computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

AN IMPROVED SELECTION TECHNIQUE FOR FAST PACKET ROUTING IN COMPUTER NETWORK

Akinwale A.T. [1)], Adebayo A.A. [3)], Folorunso O. [1)], Adebayo G.A. [2)]

[1)] Department of Computer Science, University of Agriculture, Abeokuta, Nigeria
[2)] Department of Physics, University of Agriculture, Abeokuta, Nigeria
[3)] Department of Computer Science, Moshood Abiola Polytechnic, Abeokuta, Nigeria
aatakinwale@yahoo.com, debamos04@yahoo.com, folorunsosegun@yahoo.com

Abstract: *Contention Awareness Input Selection (CAIS) technique was recently compared with traditional input selection techniques such as First Come First Serve (FCFS) and Round Robin (RR) as a way of improving routing techniques and proved to outperform the two techniques but not widely accepted because of packet starvation problem. In this paper, Contention-Age Input Selection (CAGIS) technique was designed to correct the shortcoming of CAIS. CAIS was modified by including flattening algorithm, treap and queueing theory to improve packet access and control contention age level in the input buffer. The design involved three phases of setting up the network topology using the network objects, initiating of an event scheduler and traffic sources to start/stop transmitting packet through the event scheduler which was implemented using object-oriented tool commands language of script simulation program. The paper concluded that whilst deterministic XY routing scheme and the adaptive Odd-Even routing algorithm were used with CAGIS and CAIS to compare their performance, it was observed that CAGIS performed better under the three synthetic traffic patterns of uniform, transpose and hotspot. The results has improved significantly on contention age level of packets in the input buffer by uniformly distributing them to output ports.*

Keywords: *Routing Techniques, Switch, Throughput, Treap, NoC, Queueing theory, Odd-Even and XY routing schemes.*

1. INTRODUCTION

The increasing demand of the internet capacity is leading the computer network to a fast development in the bandwidth and velocity of the transmission links. Bandwidth has become in the last years a fundamental requirement in the communication and overall in the quality of internet offered services. This problem has been solved to some extent with the development of optical fibres and wavelength division multiplexing transmission technique. Using optical carriers, it is now possible to achieve the range of multi-terabits per second inside the network. An increase in the link speed should be accompanied in performance by the routers and switches that compose the network. If this constraint is not accomplished, these elements become a bottleneck of the system, preventing the internet services demands from being satisfied.

Network-on-Chips (NoCs) have been proposed to support the trend for System-on-Chips (SoCs) integration. A NoC appears as a probably better solution to implement future on-chip interconnection architectures. NoCs present better performance of bandwidth and scalability than shared busses. NoC is a set of interconnected switches, with IP cores connected to these switches. The performance of NoC largely depends on the underlying routing techniques, which have two constituencies: *output selection* (routing algorithms) and *input selection* (arbitration process). Previous research on routing techniques for NoC has been focused on the improvement of the output selection techniques before contention-awareness input selection (CAIS), [1] an input selection technique, used in place of the traditional First-In-First-Out (FIFO) and Round robin (RR) input selections of which the two are fair to all channels but do not consider the actual traffic condition, improved routing efficiency but has it own short coming. With this, we were motivated by many open problems and under-explored topics in this dynamic subject of study which we outlined as follows;

- Bursty traffic is the kind of traffic that is usually considered as a bottleneck in network performance. Congestion occurs when too many packets are injected into the network such that the rate at which they arrived is far greater than the rate at which they are serviced.

- When priority is attached to the input port to have access to a particular output port with the hope of reducing congestion, packets in an input port with low priority are usually starved which violate one of the purposes of replacing System On-Chip Bus by NoC.
- Problem of how to run applications efficiently on the existing platform, which provides better safety and predictability.
- How to guarantee high throughput and low latency with in-order packet delivery.

The motivation of this research project is to improve both input and output selection and develop an effective routing algorithm.

2. RELATED WORKS

Previous work on routing and arbitration process from the network community includes [2], [3] that presented how to design new routing metrics to improve the throughput, and how to modify existing routing protocols to incorporate new metrics. Their models are often built on link-level abstractions of the network without fully considering the impact of the physical layer. There is little if any discussion about the fundamental performance limits, namely Shannon capacity or spectral efficiency. [4] in technical report series 'Evaluation of routing algorithms on mesh based NoCs' considered the relative performance of the algorithms. Their results indicated that in terms of total clock cycle to deliver all packets, deterministic XY routing is faster than three partially adaptive algorithms (i.e. west-first, north-last, negative-first routing algorithms). Partially adaptive can potentially speed up the time to deliver individual packets, but globally the results point out to poorer performance than the XY algorithm. [5] suggested that reducing the number of turns that a message takes may reduce blocking and hence improve performance. [6] answered two questions relevant to the design of NoCs. The first one is the choice of routing algorithm, where XY routing appears as the best one in most situations, for medium to large NoCs. The second question is the determination of the best compromise between packet size and total time to deliver the total load. Medium size packets are the best choice, due to the network buffering capacity. Small packet underutilize the network and increase segmentation and reassembly overhead, while large packet lead to network congestion and buffer saturation.

[7] in 'delay comparison of switching techniques in interconnection network,' the switching techniques were compared to access for better performance and bandwidth optimization. The summary of their work shows that Virtual-Cut-Through (VCT) switching offers message sending as pipelined one. It acts like wormhole in the low traffic load and packet switching in the high traffic loads.

