Dynamic TDM Virtual Circuit Implementation for NoC

Yu Wang[1], Kai Zhou[1], Zhonghai Lu[2], Huazhong Yang[1]
[1]NICS Lab, Circuits and System Division, E.E. Dept., Tsinghua University, Beijing, China
[2]Dept. of Electronic, Computer, and Software Systems, Royal Institute of Technology (KTH)
Email: yu-wang@tsinghua.edu.cn

Abstract— **Quality-of-Service (QoS) communication has been a key issue since the birth of Network-on-Chip (NoC). Time Division Multiplexing (TDM) techniques are well-recognized to be capable of providing guarantees in latency and bandwidth. However, current TDM VC proposals assume synchronous operation and use a costly slot allocation table to statically allocate bandwidth. In this paper, we propose a novel TDM VC implementation that releases the strict synchronous operation assumption and computes slot allocation fully dynamically. To promise VC contention-free, we use the Logical Network theory to guide the slot computation. The slot allocation information is computed once and propagated downstream. This enables adaptive VC configuration and also allows us to use smaller allocation tables in routers. We describe our dynamic TDM VC implementation mechanism, and detail the router design to support this mechanism.**

I. INTRODUCTION

The provision of communication services with well-defined performance characteristics has received significant attention in the Network-on-Chip (NoC) community because for many applications it is not sufficient or adequate to simply maximize average performance [1, 2, 3, 4, 6, 10].

There exist various proposals to achieve QoS communication in on-chip packet-switched networks. The Æthereal [1] and Nostrum [2] adopt a resource reservation strategy to offer guaranteed services. Specifically, both establish *Time-Division-Multiplexing (TDM) virtual circuits (VCs)*. The Mango [3] NoC realizes guarantees in an asynchronous (clockless) network by preserving virtual channels for end-to-end connections and using priority-based scheduling in favor of connections. The QNoC [4] addresses QoS through traffic classification in combination with a differentiated service. It characterizes traffic into four priority classes, and routers make priority-based switching decisions. Using resource reservation strategies can achieve hard performance guarantees but resource utilization may be lower because reserved resources may not be fully utilized. In contrast, priority-based schemes can achieve better resource utilization because resources are used on demand in a best-effort fashion, but they can only provide soft performance guarantees.

In this paper, we focus on TDM VCs. VC is a connection-oriented communication service between communicating entities in a packet-switched network. TDM VC means two or more connections take turns to use shared buffers and link bandwidth using dedicated time slots. TDM VC requires a connection setup or configuration phase by which a deterministic path is established and associated resources are pre-allocated. Specifically, each node along a VC route configures a time-sliced table to reserve time slots for input packets to use output links. In this way, VCs multiplex link bandwidth in a time division fashion. As long as a VC is established, packets sent over it, called VC packets, are transmitted in pipeline encountering no contention and thus have guarantees in latency and bandwidth.

As such, TDM VC makes two assumptions: (1) network routers share the same notion of time. They have the same clock frequency. It may allow phase difference but the difference must be strict [5]; (2) buffer and link allocations are coupled, meaning that the success of buffer allocation is equivalent to link allocation, and vice verse. Nevertheless, Deep-SubMicron (DSM) effects have suggested the difficulty to maintain synchronous operations in a large region of a future chip [9]. This calls for an urgent need to release the synchronous operations for on-chip TDM VCs. The coupled buffer and link allocation rely on a pre-configured slot allocation table to guarantee contention-free and ensure bandwidth. However, this static decision on the content and size of slot allocation tables is difficult to deal with dynamic application traffic, for example, spawning threads, migrating tasks or adding new functions. Hence flexible dynamic TDM VC configuration is desired for many NoC applications. This is especially true when NoC is used as a multi-core computing platform, where traffic is hard to be assumed static [10]. In this paper we address the two issues by proposing a novel TDM VC implementation. This is a dynamic mechanism by which slot allocation is computed online rather than off-line and propagated downwards. The network routers operate with the same frequency but the phase difference can be arbitrary, making it feasible to be implemented in DSM technology.

The rest of the paper is organized as follows. We outline the related work in Section II. In Section III, we discuss the operation of TDM VC and its possible implementation options. Then we detail the dynamic TDM implementation in Section IV, where we present our router design to support the dynamic mechanism. Simulation results are shown in Section V as a proof of the feasibility of our proposal. Finally, we conclude the paper in Section VI.

