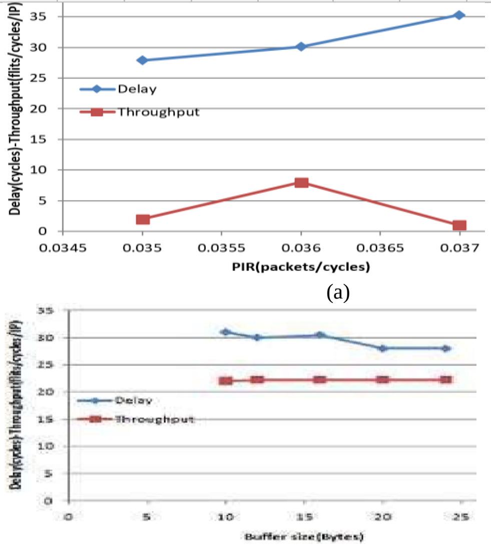
Fig. 7. (a) Delay and throughput under PIR Variant (B.S=4) (b) Delay and throughput under B.S Variant (PIR=0.030)

TABLE III. DELAY AND THROUGHPUT UNDER VARIATION OF B.S AND PIR

PIR	0.038	0.039	0.040	0.041	0.042	0.043	0.045	0.047	0.049	0.050	0.051	0.052	0.054	0.055	0.058
Delay	31.97	34.1	37	37.1	39.9	42.2	52.1	57.6	115.7	91.5	102.5	273	333	488.2	824.2
Thr.	22.7	23	24	24	25	24	26	3.2	1.3	29	30	1	1	0.7	0.2
B.S	12	12	16	24	24	24	24	24	24	72	72	72	72	144	144

TABLE IV. DELAY AND THROUGHPUT UNDER VARIATION OF B.S AND PIR

PIR	0.039	0.039	0.039	0.49	0.049	0.049	0.054	0.054	0.054
Dela y	32.64	32.2	32.7	103.3	77.6	84	310.6	250.3	336
Thr.	23.2	23.3	23.3	8	19	29	2	13	3.6
B.S	18	24	30	30	36	60	108	120	132



(a)

(b)

Fig. 8. (a), Delay and Throughput under PIR Variant (B.S=8) (b), Delay and Throughput under B.S Variant (PIR=0.037)

TABLE V. DELAY AND THROUGHPUT UNDER VARIATION OF B.S AND PIR

PIR	0.058	0.058	0.058	0.058	0.058	0.058
Delay	695.7	1110	650	698.9	703.3	940
Thr.	0.1	3	0.8	1.1	2.2	5
B.S	216	288	350	550	570	630

- ***Comparing Wire vs Wireless in Delay under the Same Conditions***

In this section, it's concluded that it's necessary to switch from a wire-based methodology to a hybrid- combination of wire and wireless. The performance of these two approaches are considered under same conditions. Traffic, No. of IP, virtual channel, buffer size and topology are the parameters which are constant for both states. We execute Noxim simulator for wire-based and Booksim for wireless approach under fully adaptive and butterfly algorithm respectively.

Among many solutions that alleviate effect of delay in networks, extending the number of V.C is justifiable. Wormhole routing with virtual channel flow control is a well-known technique from the zone of multiprocessor networks. While area and power consumption are two major overheads, it allows minimization of the size of the router's buffers and providing flexibility and good channel utilization [22].

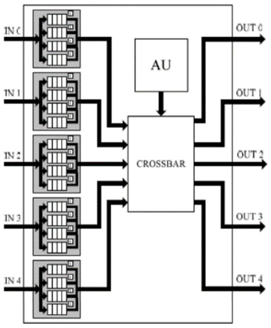


Fig. 9. Structure Virtual channel with 5 I/O

A general structure of a wormhole router with virtual channel flow control is depicted in Fig.9. This router has 5 input/output ports: 4 for connection with the routers those are next to and one to link with the local node. At each input port the virtual channels (VCs), 4 in this example, are de multiplexed and buffering regulation is FIFOs. Status information is saved for every single bit of them. After the bits are out of port through FIFOs they are multiplexed again on a single channel which enters to a crossbar. The operation of the router is controlled by an arbitration unit (AU). It determines on a cycle-by-cycle basis, which virtual channels may advance sooner.