[8] confirmed that store-and-forward and VCT in adaptive routing in (Q)NOC required the receiving switch to be able to store the whole packet before the transmission is allowed to begin. Therefore these approaches were found to be inappropriate for NoCs since buffer space is a scarce commodity in an environment limited by the silicon area and the power budget. He concluded that when the network load is low a deterministic routing becomes superior. [9] suggested that routers capable of dynamically switching between deterministic and adaptive made of operation perform better than their purely static or purely adaptive routers. This approach is known as Dynamic Adaptive Deterministic switching (DyAD). Their simulation results shown that DyAD has superior performance on the entire load spectrum. [10] introduced the used of load ID-tag for flits belonging to the same message, which is updated before the packet enters the next communication link by using a ID-tag mapping management unit. Therefore, flit from different messages can be interleaved, identified and routed according to their allocated ID slots. Their NoC guarantees in order and lossless message delivery with all traffic scenarios using static and adaptive routing algorithms. [11] in Design of a High Performance Buffered Crossbar switch fabric using Network on Chip, proposed two architectural variants: Unidirectional NoC (UDN) and Multidirectional NoC (MDN) architecture with their own "modulo algorithms" compared with the old static routing. Their result claimed to outperform the conventional CICQ architecture with several advantages which includes: speedup, load balancing, path diversity and simpler switch design by allowing simple input memory structure such as First-In First-Out (FIFO). In their system, bursty traffic is the kind of traffic with the worse performance which can be improved upon. Also implementing a good input selection and allowing the disorder of packet are considered as a future work.

The performance of NoC largely depends on the underlying routing techniques, which have two constituencies: *output* and *input selection*. From the literature, it has shown that much research work on routing techniques for NoC has been an improvement of output selection without considering the input selection. The use of CAIS in [12] to replace the traditional First-come First-serve (FCFS) and Round-Robin input selection improves the routing efficiency. CAIS decides which input channel obtains the access depending on the *contention level* of the upstream switches, when there are contentions of multiple input channels competing for the same output channel, which in turn removes possible network congestion. Our research focused on the

modification of Contention-Aware Input Selection (CAIS) to further improve routing efficiency for network-on-chip, particularly on the work of in addressing congestion at the input port.

3. HOW CAIS WORKS

With CAIS, NoCs with 2D mesh topology was considered as shown in Fig. 1 below. The cores represent the terminals connected to the switches using directed arrows.

From Fig. 1 that represents the NoC, path to be taken from source core to destination core is determined, which can be deterministic (XY) or adaptive routing (odd-even, west first, north last, e.tc.) if criterion for routing is *how path is defined*. Other criteria includes: *where the routing decision are taken* and *the path length*, all of which are referred to as *Output selection techniques.* Fig. 2 represents a switch, within which we have the input port *1,2,…,n* and output port *1,2,…,n.* Each of these ports has an internal buffer to store intermediate packets arriving to the port from upstream switches which will be queued on arrival. Between the input and output ports is the switch fabric (i.e. crossbar) that performs the arbitration process *(i.e. input selection)* where we laid more emphasis upon for improvement.

In CAIS, wormhole switching 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 no 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.

CAIS was combined with an output selection, both deterministic and adaptive, to complete the routing function. Firstly, the XY routing was 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 packet from (0, 3) to (2, 2) will take a path as follows: (0, 3) → (1, 3) → (2, 3) → (2, 2). The wormhole switching is sensitive to deadlock. To avoid deadlock, the minimal odd-even (OE) routing was 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, packet from (0, 3) to (2, 2) has two alternative paths: (0,3) → (0,2) → (1,2) → (2,2) and (0,3) → (1,3) → (1,2) → (2,2). Note that the path (0, 3) → (1, 3) → (2, 3) → (2, 2) is invalid because an east-south turn is not allowed in the switch position at (2,3) to avoid deadlock.

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 switches 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 congestion.

Now consider the input selection. Packets p1 at (3, 2) and p2 at (4, 3) both want to travel through (3, 3). In this case, a good choice would be let p1 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, CAIS was 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 (CL) 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.

