A Case for Handshake in Nanophotonic Interconnects

Lei Wang*, Jagadish Jayabalan†, Minseon Ahn‡, Haiyin Gu§, Ki Hwan Yum* and Eun Jung Kim*
*Department of Computer Science and Engineering, Texas A&M University, College Station, TX, USA
Email:{wanglei, yum, ejkim}@cse.tamu.edu
†Intel
Email: jagadish.c.jayabalan@intel.com
‡Samsung Electronics
Email: minseon0.ahn@samsung.com
§Bloomberg L.P.
Email: hgu15@bloomberg.net

***Abstract*— Nanophotonics has been proposed to design low latency and high bandwidth NOC for future Chip Multi-Processors (CMPs). Recent nanophotonic NOC designs adopt the token-based arbitration coupled with credit-based flow control, which leads to low bandwidth utilization. In this work, we propose two handshake schemes for nanophotonic interconnects in CMPs, Global Handshake (GHS) and Distributed Handshake (DHS), which get rid of the traditional credit-based flow control, reduce the average token waiting time, and finally improve the network throughput. Furthermore, we enhance the basic handshake schemes with setaside buffer and circulation techniques to overcome the Head-Of-Line (HOL) blocking. Our evaluation shows that the proposed handshake schemes improve network throughput by up to 62% under synthetic workloads. With the extracted trace traffic from real applications, the handshake schemes can reduce the communication delay by up to 59%. The basic handshake schemes add only 0.4% hardware overhead for optical components and negligible power consumption. In addition, the performance of the handshake schemes is independent of on-chip buffer space, which makes them feasible in a large scale nanophotonic interconnect design.**

***Keywords*-Handshake, Networks-On-Chip, Nanophotonic, Architecture.**

I. Introduction

Moore's law has steadily increased on-chip transistor density and integrated dozens of components on a single die. Providing efficient communication in a single die is becoming a critical factor for high performance Chip Multi-Processors (CMPs) [1]. Network-On-Chip (NOC) is a promising architecture that orchestrates chip-wide communications in the many-core era. As the on-chip network size continues to increase, the bandwidth required to support concurrent computations on all cores increases by orders of magnitude. However, many-core systems using electrical interconnects may not be able to meet scalability and high bandwidth while maintaining acceptable performance within power and area budgets [2]. Hence, architects have explored alternative technologies including electrical transmission lines [3], radio frequency (RF) signaling [4], and nanophotonics [5], [6], [7]. While electrical transmission lines and RF suffer from low bandwidth density and relatively large components, nanophotonics provides high bandwidth density, low latency, and distance-independent power consumption, which makes it a promising candidate for future NOC designs.

Optical interconnects have been leveraged to build various on-chip networks. Kirman et al. [5] propose to use optical components to build on-chip buses. and the optical network uses circuit-switching by sending set-up packets in the electrical network. Firefly [6] and Corona [7] propose ring-based networks, which win popularity by getting rid of the overhead of a second electrical network and few or no waveguide crossing even in a large scale network.

Since on-chip buffers are limited resources, flow control becomes a critical factor in the NOC design. In electrical NOC with hop-by-hop transmission, credit-based flow control is preferred since the most recent credit information is instantly available due to the short communication delay between neighbors. In addition, to get the best throughput with credit-based flow control, it is necessary to keep enough number of buffers to cover the credit round-trip delay. The short transmission delay between neighbors helps reduce the buffer requirement. On the other hand, in ring-based optical interconnects where each node is attached to the shared ring, the traffic logically becomes one-hop communication. The one-hop delay between source and destination depends on the ring size, and is normally multiple cycles[1], which makes credit-based flow control become inefficient in the ring-based optical interconnect.

In this work, we propose two handshake schemes for nanophotonic interconnects, Global Handshake (GHS) and Distributed Handshake (DHS). Instead of using the credit-based flow control, the proposed handshake schemes rely on acknowledgments between senders and receivers. A sender

This work was carried out when Jagadish Jayabalan, Minseon Ahn and Haiyin Gu were associated with Texas A&M University.

[1] The round-trip delay of the ring in Corona [7] is 8 cycles.

begins to transmit packets right after winning the channel arbitration without knowing the buffer status at the receiver side. A receiver sends *ACK* or *NACK* messages as a feedback. Packet dropping and retransmission may occur if there is not enough buffer space at the receiver. Furthermore, we propose setaside buffer and circulation techniques to overcome the Head-Of-Line (HOL) blocking in the basic handshake schemes. Packets are moved to the setaside buffer after transmission, yielding the head position to subsequent packets. The circulation technique gets rid of the extra setaside buffer by keeping packets circulating in the network until receivers have enough buffer space.