Changing the number of V.C is applicable to both wire and wireless interconnection networks, by using this method, in fact we are dealing with architecture of router. Some modifications have to be completed internally. By doing so, in following section we will observe how the delay is improved:

A) V.C=4, IP=64

At previous subsection delay and throughput in wire-based paradigm and obtained information from the machine were investigated and analyzed. Now we switch to so-called wireless methodology and study same parameters and analyze the obtained results from Booksim simulator under the same conditions. Then these two clusters of information are compared and we come up with a conclusion. In first state, network contents of 64 IPs which is working under 4 virtual channels. Obtained results in Table.6 show that, delay cycles in wireless network is always in lower level than wire-based. In order to justify the reason of better delay cycle in wireless, we understand that, all intermediate IPs between source and destination nodes are eliminated but two and this reduce the time while transferring data between any two distant IPs.

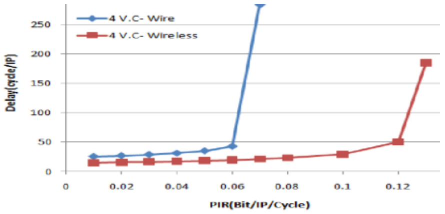


Fig. 10. Delay in wire-wireless networks-64IPS

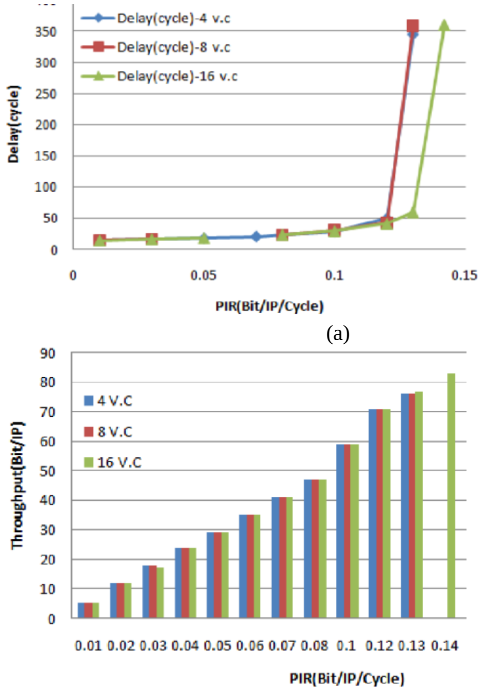
In table.6 we see that under all PIR rates delay is much more reduced for wireless network. The maximum of improvement before take-off point is belongs to PIR= 0.06(54.4%), and the least one is for PIR= 0.01(43%). Primary level of take-off point is reduced effectively, but as it's still high, is not considered. As it's observed in Fig.10, the PIR range is extended because of delay improvement (reducing the number of delay cycle). It's claimed that under the same conditions, delay is improved at least 43%. When all other states are compared too, we come up with the most enhancement state, which is occurred in high PIR rate, thus it's strongly recommended to use wireless paradigm in higher rate of PIR.

In second investigation PIR safe domain is considered. For wire – based, network just is supported to work up to PIR= 0.06, when PIR is exceeded than 0.06, routes are congested and simulator is dump. Now in wireless based, PIR is extended to 0.13 and network can work in wider range, therefore extension of PIR is led to higher performance. In order to get the best performance in wireless methodology, one of the requirement to run the network under wireless methodology is using it under highest PIR rate. With a quick look at statistics, tables and figures, it's figured out that almost the results for both networks in wireless state under 4 and 8V.C, are same, the only slight change is in PIR rate which in wire-based extended from 0.06 to 0.07, and take off point will occur at PIR=0.08. Network with 64IPs is small enough to meet all condition to run simulator properly with no congestion, that's the reason with extending of the V.C to 8, it has no effect on network delay because there is enough capacity to keep the packets in line, to prevent of traffic.