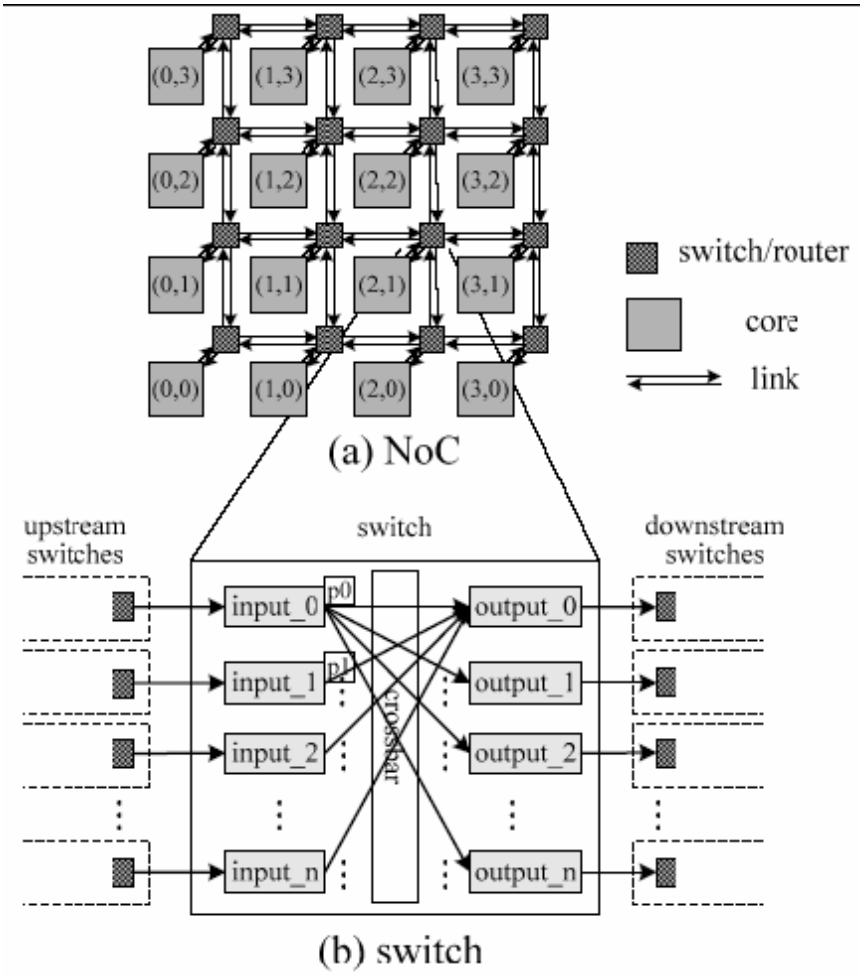

Fig. 1 – Block diagram of Network-on-Chip (NoC) and switch

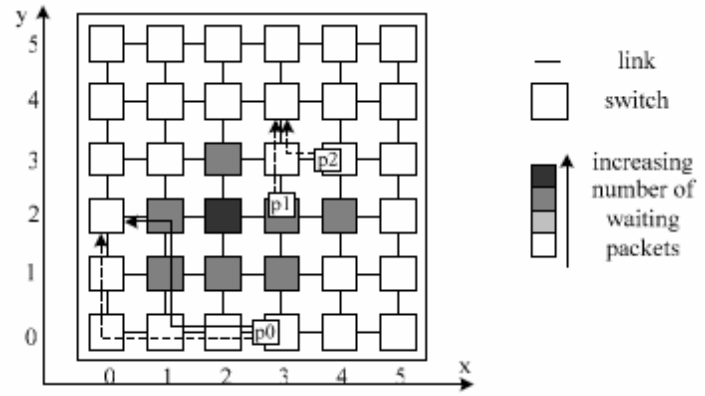

Fig. 2 – Motivation of Contention Awareness Input Selection

3.1. CAIS'S PROBLEM

Although the experimental results shown that CAIS is effective as an input selection technique as far as NoC is concerned, but the **starvation** possibility in CAIS had made it not to be generally utilized with recent output selection techniques. [13], [14], [15], etc, made use of the traditional FCFS as their input selection technique for implementing their routing techniques (output selection) and referenced CAIS as related work with starvation problem.

4. PROPOSED METHODS

The work proposes queueing theory, flattening and treap algorithms to solve starvation problem at the imput selection. The packets to be transported through the NoC observes the theory of queue which has influence on the throughput and the latency of

transmission. Flattening algorithm is the process of eliminating hierarchical constructs, like subclassing and object composition, and producing a flat set of variables and equations that can be interfaced with numerical algorithms. Input ports with individual queues that appears to be numerous in a switch in which two or more of these ports may be contending for a particular output port can be flattened to be treated as single queue.

Considering the term Treap, derived from 'Tree heap', given a set of elements and associated priorities as shown in the sample data in table 1 that represents number of packets in the input ports with their associated failures from contentions respectively, it is not completely obvious that one can construct a treap from these data and still satisfies both invariants simultaneously. Clearly the root of the treap must be the node with highest priority. To satisfy the Binary Search Tree (BST) invariant, all the nodes whose keys are less than this node must be in the left subtree of this node, and the nodes whose keys are greater must be in the right subtree as depicted in Fig. 3. Therefore, we can apply this tree construction recursively to the left and right subtrees, resulting in a treap which will now determine the port that will be given access for a particular output port with their priorities.

Applying treap to our work, each time access is granted to a port or packet arrives, the algorithm will follow the same strategy as in red-black trees. It will find the unique leaf where the element can be inserted while preserving the BST invariant and rotate to enforce the heap ordering invariant.

Table 1. Samples of packets in queue in different input ports

Contention level (*K*)	5	1	3	10	8
Age (*priority)*	0	2	4	7	9

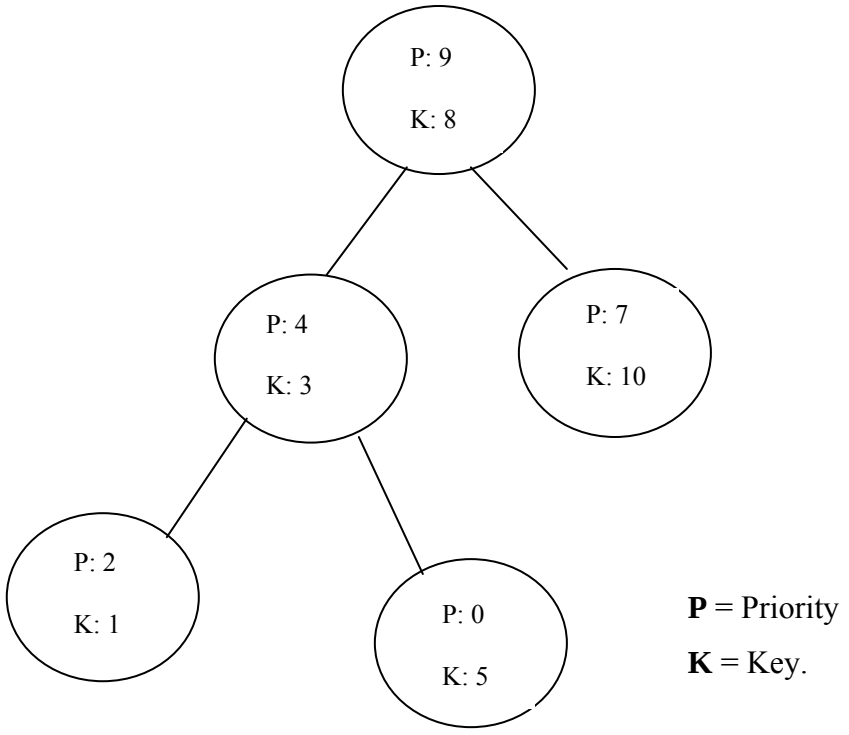

Fig. 3 – A treap with numerical keys arranged in in-order and priorities with heap order invariants.

