

LETTER

Pre-Allocation Based Flow Control Scheme for Networks-On-Chip

Shijun LIN^{†a)}, Nonmember, Li SU[†], Member, Haibo SU[†], Depeng JIN[†], and Lieguang ZENG[†], Nonmembers

SUMMARY Based on the traffic predictability characteristic of Networks-on-Chip (NoC), we propose a pre-allocation based flow control scheme to improve the performance of NoC. In this scheme, routes are pre-allocated and the injection rates of all routes are regulated at the traffic sources according to the average available bandwidths in the links. Then, the number of packets in the network is decreased and thus, the congestion probability is reduced and the communication performance is improved. Simulation results show that this scheme greatly increases the throughput and cuts down the average latency with little area and energy overhead, compared with the switch-to-switch flow control scheme.

key words: network-on-chip, system-on-chip, flow control

1. Introduction

As SoC (System-on-Chip) design is entering billion-transistor era, hundreds of IPs are integrated in one chip. Traditional bus-based synchronous communication architecture has shown its limits in bandwidth, clock synchronization and energy consumption. Network-on-Chip (NoC) paradigm is emerging as a new design methodology to overcome the disadvantages of bus-based architecture [1]. NoC design is different from traditional computer network design because of its two characteristics: 1) Area-limited and energy-limited. 2) Traffic predictability. While traditional computer network provides a general platform for all kinds of applications, NoC is developed for one application or at most a small class of applications; therefore, the characteristics of its traffic, i.e. average traffic loads between IPs, are predictable. Thus, in order to design a NoC of higher performance and lower cost, designers should make good use of its characteristics.

In the flow control domain of NoC, previous works are as follows: switch-to-switch flow control schemes, i.e. credit-based and ack/nack schemes, are widely used to avoid buffer overflow and packet drops for BE (Best-Effort) traffic [2]. However, these schemes do not limit the actual traffic injection rate directly at the traffic source. The data are sent when there are buffer spaces available in the downstream router. Then, when the injection rate is too high, congestion occurs and many packets will be delayed and stay in the network. And those delayed packets may delay other packets, which results in the deterioration of network performance. To solve these problems, Ogras [3] et al. propose

a prediction-based flow control scheme. This scheme regulates the packet injection rates at the traffic source based on the prediction of the possible congestions in the network. It decreases the number of packets in the network and thus improves the performance. However, this scheme has two drawbacks: 1) a prediction-based flow controller should be used in every router, which increases the implementation area of the router (18% increase in area); 2) additional energy consumption is needed due to the message exchange of the state of buffers between neighboring routers.

To avoid the drawbacks of the above schemes, we propose a pre-allocation based flow control scheme for BE traffic of NoC based on its traffic predictability. In the scheme, routes are pre-allocated and the injection rates of all routes are regulated at the traffic sources according to the available average bandwidths in the links. Then, the number of packets in the on-chip network is decreased. Therefore, the probability of congestion is greatly reduced and the communication performance is improved. Since only a simple injection controller module is needed and no additional message is exchanged, the area and energy overhead of the proposed scheme will be little.

2. Problem Definition

Given a directed BE communication trace graph $G(V, E)$, where each $v_i \in V$ denotes an IP, and the directed edge $e_{ij} \in E$ denotes a BE communication trace from v_i to v_j , and B_{ij} denotes the average traffic load of e_{ij} . After the topology is selected, IPs are mapped and all routes of GS (Guaranteed Service) traffic are allocated, the state of every link is known. Assume that $w_k \in L$ denotes the bandwidth of link k , $g_k \in F$ denotes the average rate of all GS traffic across link k . Then, the average available bandwidth for BE traffic in link k , Ab_k , equals $(w_k - g_k)$. Our flow control scheme is to map the BE communication traces to the target topology and then determine the injection rate of every BE trace to reduce the probability of congestion. Considering the area-power overhead, we assume that source routing mechanism is used and every BE trace is mapped to a shortest path. And the length of e_{ij} , l_{ij} , is the length of a shortest path between IP_i and IP_j . In the next section, we illustrate our flow control scheme based on the definitions in this section.