TABLE VI. COMPARING DELAY IN WIRE-WIRELESS NETWORKS

PIR	0.01	0.02	0.03	0.04	0.05	0.06	0.07
W-Delay	25.4	26.7	28.6	30.9	34.8	42.6	283.4
WL-Delay	14.61	15.33	16.22	17.1	18.16	19.47	21.07
Improvement %	43	43	44	45	48	54	93

- ***Evaluation of Delay and Throughput for: 64 IPs under 4,8,16 V.C,*** Researchers in this work are interested the results and efficiency investigation on incrementing of number of V.C, this proves, this inflation has a positive effect on reducing the delay under some circumstances, and improve the performance. If we take a look carefully into Fig.11(a) we observe, with increasing the number of V.C from 4 to 8 and then 16, under variation of PIR, for all no. of V.Cs, the delay is almost unchanged up to PIR=0.12. If PIR crosses the 0.12, two take-off points (345, 358) are seen for V.C= 4, 8 at PIR=0.13, delay is uncontrollable because congested directions have occurred in network. We can see the advantage of V.C=16, when in same network (64 IPs) for PIR>0.12, delay is still under control and network runs well. In Fig.11(a), delay is out of control and occurs at PIR=0.14, it means that blocked direction happens for a network with 16 V.C in each port too, but where the capacity of router has been expanded from 8 to 16V.Cs, this phenomenon is showed up at higher level of PIR. To conclude we must say if it's supposed to network runs under PIR<0.12, it's better to use 4 virtual channel for each port in router, because it costs less and the delay will be the same as well as two other situations. In Fig.11(b) the effect of virtual channel is studied on throughput. Obtained results for throughput are incredibly similar to each other for various number of V.C (4, 8and16) in each PIR. To study behavior of throughput, networks with different no. of IPs are taken into consideration. It's found out that the number of IPs and V.Cs have to be matched for any network, if V.C is much more bigger than what is required for the network, V.Cs are in idle state and are not useable due to not having a data transferring between any two source and destination nodes in low rate of PIR, that's the reason throughput descends, but for PIR above 0.13, when network is nearly getting congested, its anticipated that with having more number of V.Cs, it's possible to prevent a traffic or at least reducing it in network , from there all V.C are used and consequently throughput is improved.



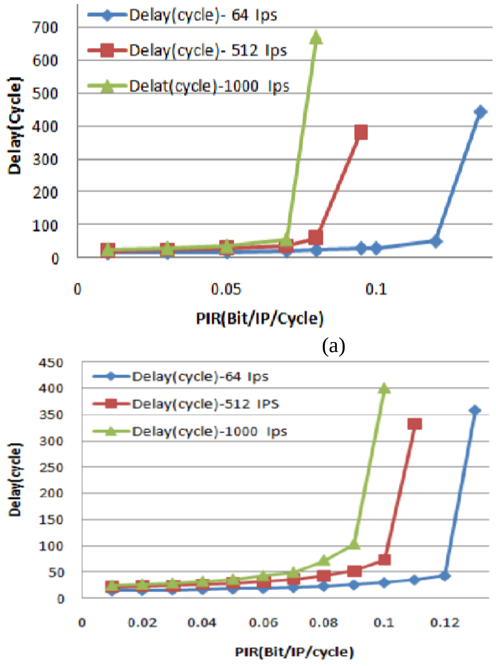
(a)

(b)

Fig. 11. (a) Delay in network with 64Ips (b) Throughput in network with 64 IPs

- ***Comparisian***

In this part we are going to see how large networks cause congestion in routes and consequently heavy traffic, and end up with an unreliable system if wouldn't be managed by different methods. Three networks with diverse numbers of nodes (64, 512and1024) under same conditions are considered to study behavior of delay parameter. If the number of V.C= 4 are kept fixed for all networks (Fig.12(a)), as we expect, the first network that reaches to take off point is larger one. This was anticipatable, because many nodes are involved in communication and directions are filled with bits carrying data. If designer intends to tackle this issue without any physical alteration or change of design structure, he/she has to keep the PIR less than 0.08, and to be in a safe margin less than 0.07 in cost of performance, or else to hold performance still high and delay acceptable, V.C has to be improved and this requires to an alteration in router design.