Our evaluation shows that the proposed handshake schemes improve network throughput by up to 62% under synthetic workloads with the packet dropping and retransmission rates below 1%. With the extracted trace traffic from real applications, the handshake schemes can reduce the communication latency by up to 59%. The basic handshake schemes add only 0.4% hardware overhead for optical components and negligible power consumption. In addition, the performance of the handshake schemes are independent of on-chip buffer space, which makes them feasible in a large scale nanophotonic interconnect design.

The rest of this paper is organized as follows. In Section II, we provide background on the silicon nanophotonic technology and present a motivating case study to highlight the inefficiency of existing optical arbitration and flow control schemes. We present optical handshake schemes in Section III. Section IV describes the architecture of a handshake optical network. In Section V, we describe the evaluation methodology and summarize the simulation results. Then, we briefly summarize the related work in Section VI. Finally, we draw conclusions in Section VII.

II. Motivation

In this section, we first present an overview of optical interconnects, including communication components, interconnect patterns, arbitration and flow control. Then, we discuss the inefficiency of existing optical arbitration and flow control schemes, which is the main issue we attempt to solve in this paper.

A. Optical Communication Components

Optical communication structures consist of a laser source (normally located off-chip), waveguides carrying light, and micro-rings or silicon ring resonators that modulate and detect optical signals. Light from the laser source travels unidirectionally in the waveguides with negligible losses. Multiple wavelengths can use the same waveguide with no interference. With dense-wavelength-division-multiplexing (DWDM), up to 128 wavelengths can be generated and carried by the waveguides [8]. Micro-rings are tuned to a particular wavelength and can be used to modulate or detect light of the particular wavelength when placed next to a waveguide. Meanwhile, rings can switch the light from one waveguide to another. The modulation, detection and diversion are controlled by an electrical signal. Ring resonators are described in [9].

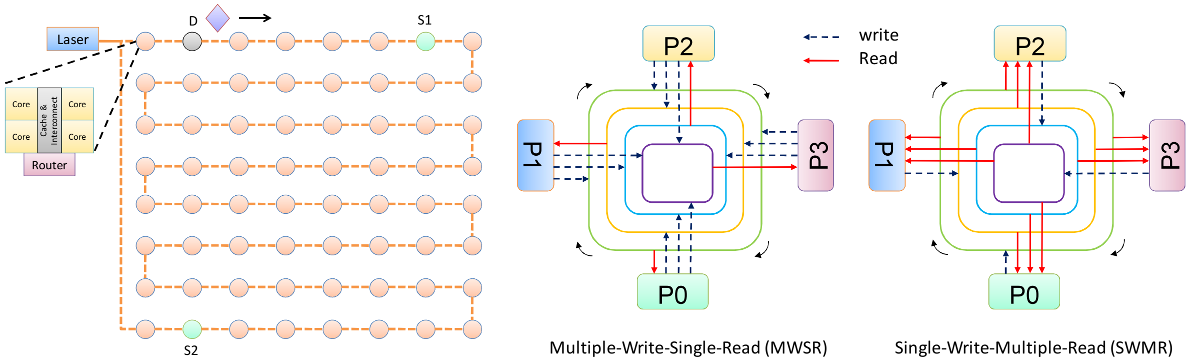
B. Interconnect Patterns

In electrical interconnects, nodes are connected to its neighboring nodes using separate electrical links, such as a 2D Mesh network, while in optical interconnects nodes are normally attached to a single communication media[2] forming a ring-based network as shown in Figure 1 (a). 64 nodes, each of which contains 4 cores, are connected through unidirectional optical rings. The ring-based optical interconnect falls into two categories: Multiple Write Single Read (MWSR) such as Corona [7], or Single Write Multiple Read (SWMR) such as Firefly [6]. Figure 1 (b) shows these two patterns. In MWSR, a node can write to all the channels except one specific channel from which the node can read, while in SWMR a node can write to a specific channel from which any other nodes can read. MWSR needs arbitration in the sender side, since a destination node can only receive one light signal at a time. SWMR benefits from not requiring any arbitration in the sender, but introduces extra communication complexity. Before sending data signals, the sender must notify the receiver of the future communication to activate the receiver's detector, which costs extra bandwidth and needs relatively expensive broadcast waveguides. Although our handshake schemes can be applied to both MWSR and SWMR, we choose MWSR as our interconnect pattern for its simplicity and low cost.