3. Pre-Allocation Based Flow Control Scheme

The proposed flow control scheme contains two steps. Be-

Manuscript received September 1, 2008.

[†]The authors are with the Department of electronic and engineering, Tsinghua University, Beijing 100084, China.

a) E-mail: linsj05@mails.tsinghua.edu.cn

DOI: 10.1587/transinf.E92.D.538

fore the illustration, the following definitions are needed.

Definition 1: The load-balance factor of link k , Lbf_k , equals $\frac{\sum_{i,j, e_{ij} \text{ across link } k} IR_{ij}}{Ab_k}$, where IR_{ij} is the injection rate of e_{ij} . $Lbf_k > 1$ means that link k is overloaded. The bigger the load-balance factor of a link is, the more overloaded the link is.

Definition 2: The maximum load-balance factor of a path ($MLbf_{path}$), which equals the maximum value of the load-balance factors of the links through which the path passes.

In the following, we illustrate the pre-allocation based flow control scheme.

Step 1: Assume that the initial injection rate of BE trace e_{ij} equals its average traffic load B_{ij} ; then, pre-allocate a shortest path for every BE trace to make the loads of all links as balanced as possible. According to the definition of the load-balance factor, the task of this step is pre-allocating a shortest path for every BE trace to make the load-balance factors of all links as close as possible. In the following, we describe the detailed operations of this step.

Step 1.1: Put all BE traces in order according to their lengths first and then average traffic loads. Let N be the number of BE traces. Then, $NO.1$ BE trace is the trace with shortest length and highest average traffic load, $NO.2$ BE trace is the trace with shortest length and second highest average traffic load, and $NO.N$ BE trace is the trace with longest length and lowest average traffic load. Initialize n , $n = 1$.

Step 1.2: Pre-allocate a shortest path for $NO.n$ BE trace. If there are several shortest paths for $NO.n$ BE trace, select the shortest path with smallest $MLbf_{path}$.

Step 1.3: If $n = N$, finish step 1 and store the pre-allocation results; if $n < N$, $n = n + 1$ and jump to step 1.2.

Step 2: Limit the injection rates of the BE traces to make sure that no link is overloaded.

Step 2.1: Put all links in order according to their load-balance factors. If the maximum load-balance factor is more than "1", reduce the injection rates of the BE traces which pass through the link with the maximum load-balance factor proportionally to make its load-balance factor equal "1". For example, given that the maximum load-balance factor is $A(A > 1)$, and the injection rates of the BE traces which pass through the link with the maximum load-balance factor are respectively IR_1, IR_2, IR_3, \dots . Then, the injection rates of the BE traces are respectively reduced to $IR_1/A, IR_2/A, IR_3/A, \dots$. Otherwise, if the maximum load-balance factor is no more than "1", jump to step 2.3.

Step 2.2: Re-compute the load-balance factor of all links and jump to step 2.1.

Step 2.3: Store the final injection rates of all BE traces.

4. Implementation of the Proposed Flow Control Scheme

IP and the sending module of Network Interface (NI) which support the proposed scheme are shown in Fig. 1. The

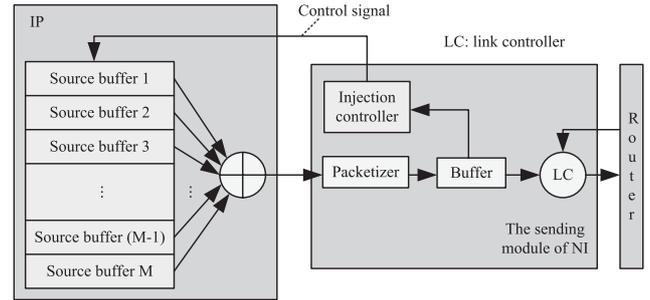


Fig. 1 Implementation of the proposed scheme.

Table 1 Comparison of throughput and latency in case 1.