(a)

(b)

Fig. 12. (a) Delay under 4 V.C (b) Delay under 8V.C

For V.C= 8 (Fig.12(b)) and 16, policy and analyze are similar, the only different is range of PIR, which is extendable by incrementing the number of V.C. In fact we create potential to control delay and even reducing it, to provide this capability, structure of router is changed, but while using this ability, designer has to be aware that, achieving to a desirable delay cycle would be in expense of losing throughput and much more cost. While designing a network, designer needs to know which factor- low delay cycle or higher throughput- is more important to application, then requirements are applied to achieve the best performance.

Basically to get a better efficiency, it's necessary to define for what purpose system is used, because it should be known in trading off between delay and throughput which one should stay in heavy side. Always two policies are observable: first, under equal number of IPs, the network with higher V.C has a lower delay and in high PIR has a better throughput (4<8<16), second, under same number of V.C, a network with lower number of IPs has a lower delay cycle. Throughput is same for different networks under poor rate of PIR, it means that designer has to refuse using much number of V.C under low level of PIR. Using of max. No. of V.C is reasonable if and only if the maximum rate of possible PIR is applied to catch a high throughput.

- ***Subnet***

This subsection is divided to 3 parts and in each part delay cycle and throughput of network for various numbers of IPs (64, 512, 1024), under diverse No. of Subnet (1.4.8.16) is evaluated respectively. Critical points, take off spots and domain of PIR are the cases that will be determined accurately. First all network's cores are divided into multiple smaller cluster of neighboring cores and call these smaller networks subnet. Whereas subnets are smaller than entire network, inter-subnet communication will have a shorter average path length than a single NOC spanning the whole system (Fig.13). Size and number of subnets should not be very large because will affect the throughput and performance of the network [26].

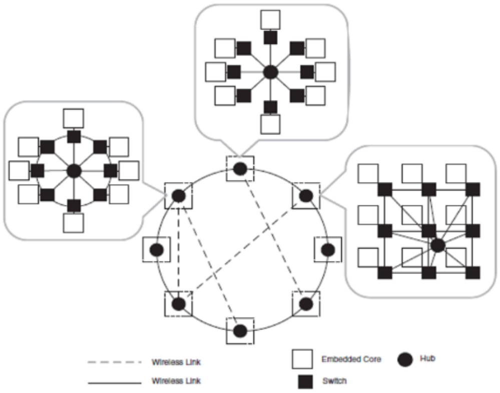
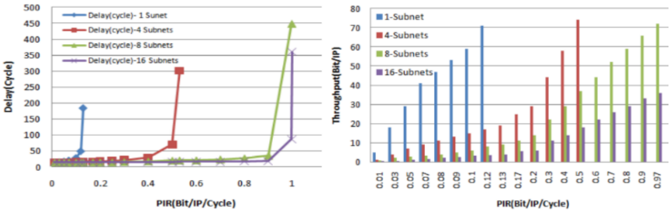


Fig. 13. Clustering of neighboring nodes to subnets

- ***Delay and Throughput for 64 IPs***

In this section concentration is on Subnet. In fact with dividing IPs in a network into different clusters, we intend to decrease the physical distance between any two distant IPs. It's quite understandable that the length of communication of any two remote IPs in a coherent network without any subnets, is much more than a network clustering into smaller group of cores as more intermediate cores are there. There is a possibility that any two cores transfer data internally in a same subnet rather than communication beyond subnets, data transferring is faster because delay cycle is reduced. In this status all switches are linked to a hub, which all other Hubs also from different Subnets communicate in wireless often. This is not discussed in this paper any longer. In first case, a network with 64IPs is considered.

