Design of TTL Based Routing Algorithm on UTAR Network on Chip Communication Architecture

Wahyudi Khusnandar[1], Fransiscus Ati Halim[2], Felix Lokananta[3]
Computer Engineering Department, Universitas Multimedia Nusantara, Tangerang, Indonesia

Received on March 20th, 2018
Accepted on June 8th, 2018

***Abstract*— XY adaptive routing protocol is a routing protocol used on UTAR NoC communication architecture. This routing algorithm adapts shortest-path first algorithm, which it will not able to work optimally if the closest route have no longer enough bandwidth to continue the packet.The router will store Packet inside and forwarded to the nearest router when closest route has enough bandwidth. This paper suggests TTL based routing algorithm to resolve this issue. TTL based routing algorithm adapts XY adaptive routing protocol by adding several limits on RTL UTAR NoC and added bit in each packet sent by router. TTL based algorithm used more bit and parameters in choosing alternative routes inside the communication architecture. Use of TTL on TTL based routing different from use TTL on communication network. Packets that carry TTL value that equal to Maximum TTL will be route using XY adaptive routing protocol. TTL based routing algorithm has shown better performance compared to XY adaptive routing on some of the experiment done using MSCL NoC Traffic Pattern Suite. This research also proves that TTL based routing algorithm cannot work optimally on small-scaled architecture.**

Keywords—UTAR NoC; XY Adaptive Routing Protocol; MSCL Traffic Pattern Suite; RTL; TTL (key words)

I. Introduction

Mesh and Torus have been selected as topologies in UTAR NoC, because Mesh topology has been considered more by designer due to its simplicity.[1] Most NoC use 2D mesh topologies because of the good tradeoff between cost and performance, XY routing is very common, for mesh topologies, although not standard, due to its property of being deadlock-free.[2] Hence UTAR NoC implements Adaptive and deterministic XY routing.[1] There is a flaw on XY adaptive routing that is being used by UTAR NoC, which is XY routing protocol cannot choose another route other than the shortest path. So if the bandwidth is full in the possible route, the packet will be delayed until the bandwidth is available in that route, although there is an available bandwidth in other route. In this paper, a routing protocol which will using all available bandwidth in a router will be discussed.

II. UTAR NoC Microarchitecture

In UTAR NoC Flit, packet that will be send by Processing Block (PB) will be encapsulated to flit. Flit contains a valid bit, is_tail bit, destination bit, vitual channel bit, source bit, and data bit.[1]

Fig.1 UTAR NoC's Flit

The "valid bit" fields define the validity of a flit. This field will be set as one when sending a new flit. "is tail" field determines whether a flit is a tail or not. A large packet can be broken down into several flits, and a tail flit is the last flit of this group. The destination field provides the information of the flit destination address. The width of this field may vary depending on the size of the network. The Virtual channel field indicates which VC is to be used to send out the flit. The source field provides the information of the flits sender address. The data field contains the data itself. [1]

The UTAR NoC is composed of a number of UTAR NoC Routers. The number of the router depends on the size of the network itself. The UTAR NoC router has five ports connecitons which are the North, East, South, West and PB connections. There are eight pins for each port of the routers:

flit_in pins is for the incoming flit from neighbour router or PB to the router.

EN_flit_in pins is for the enable signal of the incoming flit to the router.

flit_out pins is for the outgoing flit from the router to the next neighbour router.

EN_flit_out pins is for enable signal of the outgoing flit to next router the incoming flit to the router.

credit_in pins is for incoming credit from neighbour router or PB to the router.

EN_credit_in is for the enable signal of the incoming credit to the router.

credit_out pins is for outgoing credit from the router or PB to the neighbour router.

EN_credit_out pin is for the enable signal of the outgoing flit from the router [1].