II. RELATED WORK

Both Æthereal [1] and Nostrum [2] have implemented TDM VCs. They have subtle difference. The Æthereal VC, which is developed for a network using buffered flow control, is open-ended. The Nostrum VC, which is designed for a network employing bufferless flow control, is closed-loop. Nevertheless, both share the

* This project is supported by NSFC 90707002, 60506010. It is also partly supported by Basic Research Foundation of Tsinghua SIST (No SIST2017), MSRA project (No FY08-RES-OPP-026).

same implementation principle, i.e., using statically constructed slot allocation tables to promise latency and bandwidth guarantees. They can accommodate semi-dynamic scenarios where TDM VCs may be configured offline for multiple use cases and the configurations may be loaded according to the switching of use cases at run-time [6].

In [7], another TDM router implementation is presented for asynchronous concatenation of routers. But it also relies on a statically configured slot allocation table to promise bandwidth. In order to accommodate asynchronous operation of routers, it requires a large number of buffers to bridge different clock domains.

Dynamic TDM VC has been shown advantageous than static one in [8]. However, this paper only presents a dynamic slot-allocation algorithm to perform both routing and allocation of slots at run-time to establish guaranteed connections.

We present a dynamic TDM VC implementation, which retains the merits of TDM VC, relaxes the fully synchrony assumption, and also allows dynamic computation of slot allocation. In contrast to [1, 2, 7], our proposal is a dynamic one. In comparison with [8], we present an implementation instead of an algorithm. To our knowledge, we are the first to present dynamic TDM VC implementation.

III. TDM VC CONFIGURATION

We first describe TDM VC and its static configuration, and then describe our dynamic VC configuration mechanism.

A. TDM VC and Static Configuration

The on-chip TDM VCs assume that the network is packet switched and time-slotted. The network nodes share the same notion of time. VC packet routing is performed by looking up slot allocation tables in nodes along the VC path.

In static TDM VC configuration, the slot allocation tables are pre-configured at design time. The entries for the routing tables are globally orchestrated such that no simultaneous use of shared resources is possible. A slot allocation entry essentially defined a *switching rule* for packets from an input link to use an output link. Thus no run-time scheduling is necessary. VC packets synchronously advance one hop per time slot. Since a VC packet encounters no contention, it never stalls, using consecutive slots in consecutive switches.

B. Static TDM VC Configuration

We focus on the slot allocation mechanisms, which are previously static – based on a large slot allocation table containing the slot allocation information for each packet that will pass the router. Taking a 5-port router as an example, the router has five output links and five input links (North, East, South, West, and Network Interface). When a flit comes into the router from the south input link, the router should decide to which link the flit will go. If the data path is selected, a switching rule and time slot distribution method for each link of all the routers should be defined. As several virtual circuits share the same input or output links, the switching rule changes from time to time for each link. Therefore, during the routing phase, all the switching rules and time slot allocation should be calculated and then stored in the time slot allocation table in each router.

Although the static TDM VC configuration using static time slot is easy to implement, excessive memory will be a very large overhead especially when the traffic is more dynamic. Also, if the slot allocation is static, adjacent routers should be kept strictly synchronous to avoid time slot synchronization errors.

C. Dynamic TDM VC Configuration Mechanism

In this paper, we resolve the problem in a dynamic way: let each router or link to decide the time slot allocation according to the data packet header; the time slot information is packed into a time slot packet, and then transferred to the downstream links as the switching rule. Hence, each router changes the time slot information dynamically according to traffic. We will introduce the time slot information first.

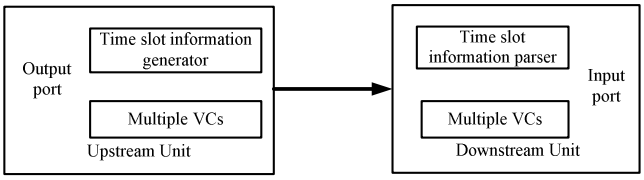

Figure 1. Time slot generation and transmition from the upstrem unit to the downstream unit.