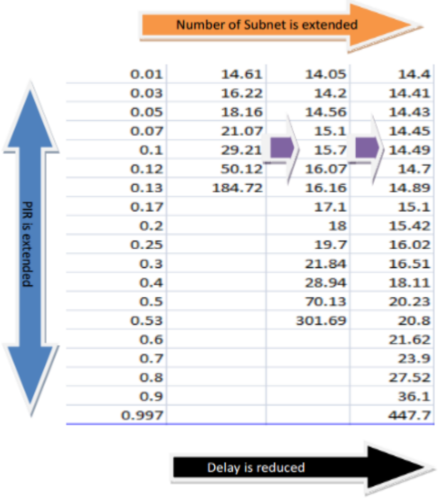
Simulation is started with Subnet=1, region of PIR is limited to 0.12 due to heavy traffic. Delay cycle is constant for any other of numbers of Subnets (4.8.16). In explain, we say if PIR rate is held low and the network is broken into smaller subnets, as the rate of traffic is fixed, it doesn't affect the performance. Now we discuss on throughput for (0.01< PIR<0.12). In spite of delay cycle in Fig.14(b) and Table.8, throughput has the best proficiency for 1= subnet, because running of network in higher level of PIR rate means use of maximum capacity of each node. If we are working with small group of IPs, PIR can be kept in maximum rate with no failure. By developing the number of subnets, network is broken into smaller groups of IPs, this created this potential to run the simulator under higher rate of PIR due to less data communication. When the PIR rate is still low, that is a reason many of IPs are not even involved and throughput would be poor. To reduce the time latency network has to be divided into smaller parts of 4, thus PIR range is extended to 0.5 for 4 Subnets, while PIR is below 0.5 delay cycle is within reasonable range and network works properly. The delay cycle is the same for all number of Subnets like previous case, but throughput is in maximum rate, for 4 Subnets in this range: 0.12<PIR<0.5 due to using all cores in highest possible efficiency. For each No. of subnet there is an authorized range of PIR. Next step is simulation of network for 8 subnets which PIR is between 0.5 and 0.97. We obtain the results for others subnets too, it's concluded that delay cycle is same for subnet=8and16 but as the network has been divided into very small groups of IPs, Subnets=16 has better delay cycle slightly, but we would prefer to ignore it due to some overheads like cost of design and complicated implementation. The best throughput belongs to Subnets=8 in range of 0.5 and 0.97 for PIR (Fig.14(a) and Table.7). A very important conclusion in this subsection is that, using 16 Subnets for a network with 64 IPs is almost insignificant, because it doesn't create an impressive improvement on performance.



(a) (b)

Fig. 14. (a), Delay in a network with 64 IPs (b) Throughput in a network with 64 IPs

TABLE VII. DELAY IN NETWORK WITH 64IPS

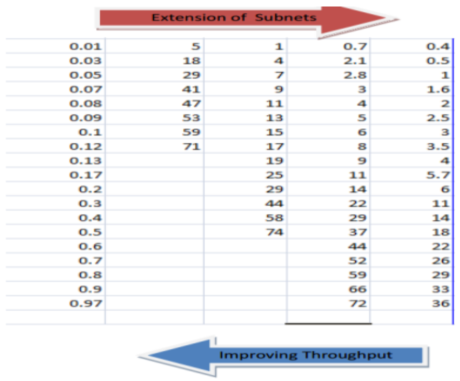

Number of Subnet is extended
PIR is extended
Delay is reduced

0.01	14.61	14.05	14.4
0.03	16.22	14.2	14.41
0.05	18.16	14.56	14.43
0.07	21.07	15.1	14.45
0.1	29.21	15.7	14.49
0.12	50.12	16.07	14.7
0.13	184.72	16.16	14.89
0.17		17.1	15.1
0.2		18	15.42
0.25		19.7	16.02
0.3		21.84	16.51
0.4		28.94	18.11
0.5		70.13	20.23
0.53		301.69	20.8
0.6			21.62
0.7			23.9
0.8			27.52
0.9			36.1
0.997			447.7