5. CONTENTION-AGE INPUT SELECTION ARCHITECTURE

We choose XY routing scheme as a representative of deterministic routing scheme, which is a minimal path routing algorithm and free of deadlock and livelock. In addition, compared to other adaptive routing algorithm without virtual channel support, odd-even and DyAD provide more even adaptiveness. So, odd-even routing algorithm was chosen for adaptive routing scheme as being implemented in CAIS. The platform under consideration is composed of a n*n array of tiles which are interconnected by 2D mesh network. Fig. 5 illustrates the detailed architecture of a switch for CAGIS. Each input port in Fig. 4 has a separate small memory (typically the size of several flits) which buffers the input packets before delivering them to the output ports.

Theoretically, multiple input channels requesting simultaneously the access of the same output channel, e.g., packets p0 of input_0 and p1 of input_1 can request output_0 at the same time. The input selection of CAGIS chooses one of the multiple input channels to get the access. Similar to CAIS, the basic idea 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 conditions of the upstream switches. More precisely, each output channel within a switch observes the contention level (CL) (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 will be granted to the input channel which has the highest contention level acquired from the upstream switch. This input selection will remove possible network congestion by keeping the traffic flowing even in the paths with heavy traffic load, which in turn improves routing performance.

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.

Considering the above selection mode, if we limit the selection priority to contention level alone, there will be starvation of packet for input port with lower CL continuously competing with channels that have higher CL. Therefore, we designed CAGIS to consider another priority parameter in addition to CL, with the name of AGE (i.e. number of failures or attempts for a particular output port, not time of arrival) in a way that input channels with low CL and high AGE have the opportunity to win. So, for every input channel, access will be granted to an output port by measuring the priority of a compound of CL and AGE. The initial value of AGE for every channel is zero. When some input channels compete with each other to achieve a specific output channel, finally only one channel will succeed.

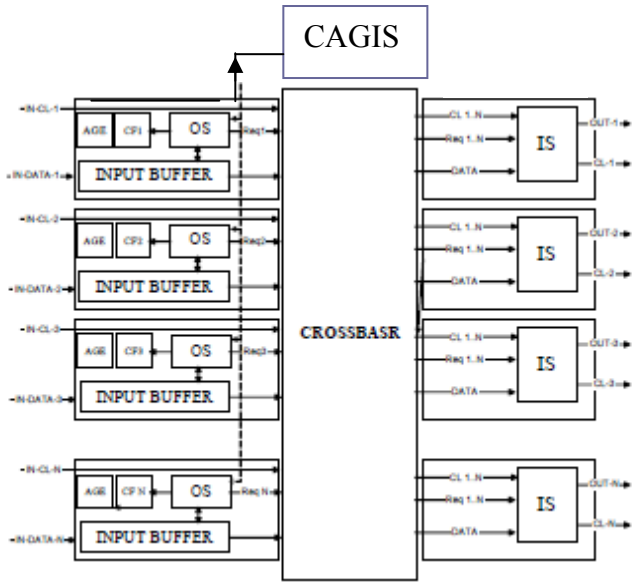

Fig. 4 – Contention-Age Input Selection Architecture

After this, the AGE of the winner channel will reset to zero and AGE of the other channels entering in the competition will increase for one unit. With this new criteria each time that an input channel compete with other input channels to achieve specific output channel, in case of failure, it's AGE increase one unit and this increases its priority for the following competitions and this itself increases the opportunity of success. Finally this channel would be able to gain its desired output channel.

Flattening the input ports contending for a particular output port after the first attempt called for the application of Treap-algorithm that made the CL the *key* and AGE the *priority*. The treap of these two ordered universes is a rooted binary tree with node set *A* that is arranged in in-order with respect to the keys and in heap-order with respect to the priorities. Through the application of treap, the node with largest priority becomes the root that will be granted access and the allotment of the remaining items to the left and right subtree is then determined by their keys. Some other criteria we considered in addition are stated in Fig's. 5 to 8 which show the pseudo code of CAGIS and treap algorithm.

```
01    process observe_cl(req_0..n)
02    Begin
03        out_cl_i <= number of request to
      the i^th output
04    Process select_Age
05    begin
06        for  j = 1 to M DO
07          age_j = P_j
08    end
09    for i = 1 to N DO
10      for j = 1 to M DO
11       begin
12         Treap_insert (out_cl_i, Age_j);
13         A_ij = root of P_ij
14       end
15     end
16    End
```

Fig. 5 – Pseudo Codes of CAGIS

Other criteria considered for granting access to an output port for our implementation are as follows:

- Granting access to an output port not to be based on time of arrival of packet to avoid FCFS principle.
- A switch can send packet to many output port in any direct except the sender.
- The capacity of the input buffer to determine the contention level.
- Stability in the queue determined by (service time/ inter-arrival time) < 1.
- Packets are not randomly picked but randomly distributed by the mechanism.

The function below represents access grant process.

$$A(i,\ j)\ = \begin{Bmatrix} N\ \ len_j \subseteq cl_i\ \max(age_i)\ \forall_{i,j} \in N,\ M \\ 1\ \ \ i \neq j \end{Bmatrix}$$

where

A = Access grant age = Age of packet in an input buffer awaiting service

cl = input port len = length of packets in the input buffer

i = 1,2, . . ., N j = 1,2, . . . , M

6. IMPLEMENTATION ENVIRONMENT

The design is implemented using object oriented tool command language (OTcl) on Network Simulator version 2 (NS-2). As shown in the simplified user's view of Fig. 9, NS is an Object-oriented Tcl (OTcl) script interpreter that has a simulation event scheduler and network component object libraries, and network set-up (plumbing) module libraries.

6.1. FUNCTIONS OF THE DEVELOPED RO ROUTING TECHNIQUE

Using NS-2, the OTcl script that we developed performed the following.

- Initiates an event scheduler.
- Sets up the network topology using the network objects.
- Tells traffic sources when to start/stop transmitting packets through the event scheduler and most importantly, implements the contention-age input selection technique of algorithms of treap, flattening, queue, odd-even and XY, etc.