Time slot information is generated in the upstream unit and then transmitted to the downstream unit as shown in Figure 1. When the traffic changes: a certain data packet's transmission process is going to finish, or a new VC is about to established, the bandwidth distribution of a certain link will change. Consequently, the upstream transmission unit shapes the distributed time slot according to the traffic change dynamically and generates the new time slot information at the output link. At the same time, the upstream transmission unit sends the time slot information to guide the downstream transmission unit in order to adjust the switching rule according to the new time slot information. The switching rule guides the downstream unit on how to transmit the receiving data flits. The data source has no responsibility to generate the time slot information, and the time slot information generation logic is integrated in routers.

In this way, the router does not need a static slot allocation table. It requires small memory to store the time slot information. It is possible to prolong the time in which the data is predictable and allow a not very strict synchronization between the upstream and down stream units. Therefore, the system becomes more robust.

The time slot generation needs to take a cycle or more to generate time slot information; however, this overhead can be hidden by the data transmitting latency. When a new VC is about to established, the upstream transmission unit generates time slot information to inform the downstream transmission unit the switching rule. Since each unit has different VCs, it is necessary for each unit to generate its own time slot information. The data flits transmission should be stopped and turn to time slot information synchronization phase to fix the switching rule of the downstream unit. So the performance overhead of our method is a few cycles for transmitting the time slot information to the downstream units and updating the switching rule when the traffic changes.

IV. DYNAMIC TDM VC IMPLEMENTATION

In this section, we introduce the detailed router design supporting the dynamic TDM VC configuration.

A. Data packet and Time slot information packet

We distinguish *data packet* and *slot information packet*. A data packet is split into multiple flits for transmission: *header*, *payload* and *tail* flits. The header flit contains source address, destination address, number of flits of the packet, transmission bandwidth required, header checksum and so on. The tail flit signifies the end of packet. The source address and destination address are the unique symbol for classifying different VCs.

Slot information packet is extracted from a header. As shown in Figure 2, there are two main operations on the time slot information packet: *generator* and *parser*. The time slot information generator at the output link is responsible for generating time slot information packet for downstream units; while the time slot information parser at the input link is responsible for extracting the useful information and generating new switching rules for incoming data.

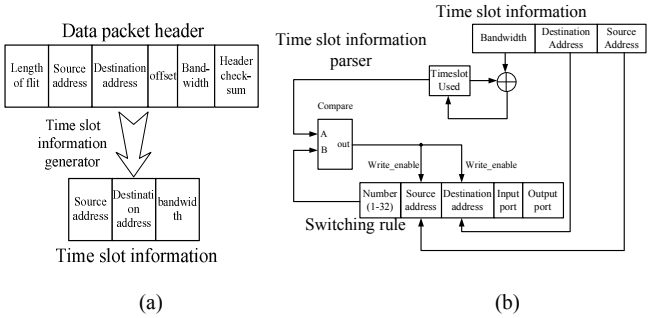

Figure 2. (a) Slot information generator (b) Slot information parser.

If a new data packet is first transferred to a router, the time slot information packet is generated in the time slot generator module according to the new data packet as shown in Figure 2 (a). The transmission rule is changed according to the new VC bandwidth and the Logical Network theory [6]. At the same time, the upstream unit should send time slot information of all the VCs passing through the link to the downstream unit. The downstream unit renews the switching rule using the time slot parser shown in Figure 2(b). The transmission between upstream and downstream unit is established successfully after all the time slot information is passed from the upstream to the downstream unit. The switching rule is a slot allocation table which is dynamically updated.

When a data tail reaches the downstream unit, it is the time that a VC is about to be torn down. The link itself frees the time slot hold by this VC. Freeing a time slot does not need to send time slot information from upstream to downstream units. The router updates the switching rule when a data tail is received.

B. Router Design

In our router, each link has its own *Receiving control module*. The receiving control module contains the switching rule. According to the switching rule, each link knows how to switch received information and which direction the information should head for. When new time slot information packet is received, the *Slot information parser* generates a new switching rule.

Equally, each output link has a transmission rule. The transmission rule is held by the *Transmission control module*. The transmission rule tells the output link the bandwidth of the VC that goes to the output link. The transmission control module switches the new time slot information from the source link to the destination link. The destination link then stops transmitting data flits and begins to generate the new time slot information packet.

Time slot information generator works when new VC is established. There is a queue buffer to store the time slot information. At the same time, the queue buffer pipes the data out to the output link.