- ***Delay and Throughput for 512 IPs***

To be continued, numbers of IPs are extended to 512. In contrast with first case two major differences are observable. In delay Fig.15(a), first the domain of PIR has been reduced, in network with 64 IPs, for 1 subnet, PIR can support the network up to 0.1, now in 512 IPs case, PIR is reduced to 0.09, if we extend the Subnets to 4, 8 and then 16, following results are obtained:

TABLE VIII. THROUGHPUT IN NETWORK WITH 64 IPS


Extension of Subnets
Improving Throughput

0.01	5	1	0.7	0.4
0.03	18	4	2.1	0.5
0.05	29	7	2.8	1
0.07	41	9	3	1.6
0.08	47	11	4	2
0.09	53	13	5	2.5
0.1	59	15	6	3
0.12	71	17	8	3.5
0.13		19	9	4
0.17		25	11	5.7
0.2		29	14	6
0.3		44	22	11
0.4		58	29	14
0.5		74	37	18
0.6			44	22
0.7			52	26
0.8			59	29
0.9			66	33
0.97			72	36

TABLE IX. VARIATION OF PIR UNDER DIFFERENT NUMBER OF SUBNET

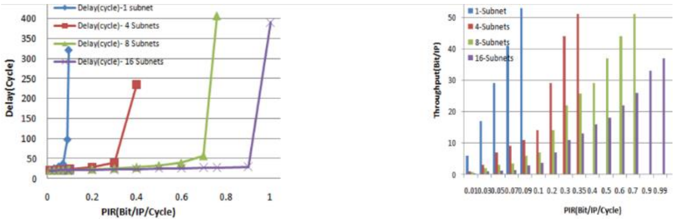
PIR-512IPs	0.09	0.3	0.7	0.99
PIR-64IPs	0.12	0.4	0.9	0.97
SUBNET	1	4	8	16

Results are provided in Table.9, for each number of subnet the maximum authorized rate of PIR is shown. The level of PIR under same number of subnets but different number of IPs is decreased.

It was expected, because for equal number of subnets, more cores are located in small networks. To get a higher performance, rate of PIR is enhanced, due to high data transferring traffic, congested directions are raised up and network is dump. As networks become hugger and more complicated, the numbers of Subnets are increased in appropriate ratio. The relation between Subnets and Delay cycles from one side and the zone of PIR for each numbers of Subnets from the other side, are presented below:

Delay cycle (64 & 512 IPs): 1subnet > 4 Subnets > 8 Subnets > 16 Subnets

0.01≤ PIR (1.S) <0.09≤ PIR (4.S) <0.3≤ PIR (8.S) <0.7≤ PIR (16.S) ≤0.999



(a) (b)

Fig. 15. (a) Delay in a network with 512 IPs (b) Throughput in a network with 512 IPs

For a network with 512 IPs the largest domain of PIR belongs to subnet=8. We discuss on the Fig. 15(a), if we follow the shape of figure for 16 Subnets, it is understood network has less delay cycle in compare with other subnets under same rate of PIR. If quicker data communication in network is the only criteria, designer should go for a highest numbers of Subnets. Now we discuss on Fig. 15(b). Under any circumstance throughput is above 50%, while in case with 64 IPs it was more than 70%. We see that when we have extended the network IPs by factor of 8, (700%), the throughput has reduced just by -20% due to overheads such as power consumption and not using all IPs under maximum rate of PIR. If ignore the range of PIR over than 0.7, in rest of domains we have a competition between Subnets to achieve higher throughput, and always the network with less number of subnets overcomes to others conditions. If level of PIR is increased to greater than 0.7, other players are out of competition one by one, only a network with 16 subnet is supported, if it is supposed to run the network in this range, designer has to use 16 subnets.