6.2. THE NETWORK ANIMATOR

The Network Animator (NAM) provides the visual interpretation of the flow of packets in the network and other task listed above. Fig. 10 is an interface that shows transmission of packets using the contention level and age as the key and priority respectively for access to node 4 through the gateway that we used as our "arbiter".

```
Procedure Treap_Insert ( (cl, Age) : item, T : treap)
 If T = tnull then
   T ← NEWNODE ()
   T → [key,priority,lchild,rchild] ← [cl,Age, tnull,tnull]
   else if cl < T → key then
      Treap_insert ((cl, Age), T → lchild)
            if T → lchild→priority > T → priority then
               Rotate_Right(T)
              else if if cl > T → key then
      Treap_Insert Treap_insert ((cl, Age), T → rchild)
            if T → rchild→ priority > T → priority then
               Rotate_Left(T)
 else (* key cl already in treap T *)
```

Fig. 6 – Procedure for Treap insertion

```
Procedure Treap_Delete (cl : key, T : treap)
      If tnull → key ← cl
        Rec_Treap_ Delete(cl, T)
Procedure Rec_Treap_Delete (cl : key, T : treap)
      If cl < T →key then
        Rec_Treap_Delete(cl, T → lchild)
      else if cl > T → key then
        Rec_Treap_Delete(cl, T → rchild)
      else Root_Delete(T)
Procedure Root_Delete ( T : treap)
      if Is_Leaf_Or_Null(T) then T ← tnull
      else if T → lchild→priority > T →rchild→
priority then
                  Rotate_Right(T)
                 Rotate_Left(T→rchild)
            Else Rotate_Left(T)
             Rotate_Delete(T →lchild)
```

Fig. 7 – Procedure for Treap Deletion

```
Rotate_Left( T : Treap)
        [T, T → rchild, T → rchild → lchild] ← [T → rchild, T → rchild → lchild, T]
Procedure Rotate_Right( T : Treap)
        [T, T → lchild, T → lchild → rchild] ← [T → lchild, T → lchild → rchild, T]
Function Is_Leaf_Or_Null (T : treap) : Boolean
        return (T → lchild = T → rchild)
```

Fig. 8 – Procedure for Treap Rotation

Depending on the user's purpose for an OTcl simulation script, simulation results are stored as trace files, which can be loaded for analysis by an external application:

1. A NAM trace file (file.nam) for use with the Network Animator Tool
2. A Trace file (file.tr) for use with XGraph or TraceGraph.

Our OTcl script reflects the diagram in Fig. 11, which describes TCL file implemented on NS producing NAM trace file and Trace file that we used in comparing the performance of CAIS and CAGIS.