Control signal module is the most important part of the router. Each action of the router is controlled by this unit. This module guides the other modules' work: to receive/transmit the data, VC configuration, to parse/generate/transmit time slot information, and etc.

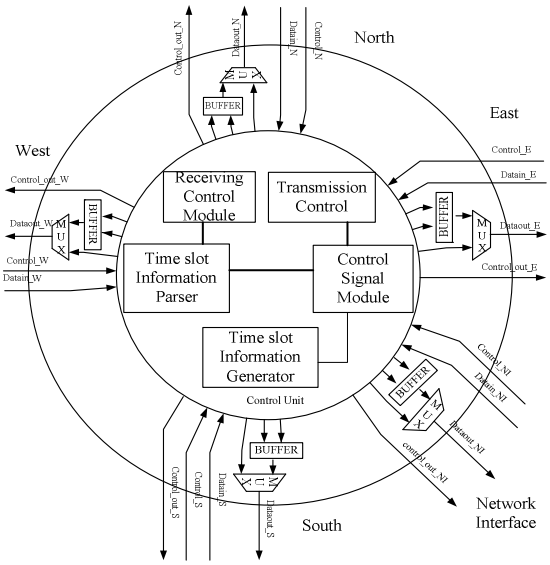

Figure 3. Router design.

V. SIMULATION RESULTS

We have realized the router design in SystemC. We devise simulations to show how slot allocation is dynamically configured in this section.

We assume that a data stream A comes into the router from the North link, requiring 6/32 bandwidth of the East link to transmit the data stream out of the router. When the VC for A is established, a new data stream B comes from the South link of the router, and requires 7/32 bandwidth of the East link to move ahead. These two VCs share the bandwidth resource of the East link. We show the whole process of A and B coming into the router from different links and going out of the router from the same link in a TDM manner.

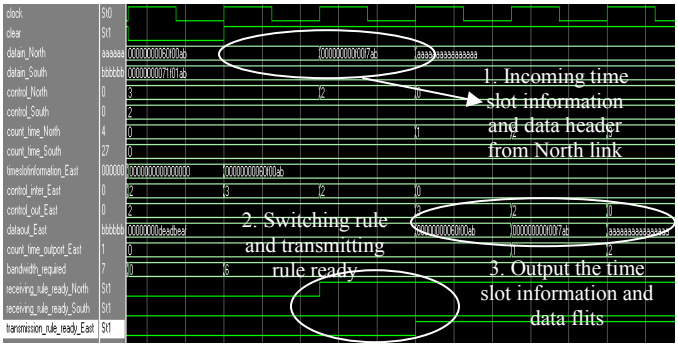

Figure 4. Data stream A comes from North link and goes to East link.

Figure 4 shows the establishing process of the first VC from North link. When the clear signal is set to be high, the system begins working. First, the slot information is coming from the north port. The *control_north* signal recognizes this is a slot information packet. The data header comes right after the slot information. Then, the receiving control module and the slot information parser will update the switching rule at the North link, and then set *receiving_rule_ready_North* to high. At the mean time, the slot information packet is sent to the East link according to the routing table. The internal control signal *control_inter_East* tells that new slot information will come; this new information will be buffered at the East link. The transmission control module will find out the extra bandwidth at the output link, and create the VC with required

bandwidth. When the transmission rule is ready, *transmission_rule _ready_east* is set to high. The time slot information and the data flits are sent out at the East link.

When a new data stream B comes into the router from the South link, the router will also update the switching rule at the South link and the transmission rule at the East link. The transmission control module will allocate adequate bandwidth for both data stream A and B. After the VC for data stream B is established, two time slot information for A and B will be sent out before the data flits transmission; so that the downstream unit can update its switching rule according to the two time slot information packets.

Figure 6 shows the stabilized output of the router. Data packet A (aaaaaaaa), B(bbbbbbbb) occupies 6 and 7 of the total 32 time slots at the East link respectively. The circuit transmits no data (deadbeaf) in the other 19 time slots.