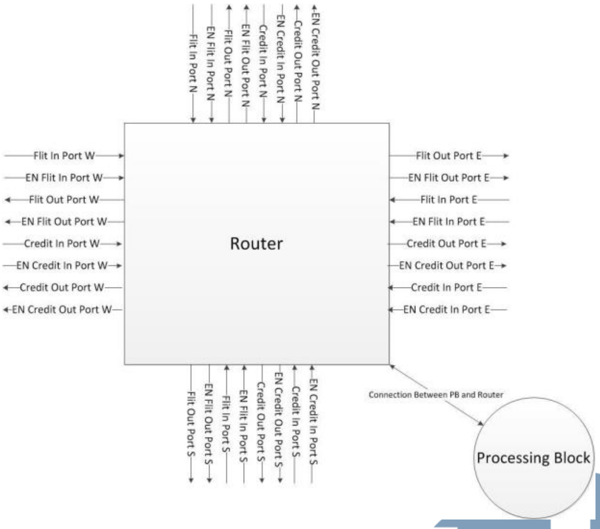

Fig.2 UTAR NoC Router

The router has five input ports and output ports corresponding to the four neighboring directions and the local processing element (PE) port. The major component which constitute the router are input buffer, route computation logic, virtual channel allocator, switch allocator, credit mechanism, and crossbar switch. The UTAR NoC router is input-buffered, in which packets are stored in buffer only at the input ports. The buffers are responsible for storing flits when they enter the router, and housing them throughout the duration in the router. The router computation logic determines all of the path from the source of the flit to its destination. The allocator (VC and crossbar) determines which flits are selected to proceed to the next stage where they travese the crossbar. The credit mechanism takes its role in the flow control protocol. Finally, the crossbar switch is responsible for physically moving flits from the input port to the output port.

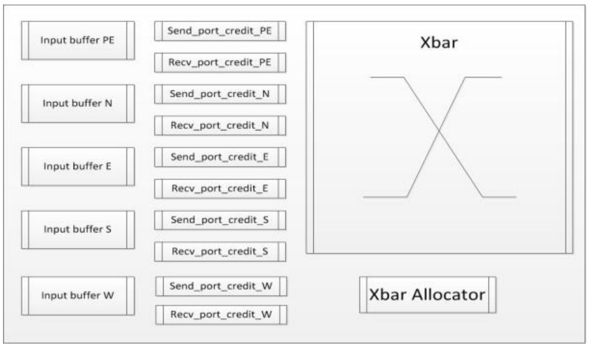

Fig.3 UTAR Router Microarchitecture

III. Adaptive XY Routing

The Adaptive XY routing algorithm is derivative of classic XY routing algorithm [3]. XY routing algorithm will using current coordinates for the router current address(Cx, Cy) and will compare with the destination router address(Dx,Dy) of the packet, depends up on the comparison output of routing algorithm routes the packets. If (Dx > Cx) flit will move to East else, it takes the West turn up to (Dx Cx) become equal this passion is called as horizontal alignment. Now (Dy, Cy) undergo compression, if it is found that (Dy > Cy) then flit moves toward North else flit moves toward South up to (Dy Cy) become equal. When (Cx Dx) and (Cy Dy) are equal, means that packet has reach its destination.

In UTAR NoC, router has information about congestion's condition on each port. The congestion value equal to the remaining bandwidth on each port. The congestion value on port will reduces when router sending a flit throught that port, and the congestion value on port will increases when that port receive a credit from other router.

Adaptive XY routing will be used this information to choose the right path for the packet to go through. Adaptive XY routing works like XY routing, but in addition Adaptive XY routing will check congestion's condition on each posible route. If (Dx > Cx) and (Dy > Cy) but congestion value on X-dimension is less than congestion value on Y-dimension, if we use XY routing usually the packet will be sent to East first until (Dx Cx) are equal, but if we use Adaptive XY routing the packet will be sent to North first due congestion value on X-dimension is less than congestion value on Y-dimension..

IV. The Time-to-live (TTL)

The time-to-live (TTL) is the number of hops that a packet is permitted to travel before being discarded by a router. [4] An IP TTL is set initially by the system sending the packet. It can be set to any value between 1 and 255; different operating systems set different defaults. Each router that receives the packet subtracts at least one from the count; if the count remains greater than zero, the router forwards the packet, otherwise it discards it and sends an Internet Control Message Protocol (ICMP) message back to the originating host, which may trigger a resend. [5]

A specific TTL number can indicate the maximum range for a packet. For example, zero restricts it to the same host, one to the same subnet, 32 to the same site, 64 to the same region and 128 to the same continent; 255 is unrestricted. [4]

V. TTL Based Routing

TTL based routing works differently from adaptive XY routing. TTL based routing at first will works like adaptive XY routing, but when congestion value on all possible shortest path route equal to some value, TTL based routing will give a new route that have better congestion value than all possible route. The route that given will be ensured not the route that packet comes from. TTL based routing using shortest path route to avoid a livelock, and make sure the packet will not wonder too long in the network.

TTL on TTL based routing works like TTL on Network communication. The different is the packet that has reach certain amount of value, the packet wont be dropped, however, the packet will get the top priority to be sent to the shortest path. When TTL on packet has reach amount of value, TTL based routing will work same as adaptive XY routing.