C. Arbitration and Flow Control

With limited on-chip buffer resources, arbitration and flow control become the most critical factors in the NOC design. In nanophotonic interconnects, packets traverse through optical channels in a wave-pipelined manner, which allows a single optical channel to be divided into several segments, and each segment is similar to a single-cycle bus. For example, on a 576 mm^2 chip with 64 nodes and a 5GHz clock, the round-trip time of an optical channel is 8 cycles [7], and thus it can be divided into 8 segments. Considering the specific characteristics of optical channels, the arbitration of a shared optical channel can take two methods: global arbitration and distributed arbitration. Global arbitration is like a bus-based interconnect. In the whole round-trip time, only one sender and one receiver can use the channel. Distributed arbitration considers the wave-pipelined packet transmission. If two packets are not overlapped in the same segment at the same time, they can concurrently traverse in the same optical channel. Prior work [10], [11] adopts token-based arbitration, in which a photonic token represents the right of transmitting packets in a channel. *Token channel* [11] is proposed with global arbitration, while *token slot* [11] and *token stream* [10] are designed using distributed arbitration.

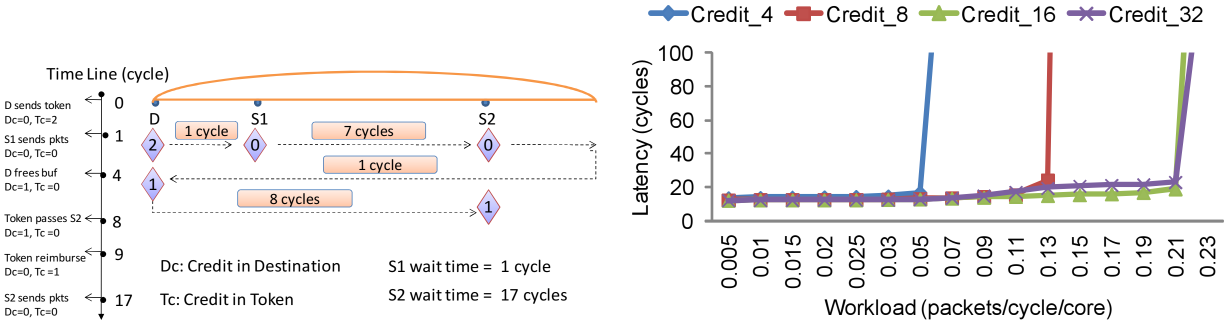
[2] This single communication media is composed of many separate channels, and each channel consists of a couple of waveguides.



(a) Ring-Based Network Architecture.

(b) MWSR and SWMR.

Figure 1. Optical Interconnect Patterns.



(a) Coupled Arbitration and Flow Control.

(b) Performance of Token Slot with Different Number of Credits in Uniform Random Traffic (Detailed simulation configuration can be found in Section V.).

Figure 2. Arbitration and Flow Control in a Token-Ring Network Architecture.

Electrical on-chip interconnects adopt credit-based flow control, in which the upstream router maintains a count of the number of free buffers available downstream. This count is decremented each time a flit[3] is sent out. On the other hand, when a flit leaves the downstream node and frees its associated buffer, a credit is sent back upstream and the corresponding free buffer count is incremented. Inherited from credit-based flow control, the above token-based arbitration schemes integrate the credit information into the arbitration token[4].

D. Case Study

Credit-based flow control benefits from the short transmission delay (normally one cycle) between neighboring nodes. However, in optical interconnects, the one-hop delay between senders and receivers[5] is multiple cycles, which delays the synchronization of the credit information between senders and receivers. Figure 2 (a) shows such a case. We assume the round-trip time of the ring is 8 cycles. Nodes S_1, S_2 and D are connected with a ring, as shown in Figure 1 (a). Nodes S_1 and S_2 want to send packets to Node D. Before sending a packet, S_1 and S_2 need to get a token from Node D, which also carries the credit information of Node D, indicated by T_c in Figure 2 (a). In cycle 0, Node D sends out the token, and its local credit (shown as D_c) becomes zero. In cycle 1, the token arrives at Node S_1 that consumes all the credits. Node S_2 cannot send packets when the token arrives at Node S_2, because there are no credits left in the token. Node S_2 should wait until the token returns to Node D and gets reimbursed. As shown in Figure 2 (a), in cycle 4 Node D has newly freed buffer space (D_c becomes 1). However, the token cannot get this information immediately since it is in the middle of transmission. Finally, it takes 17 cycles before Node S_2 has a chance to send a packet.