- ***Delay and Throughput for 1024 IPs***

The same policy is followed, only two differences are pointed out in contrast of networks with 512 and 64 IPs:

a) Largest domain of PIR has been shifted

When size of network is extended almost twice, the safe margin of PIR has moved to range between 0.05 and 0.9 under 16 Subnets. In a network with 512 cores and 8 subnets, every 64 IPs are clustered in a subnet. We compare it with 1024 cores and 8 subnets, every 128 cores are in each subnet. More number of the IPs leads to heavier traffic in data communication and congested direction in lower level of PIR, therefore to prevent unexpected failure in network with 1024 IPs, we need to extend number of subnets, and this is resulted in higher level of PIR.

b) Throughput has degraded by -10%

We have kept the delay cycle under control in expense of cost and losing throughput. In fact when size of network is extended, number of subnets is increased to still control the delay, but on the other hand for each subnet one Hub is required, thus more number of subnets need more wireless link and more power consumption. It's concluded that sophisticated design reduces throughput. Therefore it's up to user to choose what he needs, whether fast data transferring or high throughput. In this work we have studied the effect of 1,4and8 and 16 subnets on networks with 64,512 and 1024 IPs for PIR variation.

VI. Conclusion

Fast data communication and high throughput are always two very important elements which are led to high performance. To achieve high speed, delay should be controlled in data propagation between source and destination IPs. When PIR rate is varied, buffer channel is congested and it causes a traffic in routes, consequently delay is increased and throughput is reduced. To alleviate this issue, one solution is extending of size of buffer. When switching from wire–based network to wireless, in particular in larger network, delay is improved greatly. In a small network with 64 IPs, 512 and 1024 IPs, time latency is improved by 43%, 63.5% and 69% respectively.

Another observation is, when number of subnets are exceeded from an authorized range, throughput is reduced under lower rate of PIR, and therefore user has to choose whether fast data transferring or high throughput is the priority. During this work and simulation we observed that with developing the number of IPs in network, the PIR safe domain is shifted from low rate of PIR and less subnets block towards extended subnets and higher rate of PIR. When V.C is extended for equal number of IPs, the network with more V.C has a lower time latency and only in high rate of PIR has a better Throughput (4<8<16). Refusing apply of extra number of V.C under low level of PIR for poor rate of PIR is another remarkable point, because it delivers same throughput. By extending V.C from 1 to 8, in a small network, delay is enhanced by 18%, and in a large one, its 10%. As an extra ordinary result, if the Subnet would be extended from 1(wire network) to 16 in wireless network, delay is enhanced by 71% in a small network.

Therefore, it's always recommended, wireless interconnection paradigm, has to be applied under high rate of PIR.

References

[1] Senthilkuamr et al., International Journal of Advanced Research in Computer Science and Software Engineering 2 (9), September- 2012, pp. 103-108, "A Heterogeneous Network-on-Chip Architecture for Scalability and Service Guarantees"

[2] P. P. Pande, C. Grecu, M. Jones, A. Ivanov, R. Saleh, Performance evaluation and design tradeoffs for network-on-chip interconnect architectures, IEEE Trans. on Computer 54 (8) (2005) 1025-1040.

[3] L. Seiler, D. Carmean, E. Sprangle et al., "Larrabee: a many-core x86 architecture for visual computing,"IEEE Micro, vol. 29, no. 1, pp. 10–21, 2009.

[4] L. Bononi, N. Concer, Simulation and analysis of network on ship architectures: Ring, spidergon, and 2d mesh, DATE Proc. (2006) 6 pages.

[5] M. A. Yazdi, M. Modarressi, and H. Sarbazi-Azad, "A load-balanced routing scheme for NoC-based systems-on-chip," in Proceedings of the 1st Workshop on Hardware and Software Implementation and Control of Distributed MEMS, (DMEMS '10), pp. 72–77, Besan, TBD, France, June 2010.

[6] Y. C. Lan, S. H. Lo, Y. C. Lin, Y. H. Hu, and S. J. Chen, "BiNoC: a bidirectional NoC architecture with dynamic self-reconfigurable channel," in Proceedings of the 3rd ACM/IEEE International Symposium on Networks-on-Chip, (NoCS '09), pp. 266–275, May 2009.