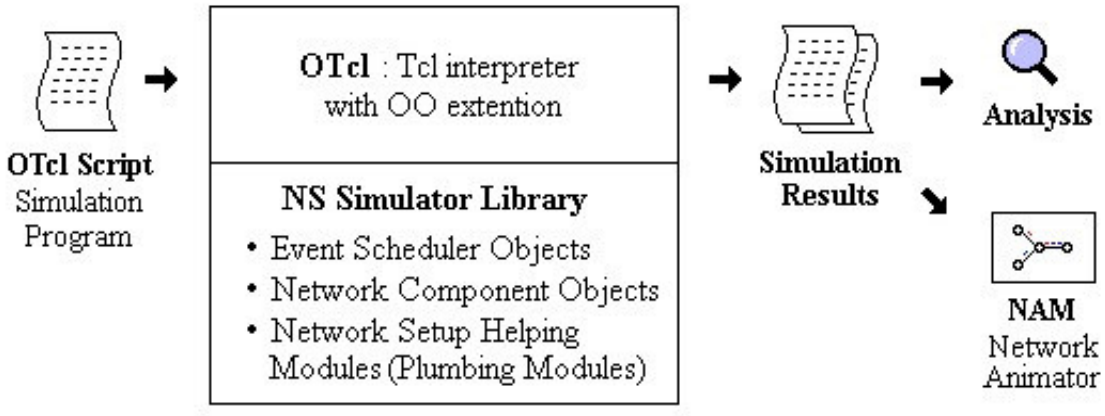

Fig. 9 – Simplified User's View of Network Simulator-2

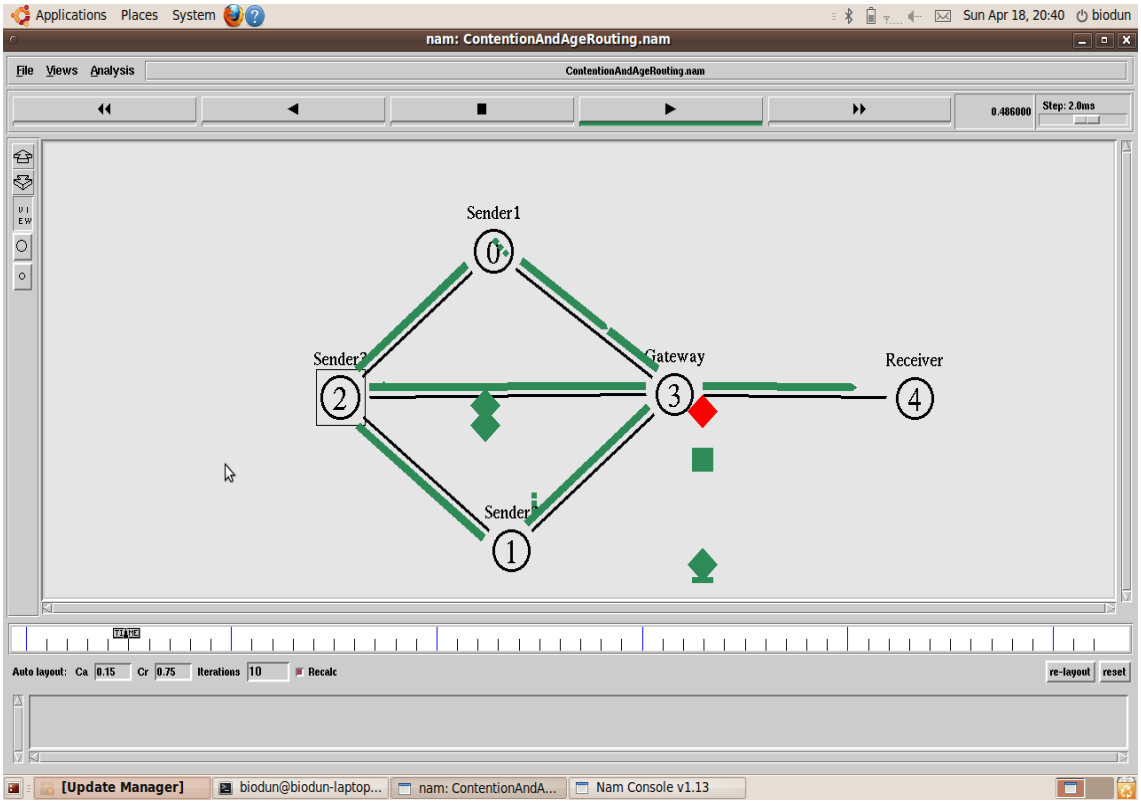

Fig. 10 – Network Animation showing movement of packets in the network

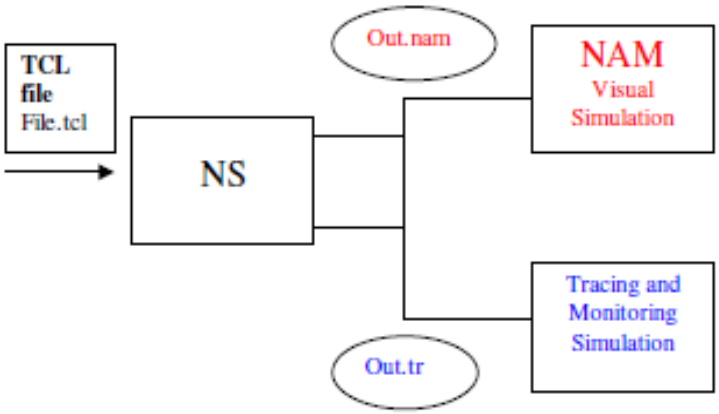

Fig. 11 – Flow of events for a Tool Command Language file run in Network Simulation

7. EXPERIMENTAL RESULTS

Experiments are conducted to evaluate the performance of the contention-age input selection (CAGIS) and give a comparison between CAGIS and contention-aware input selection (CAIS), in which research have shown to be better than traditional input selections, especially when the network is congested. Two traditional input selections have been used in NoC, First-Come-First-Served (FCFS) input selection and round-robin input selection. Due to the outstanding performance of CAIS compare to both FCFS and round-robin, CAIS was selected to compare with CAGIS. Both CAGIS and CAIS were combined with a deterministic output selection (XY routing) and an adaptive output selection (Odd-Even routing). Four switch models were developed using simulator as above to implement the four routing schemes: XY+ CAIS, XY+CAGIS, OE+ CAIS and OE+CAGIS. Simulations were carried out on a 4 x 4 mesh NoC using these four switch models. As in previous work, the performance of the routing scheme was evaluated through latency-throughput chart. Basically, in addition to latency-throughput, our evaluation joined packet starvation to determine the best. For a given packet injection rate (i.e., the number of packets injected to the network per cycle), a simulation was 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 were 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 three different traffic patterns, which include three synthetic traffic patterns (uniform, transpose and hot spot).

7.1. SYNTHETIC TRAFFIC

In the experiments, we considered three synthetic traffic patterns: uniform, transpose, and hot spot. 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 sends packets to the core at (4-j, 4-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. 12 shows the performance of the two routing schemes under uniform traffic. The X-axis represents the packet injection rate per node (the packet injection rate for the whole NoC is 16 times higher), and the Y-axis represents the average packet latency. As shown in Fig. 12, the two as the traffic load increases, the packet latency rises dramatically due to the network congestion. Comparing the chart of OE + CAIS and OE + CAGIS, it can be seen that, using the OE output selection, CAGIS performs significantly better than CAIS. Similarly, the chart of XY + CAIS and XY + CAGIS show that CAGIS also outperforms better than CAIS when using XY output selection, given us the percentage improvement of 47.61% and 44% for OE and XY respectively with CAGIS over CAIS. As reported in [16] 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. 13 shows the performance of the two routing schemes under transpose traffic. It can be seen that CAIS and CAGIS have almost the same performance when using the XY output selection; FCFS will work slightly better than both CAIS and CAGIS 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 with the percentage improvement of 25% and 60% respectively with CAGIS over CAIS on the routing performance. Fig. 14 shows the routing performance under hot-spot traffic. Once again, it can be seen that CAGIS significantly outperforms CAIS, either using XY or OE output selection with the percentage improvement of 47.21% and 76.19%. Furthermore, although the OE output selection performs worse than the XY output selection when using the CAIS input selection, it achieves similar performance as XY when using the CAGIS input selection.