Token slot [11] and token stream [10] try to solve the above problem by adopting multiple tokens. Instead of piggybacking all the credits in a single token, token slot and token stream represents one credit per token. The number of tokens depends on the number of credits at destination nodes. Destination nodes stop generating tokens if no more

[3] Flit is the basic flow control unit in NOC. A packet can be composed of multiple flits.

[4] In token channel [11], the number of credits stored in a token can be more than one, while in token slot [11] and token stream [10] a token represents one credit.

[5] Since all the nodes are attached to shared rings, any traffic becomes one-hop communication.

credits are available, making the network performance rely on the size of buffer space as shown in Figure 2 (b). We observe that a certain amount of on-chip buffers should be provided to avoid performance degradation. Note that to get the best throughput with credit-based flow control, it is critical to keep enough number of buffers to cover the credit round-trip delay. This situation becomes more serious in a big network with a long round-trip time of rings. Therefore, credit-based flow control coupled with token-based arbitration is inefficient in the ring-based nanophotonic interconnect design.

III. Optical Handshake

In this section, we propose two handshake schemes, Global Handshake (GHS) and Distributed Handshake (DHS). Instead of using the credit-based flow control, the proposed handshake schemes rely on acknowledgments between senders and receivers. GHS uses global arbitration, while DHS adopts distributed arbitration. Given the high bandwidth density of nanophotonics, the channels are often wide enough so that a large data packet can fit in a single flit[6]. In this work, we assume each packet contains a single flit.

A. Global Handshake

With global arbitration, GHS has a single token relayed among different senders. Since there are multiple writers but only a single reader in MWSR, the reader or the destination node is responsible for sending out the arbitration token. We define the single reader or destination node as the *home* node. When a node detects and removes the token, it has exclusive access to the data channel and starts to send packets in the next cycle. If there are no more packets to be sent, the token is released to the other nodes and finally returns to the home node. It takes multiple cycles for a packet to arrive at the home node. Since senders have no information about the buffer status of the home node, the packet cannot be removed from the sender side after it is sent. When the packet arrives, the home node will check its buffer status. If there is free buffer space, the packet is stored into the buffer and an *ACK* message is sent back to the source node. Otherwise the packet is dropped and a *NACK* message is sent. When the source node receives an *ACK* message, the packet is removed from the input buffer and the following packets are ready for transmission. If a *NACK* message is received, the packet is waiting for retransmission.

Figure 3 shows the operation of GHS. In this example, Node P_0 is set as the home node, and the other nodes try to send packets to P_0. We assume it takes one cycle for the token to traverse between two neighboring nodes[7]. In Cycle 0, Node P_0 sends out the arbitration token, which will circulate in the token channel. Since Node P_1 has no request, the token passes Node P_1 and arrives at Node P_2 in Cycle 1. In Cycle 2, Node P_2 begins to send a data packet. Because Node P_2 has no more packets to send, it releases the token. In Cycle 3, Node P_3 gets the token and sends its data packet, which follows the packet from Node P_2 in a wave-pipelined manner. The token stays in Node P_3, since it has more packets to send. In Cycle 4, the packet from Node P_2 arrives at the home node which has free buffer slots. An *ACK* message is sent to Node P_2 through the handshake waveguide.

GHS gets rid of the credit-based flow control. There is no credit information in the arbitration token. Senders can send a packet without knowing the buffer status at the home node even though there could be no buffer slots available at the home node in the current cycle. If the home node frees a buffer slot one cycle before the packet arrival, the packet can be successfully delivered. With limited buffer space, packet dropping and retransmission may occur. Based on our evaluation, packet dropping and retransmission rate is less than 1% even in high workloads. Decoupling channel arbitration with flow control, GHS shortens the average waiting time. Figure 4 shows the same example as Figure 2 (a) with GHS, where the waiting time for Node S_2 is reduced from 17 cycles to 8 cycles. Using global arbitration, GHS has only one token circulating around the channel. Therefore, after releasing the token, it takes a whole round-trip time for a node to get the token again, even when other nodes have no packets to send. This situation becomes notable in a large network, in which the token round-trip time can be tens of cycles. Distributed Handshake (DHS) uses multiple tokens to overcome this.