[7] M. Coppola, R. Locatelli, G. Maruccia, L. Pieralisi, A. Scandurra, Spidergon: a novel on-chip communication network, Proc. International Symposium on System-on-Chip (2004) 250-256.

[8] F. Karim, A. Nguyen, S. Dey, An interconnection architecture for networking systems on chip, IEEE Microprocessors 22 (5) (2002) 36-45.

[9] S. Suboh, M. Bakhouya, T. El-Ghazawi, Simulation and evaluation of on-chip interconnect architectures: 2d Mesh, Spidergon, and WKrecursive networks, NoCS Proc. (2008) 205-206.

[10] A. Ghasemi , M. Haghi and S. Moradi "An Investigation on the Effects of Subnet Extension in Delay and Throughput in Network-on-Chip" Journal of Circuits, Systems, and Computers Vol. 25, No. 2 (2016) 1650015 (11 pages), Journal of World Scientific Publishing Company DOI: 10.1142/S0218126616500158.

[11] U. Y. Ogras, R. Marculescu, Analytical router modeling for networkson-chip performance analysis, DATE Proc. (2007) 1-6.

[12] Pande P.P. et al. Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures. IEEE Computer, vol. 54(8), 2005.

[13] K. Lahiri, S. Dey, A. Raghunathan, Evaluation of the traffic performance characteristics of system-on-chip communication architectures, VLSI Design Proc. (2001) 29.

[14] A. Hansson, M. Wiggers, A. Moonen, K. Goossens, M. Bekooij, Applying dataow analysis to dimension buffers for guaranteed performance in networks on chip, NOCS Proc. (2008) 211-212.

[15] M. Moadeli, A. Shahrabi, W. Vanderbauwhede, M. Ould-Khaoua, An analytical performance model for the spidergon NoC, 21st AINA Proc. (2007) 1014-1021.August,2011

[16] P. Bogdan, R. Marculescu, Quantum-like effects in network-on-chip buffers behavior, Proc. of the 44th Design Automation Conference (2007) 266-267.

[17] M. Bakhouya, S. Suboh, J. Gaber, T. El-Ghazawi, Analytical modeling and evaluation of on-chip interconnects using network calculus, Proc. of the 3rd ACM/IEEE International Symposium on Networks-on-Chip (2009) 74-79.

[18] Sheraz Anjum, Ehsan Ullah Munir,Waqas Anwar and Nadeem Javaid, Research Journal of Applied Sciences, Engineering and Technology 5(2): 353-356, 2013, "Object Oriented Model for Evaluation of On-Chip Networks".

[19] K. change and et al. Performance evaluation and design trade-offs for wireless network-on-chip architectures ACM Journal on Emerging Technologies in Computing Systems (JETC, Volume 8 Issue 3, August 2012 Article No. 23 .

[20] BookSim 2.0 User's Guide Nan Jiang, George Michelogiannakis, Daniel Becker, Brian Towles and William J. Dally May 7, 2013.

[21] J. Kim and et al. Flattened Butterfly Topology for On-Chip Networks IEEE Computer Society Washington, DC, USA ©2007 Pages 172-182 ISBN:0-7695-3047-8 doi> 10.1109/MICRO.2007.15

[22] Kumar, S. ; Lab. of Electron. & Comput. Syst., R. Inst. of Technol., Stockholm, Sweden ; Jantsch, A. ; Soininen, J.-P. ,IEEE Computer Society Ann Publiched date 24 April 2014ual Symposium on VLSI, " A network on chip architecture and design methodology"

[23] Ville Rantala, Network on Chip Routing Algorithms University of Turku, Department of Information Technology Joukahaisenkatu 3-5 B, 20520 Turku, Finland No 779, August 2006

[24] I. Cidon and K. Goossens. Network and transport layers in networks on chip. In G. De Micheli and L. Benini, editors, Networks on Chips: Technology and Tools, The MK Series in SoS, chapter 5, pages 147–202. Morgan Kaufmann, July 2006.

[25] http://www.diit.unict.it/users/mpalesi/nocarc09/slides/pande.pdf