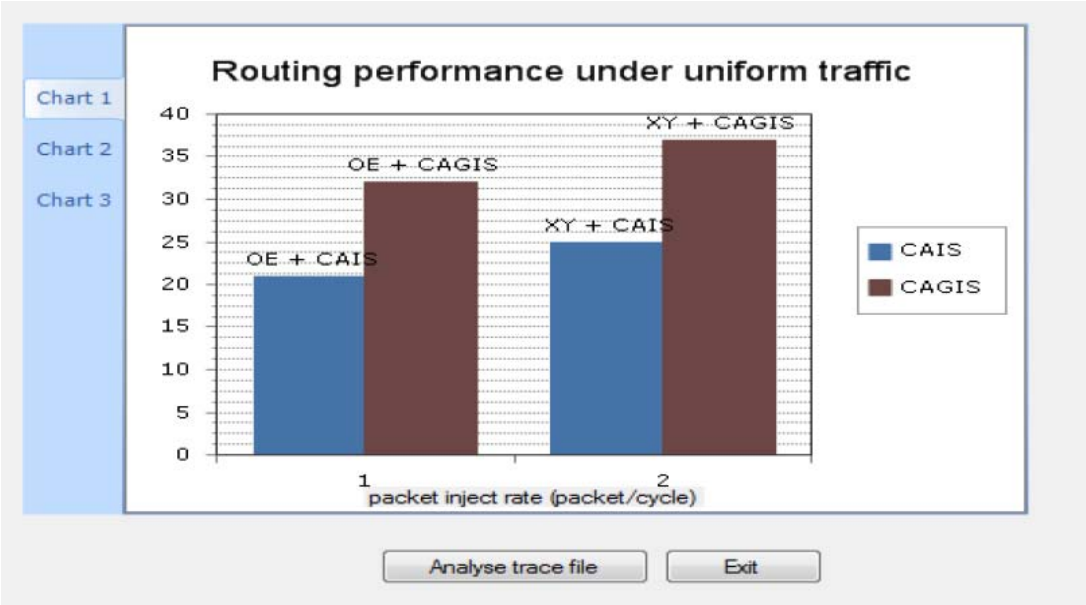

Fig. 12 – Routing performance under uniform traffic

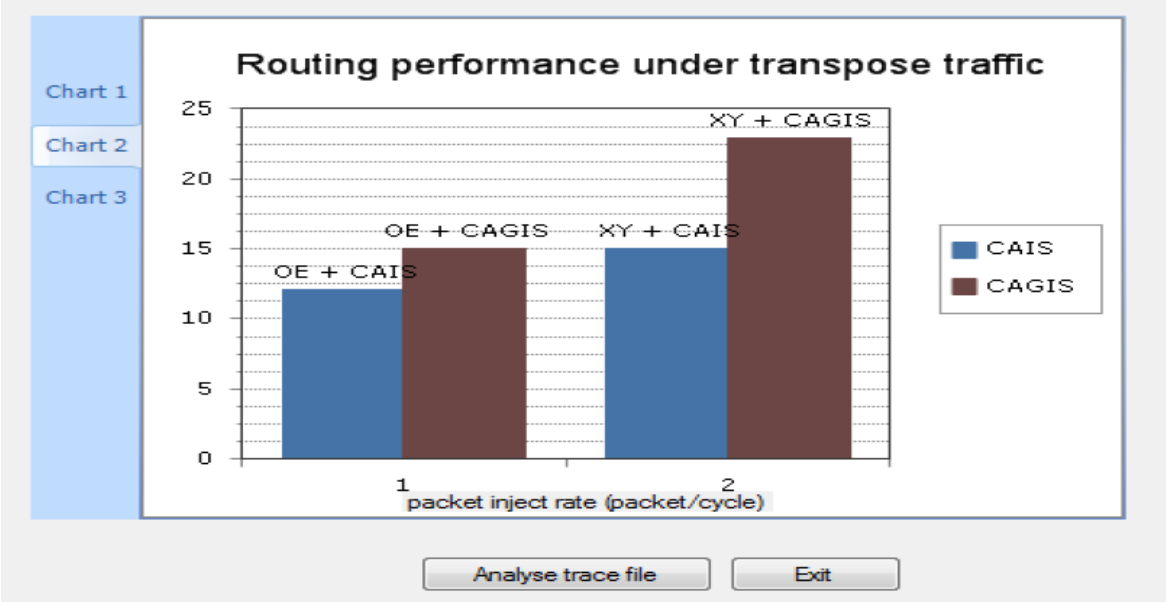

Fig. 13 – Routing performance under transpose traffic

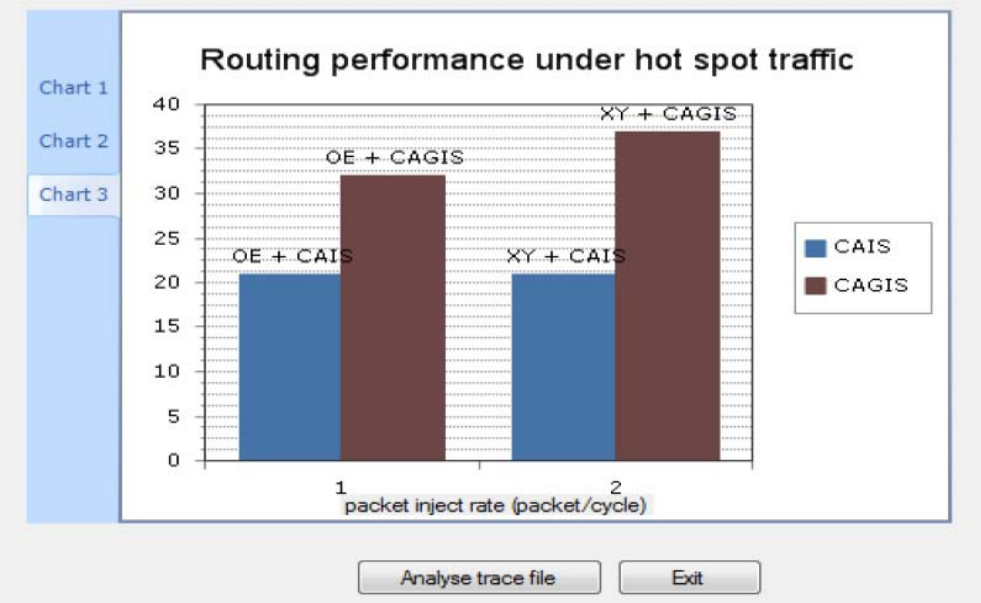

Fig. 14 – Routing performance under hot spot traffic

8. CONCLUSION AND FUTURE WORK

In this paper, we presented contention-age input selection for improving routing technology after investigating the impact of input selection, which consequently improves the routing efficiency. When there are contentions of multiple input channels competing for the same output channel, with the introduction of flattening and treap algorithm in CAGIS, access is granted in comparison with CAIS method, where input channel obtains the access depending on the contention level of the upstream switches alone. Age (number of attempts) of packet in the input buffer is also a priority in addition to CL considered for access by CAGIS. This in turn removes possible network congestion and reduces packet starvation due to low priority assigned to the port. Simulation results with different synthetic patterns show that, when combined with either deterministic or adaptive output selection, CAGIS achieves significant better performance than CAIS.