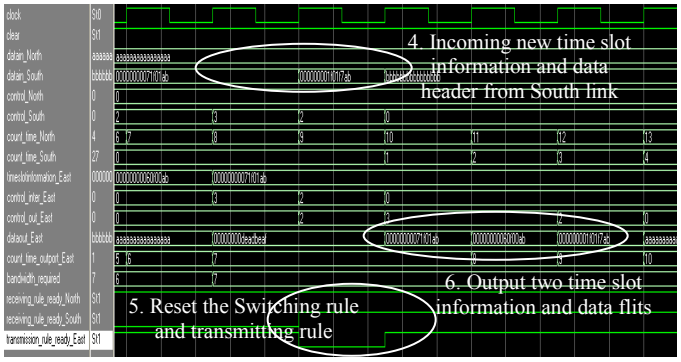

Figure 5. Data stream B comes from South link and goes to East link.

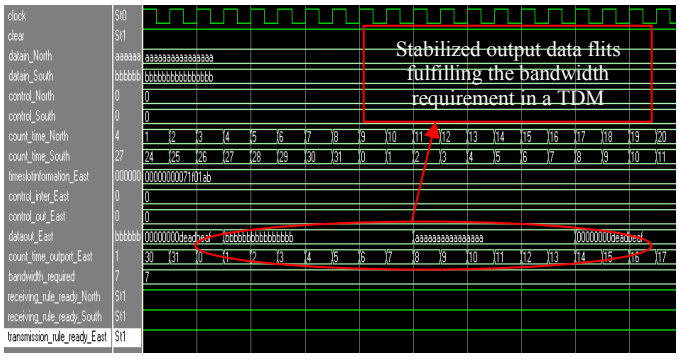

Figure 6. The stabilized data transmission of East link after slot allocation.

VI. CONCLUSION

In this paper, we have proposed a dynamic TDM VC configuration mechanism and its router design. In comparison with a static TDM VC, our VCs are configured on the demand of traffic. It also relaxes the strict synchrony requirement because it performs local synchronization between slot allocation and data transmission. We believe such a dynamic mechanism is desired for dynamic traffic applications and suitable for on chip implementations.

Our future work is to make a synthesizable router design in order to characterize the design's area and power consumption, and to compare with a statically-configurable router.

REFERENCES

[1] K.~Goossens, J.~Dielissen, and A.~Radulescu, The Æthereal network on chip: Concepts, architectures, and implementations, *IEEE Design and Test of Computers*, 22(5), pp. 21--31, Sept-Oct 2005.

[2] M.~Millberg, E.~Nilsson, R.~Thid, and A.~Jantsch, ``Guaranteed bandwidth using looped containers in temporally disjoint networks within the Nostrum network on chip," in *Proceedings of the Design Automation and Test in Europe Conference*, February 2004.

[3] Tobias Bjerregaard and Jens Sparso. A router architecture for connection-oriented service guarantees in the MANGO clockless network-on-chip. In *Proceedings of the conference on Design, Automation and Test in Europe*, pages 1226 -- 1231, March 2005.

[4] E.~Bolotin, I.~Cidon, R.~Ginosar, and A.~Kolodny, QNoC: QoS architecture and design process for network on chip, *The Journal of Systems Architecture*, December 2003.

[5] E.~Nilsson and J.Oberg, Reducing peak power and latency in 2D mesh NoCs using globally pseudochronous locally synchronous clocking," In *Proceedings of the International Conference on Hardware/Software Codesign and System Synthesi,* September 2004.

[6] Zhonghai Lu and and Axel Jantsch. TDM Virtual-Circuit Configuration for Network-on-Chip. IEEE Transactions on Very Large Scale Integration Systems 16(8):1021-1034, August, 2008.

[7] J.~Liu, L.-R. Zheng, and H.~Tenhunen. Interconnect intellectual property for network-on-chip. Journal of System Architectures, Special issue on networks on chip, 50(2):65--79, Feb. 2004.

[8] T.~Marescaux, B.~Bricke, P.~Debacker, V.~N. Nollet, and H.~Corporaal, Dynamic time-slot allocation for {QoS} enabled networks on chip, In *Proc. Of the IEEE 3rd Workshop on Embedded Systems for Real-Time Multimedia*, September 2005, pp. 47--52.

[9] Shekhar Borkar. Thousand Core Chips: A Technology Perspective. Proceedings of the 44th Annual Conference on Design Automation, Pages: 746 – 749, 2007.

[10] Kundu, Partha; Peh, Li-Shiuan, editors, On-Chip Interconnects for Multicores, IEEE Micro, 27 (5), 2007