In order to implements TTL based routing to UTAR NoC, some RTL must be changed. UTAR NoC's flit needs to be added TTL field. TTL field can flexibly modified depends on network size and how many times packet allowed to hop.

valid_bit	is_tail	destination	virtual_channel	TTL	source	data
1-bit	1-bit	8-bit	1-bit	3-bit	8-bit	56-bit

Fig.4 Modified UTAR NoC's Flit

VI. MSCL NoC Traffic Patern Suite

MCSL NoC traffic pattern suite is a realistic traffic benchmark suite. It currently includes a set of realistic traffic patterns for eight typical MPSoC applications and cover popular NoC architecture in various scales. MCSL captures not only the communication behaviors in NoCs but also the temporal dependecies among them. Each traffic pattern in MCSL has two versions, a recorded traffic pattern and a statistical traffic pattern. The recorded version provides detailed communication traces for comprehensive NoC studies, while the statistical version helps to accelerate NoC exploration at the cost of accuracy. [6]

The methodology uses formal computational models to capture both communication and computation requirements of applications. It optimizes application mapping and scheduling to faithfully maximize overall system performance and utilization before extracting realistic traffic patterns through cycle-accurate simulations. [6]

MSCL NoC traffic pattern suite will be used to measure the throughput, and latency of both adaptive XY routing and TTL based routing.

VII. Simulation Result

The simulations were performed using mesh topologies with network sizes of 8x8, 4x4, and 2x2 as they represented several topologies that approached the actual network size, and the processing time of the application will be scaled by 100%, 50% and 20%. After both routing is running all the MPSoC application provided by MSCL NoC Traffic Patern Suite and MPSoC application that have been scaled using all the network size that suggested, we will got the comparison result of throughput and latecy value for the both routing on each applications.

In the simulation resulted the percentage comparison of the value of throughput of TTL based routing on adaptive XY Routing (in the pie diagram) shown in figure 5 and the percentage of Latency value comparison between TTL based routing values on adaptive XY Routing (in the pie diagram) shown in Figure 6 as follows:

From Fig.5 we can assume that TTL based routing can finish the application first as TTL based routing has 62.2% better throughput value than adaptive XY routing.

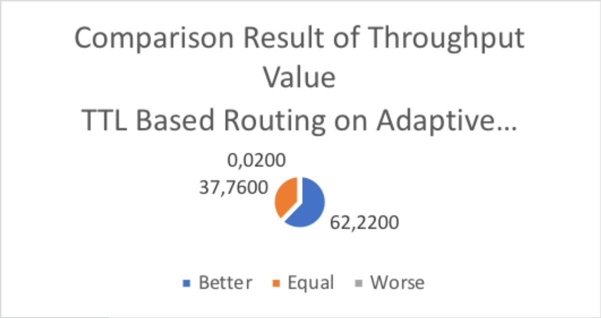

Fig.5 Comparison result (Throughput)

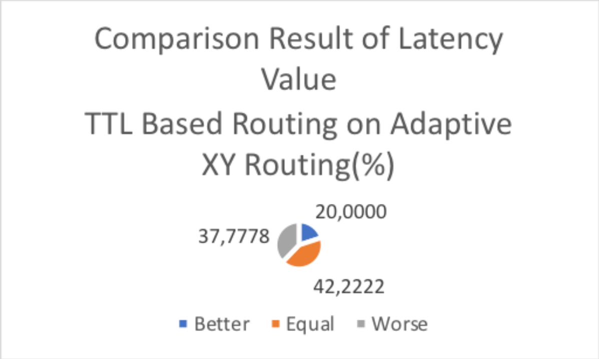

Fig.6 Comparison result (Latency)

This happens because TTL based routing use all available bandwidth on a router so the packet will not be stored on router when the shortest route have no available bandwidth, instead the packet will be sent to the route with an available bandwidth. However, this will cause the latency worse than adaptive XY routing, because the packet will wonder around in the network if the shortest route to the destination ceaselessly has no available bandwidth. As shown at Fig.6 that TTL based routing has 37.8% worse latency value. Due to 2x2 network size test, TTL based routing has 42.2% equal latency value because TTL based routing cannot give more optional route.

References

[1] F. Lokananta, "Network-on-Chip Communication Architecture Design, Analysis, Optimization and Evaluation in a Multi-Processor System-on-Chip," 2015.

[2] E. Cota, A. de Morais Amory and M. Soares Lubaszewski, Reliability, Availability and Serviceability of Networks-on-Chip, Springer, 2011.

[3] S. D Chawede, M. A Gaikward and R. M Patrikar, "Review of XY Routing Algorithm for Network-on-Chip Architecture," *International Journal of Computer Applications,* vol. 43, no. 21, pp. 20-23, 2012.

[4] The Linux Information Project, "Time-to-live Definition," 2005. [Online]. Available: http://www.linfo.org/time-to-live.html. [Accessed 24 July 2017].

[5] M. Rouse, "time-to-live(TTL)," November 2015. [Online]. Available: http://searchnetworking.techtarget.com/definition/time-to-live. [Accessed 24 July 2017].

[6] Jiang Xu, "MSCL Realistic Network-on-Chip Traffic Patterns," 2017. [Online]. Available: http://www.ece.ust.hk/~eexu/traffic.html. [Accessed 25 July 2017].