Some areas that we consider requiring improvement that we did not cover in the cause of our study includes: deeper research in multicast traffics and combining it with unicast flows. Using CAGIS as input selection mechanism with "DyAD – Smart routing for network-on-chip or any output selection that permits swapping form deterministic to adaptive routing based on the behavior of the network traffic. Also, End-to-End flow control should be implemented to have a control of the load in the network when NIs are considered infinite. There is also a need to analyze the general topologies of NOC's and use statistical analyses to evaluate the performance of CAIS as against CAGIS.

9. REFERENCES

[1] W. Dong, B. M. Al-Hashimi and M. T. Schmitz, Improving routing efficiency for network-on-chip through contention-aware input selection, *Proceedings on 11th Conference ASP-DAC*, Japan, (2006).

[2] R. R. Cessa, *Design and Analysis of Reliable High Performance Packet Switches*, PhD thesis, Plytechnic University, 2001.

[3] D. Boer, D. P. Kroese and R. Y. Rubinstein, A fast cross-entropy method for estimating buffer overflows in queueing networks, *Journal of Management Science*, 50 (7) (2004). pp. 883-895.

[4] V. D. Aline, Evaluation of routing algorithm on mesh based NoCs, *Technical report series of GI-edition proceeding of 10th International Conference on Innovative Internet Community Services*, Thailand, (2010).

[5] C. Glass and L. Ni, The turn model for adaptive routing, *Journal of the Association for Computing Machinery*, 41 (5) (1994). pp. 874-902.

[6] D. Xiang, Y. P. Zhang and J. Wu, Deadlock free adaptive routing in meshes based on cost effective deadlock schemes, *International Conference on Parallel Processing*, (2007).

[7] J. Henkel, W. Wolf and S. Chakradhar, On-chip Network: A scalable communication centric embedded system desighn paradigm, *VLSI Design*, India, (2004). pp. 845-851.

[8] Z. Vladimir, Adaptive routing in (Q)Noc, *LNI Proceedings*, (2008). http://subs.emis.de/LNI/proceedings/proceedings165/P-165.pdf.

[9] R. Marculescu and J. Hu, Dyad: smart routing for network-on-chip, *Proceedings of the 41st annual conference on design automation*, ACM, New York, (2004). pp. 260-263.
[10] N. Kavaldjier, G. J. M. Smit and P. G. Jansen, A virtual channel router for on-chip network, *IEEE International SOC Conference*, USA (2004). pp. 289-293.
[11] V. S. Iria, *Design of a High Performance Buffered Crossbar Switch Fabric using Network-on-Chip*, MSc. Dissertation, Department of Electrical and Computer Engineering, Delft University of Technology, (2008).
[12] L. Mhamdi and M. Hamdi, MCBF: A high performance scheduling algorithm for buffering crossbar switches, *IEEE communication letter*, 7 (9) (2003). pp. 451-453.
[13] H. C. Myong, L. Mieszko, S. S. Keun, K. Michel, W. Tina and D. Srinivas, *Oblivious routing in on-chip bandwidth adaptive networks*, Computer science and artificial intelligence laboratory technical report, 2009.
[14] S. Kumar, A. Jantsch, J. P. Forsell, M. Millberg and M. Oberg, A network on chip architecture and design methodology, *ISVLSI*, USA (2002). pp. 117-124.
[15] T. G. Mattson, R. V. Der Wijngaart, Programming on the intel 80-core network-on-chip terascale processor, *Proceedings of the 2008 ACM/IEEE conference on supercomputing*, Piscataway, USA, (2008). pp. 1-11.
[16] G. M. Chiu, The odd-even turn model for adaptive routing, *IEEE Transactions on Parallel and Distributed Systems*, (11) (2000). pp. 729-738.

Adebayo Amos Adeleke *received the B.Tech. degree in Computer Science from the Federal University of Technology, Akure, Ondo State, Nigeria, in 2003, and the M.Sc. degree in Computer Science from University of Agriculture, Abeokuta, Ogun State, Nigeria, in 2011.*

His areas of research are Knowledge Management, Network Optimization, Information Technology, Computer Security and Data Mining. From 2005 till date, he is a lecturer in the department of Computer Science, Moshood Abiola Polytechnic, Abeokuta, Ogun State, Nigeria.

Akinwale Adio Taofiki *received his Magister in Informatics from Warsaw University, Warsaw, Poland, M.Sc./PhD in Economic Cybernetics and Computer Science from Oskar Lange University, Wroclaw, Poland.*

His area of research interest is knowledge database management system, modeling and simulation.

Olusegun Folorunso *is a Senior Lecturer in the Department of Computer Science, University of Agriculture, Abeokuta. He obtained a B.Sc. degree in Mathematical Sciences from the University of Agriculture, Abeokuta in 1992, M.Sc. in Computer Science from University of Lagos in 1997 and a PhD in Computer Science in 2003 from the University of Agriculture, Abeokuta.*

His research interest includes Adoption of Information Systems strategies, Human Computer Interaction (HCI), Knowledge Management, Image Processing and Computational Intelligence. He is a member of Nigeria Computer Society and Computer Professional Registration Council of Nigeria. He has published in reputable international and local Journals.

Adebayo G.A. *is a Senior Lecturer in the Department of Physics, University of Agriculture, Abeokuta. He obtained a B.Sc degree in Physics from University of Agriculture, Abeokuta, M.Sc/PhD in Physics from University of Ibadan, Ibadan.*

His area of research interest is in theoretical physics. He has published in both local and international journals.