Case 1 (without GS traffic)		Switch-to-switch scheme	Our scheme	Improve-ment
Traffic load (0.2191 flits /IP /cycle)	Throughput (flits/IP/cycle)	0.2130	0.2185	3%
	Average source latency (cycles)	72	28	61%
	Average network latency (cycles)	48	13	73%
Traffic load (0.6024 flits /IP /cycle)	Throughput (flits/IP/cycle)	0.3659	0.4906	34%
	Average source latency (cycles)	392	163	58%
	Average network latency (cycles)	50	14	72%

receiving module of NI and the router in the proposed scheme are respectively the same with those in the traditional switch-to-switch scheme, thus it is not shown. In the IP, the data with different destination addresses (belong to different BE traces) are stored in different source buffer. And IP sends the data according to the control signal from the corresponding NI. The change in IP is only the store method of the data and the total size of buffers is the same; therefore, the area overhead in IP is neglectable. In the sending module of NI, an injection controller is used to control the injection rates of BE traces. The frequency of the injection clock is set to be equal to the total injection rate of the corresponding IP. A time slot table is generated according to the results of step 2.3 and is stored in the injection controller. We use fixed-length packetizing mechanism and in every time slot, a packet with the corresponding destination address is generated. Then, according to the time slot table, the injection controller informs the corresponding IP when the data are sent and which trace the data belong to by the control signal.

5. Experimental Results and Conclusions

A 4×4 mesh NoC with 16 IPs, 16 NIs and 16 routers is used to study the throughput, average source latency, average network latency and energy of the pre-allocation based flow control scheme and traditional switch-to-switch scheme. Here, throughput is the average number of flits that on-chip network can handle every cycle per IP; average source latency and average network latency of flits are re-

Table 2 Comparison of throughput and latency in case 2.

Case 2 (with GS traffic)		Switch-to-switch scheme	Our scheme	Improve -ment
Traffic load (0.2191 flits /IP /cycle)	Throughput (flits/IP/cycle)	0.2113	0.2171	3%
	Average source latency (cycles)	91	46	49%
	Average network latency (cycles)	51	24	53%
Traffic load (0.6024 flits /IP /cycle)	Throughput (flits/IP/cycle)	0.3591	0.4608	28%
	Average source latency (cycles)	423	241	43%
	Average network latency (cycles)	54	25	54%

Table 3 Comparison of energy and area.

	Switch-to-switch scheme	Our scheme	increase
Total energy of all flits (J)	1.23	1.24	0.8%
Area of a router and a NI (ALUTs)	4071	4149	1.9%

spectively the average number of cycles experienced at the source buffer and in the on-chip network. We assume 8-flit packet, 32-bit flit size, wormhole router with 2 virtual channels, localized self-similar traffic. In the simulation, we consider two cases. In case 1, we assume no GS traffic. In case 2, we consider the effect of GS traffic and use high-priority background traffic with Poisson distribution to model the effect of GS traffic in every link; and we assume

the average rate of GS traffic is about 10% of the link bandwidth. The throughput and latency comparison results of case 1 and case 2 are respectively shown in Table 1 and Table 2. We use the energy model proposed in [2] to estimate the energy and we estimate the area of router and NI based on FPGA EP2S180F1508C5. The energy and area comparison results are shown in Table 3. From Table 1, Table 2 and Table 3, we can see that the proposed scheme could greatly improve the throughput and cut down the average latency with little energy and area overhead.

Acknowledgement

This work is partly supported by National Natural Science Fund (NNSF-90607009), partly supported by the National High Technology Research and Development Program (No.2008AA01Z107) and partly supported by the National Basic Research Program (No.2007CB310701).

References

- [1] A. Ivanov and G. De Micheli, "The network-on-chip paradigm in practice and research," *IEEE Des. Test Comput.*, vol.22, no.5, pp.399–403, 2005.
- [2] J. Hu and R. Marculescu, "Energy- and performance- aware mapping for regular noc architectures," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol.24, no.4, pp.551–562, 2005.
- [3] U.Y. Ogras and R. Marculescu, "Analysis and optimization of prediction-based flow control in networks-on-chip," *ACM Trans. Des. Autom. Electron. Syst.*, vol.13, no.1, Article 11, pp.1–28, 2008.