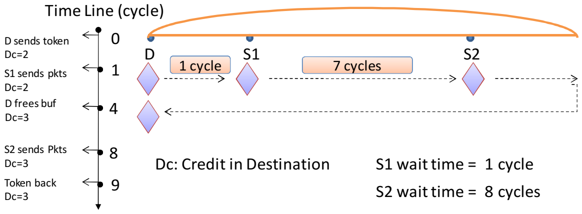


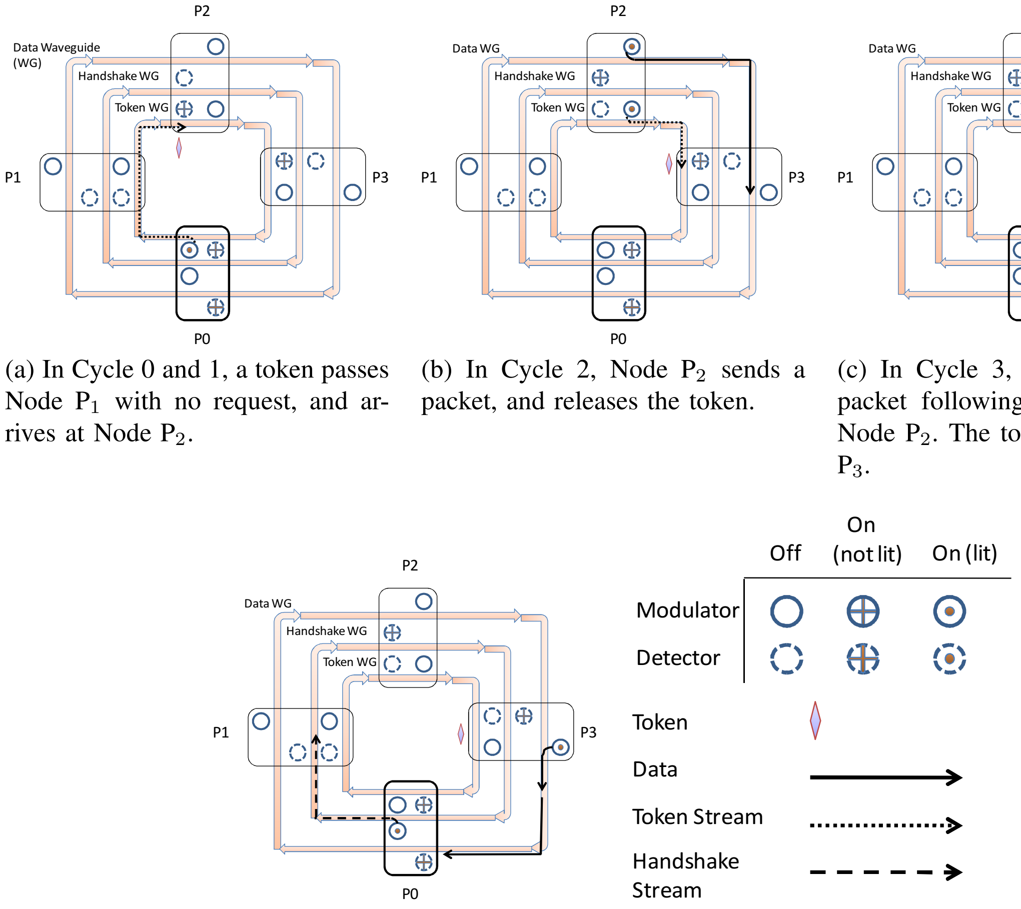
Figure 4. Decoupled Arbitration and Flow Control.

B. Distributed Handshake

DHS considers the wave-pipelined packet transmission in optical links. Home nodes keep generating a token every cycle. Multiple tokens divide the channel into back-to-back segments. In a cycle, only a portion of the network nodes are able to detect the token. If the token is taken by a node, there is no releasing operation for the token and it is removed from the network forever. A sender can only send one packet with a token. Like GHS, packets cannot be removed from the sender side until an *ACK* message is received. Figure 5 shows a walk-through example of DHS. Home Node P_0 keeps generating a token every cycle. In Cycle 0, a token

[6] With a multi-flit packet, we can add the header information into each flit.

[7] In Corona [7], a token can pass eight nodes in one cycle.



(a) In Cycle 0 and 1, a token passes Node P_1 with no request, and arrives at Node P_2.

(b) In Cycle 2, Node P_2 sends a packet, and releases the token.

(c) In Cycle 3, Node P_3 sends a packet following the packet from Node P_2. The token stays in Node P_3.

(d) In Cycle 4, the packet from Node P_2 arrives at the home node with free buffer slots. An *ACK* message is sent to Node P_2.

Figure 3. A Global Handshake Example.

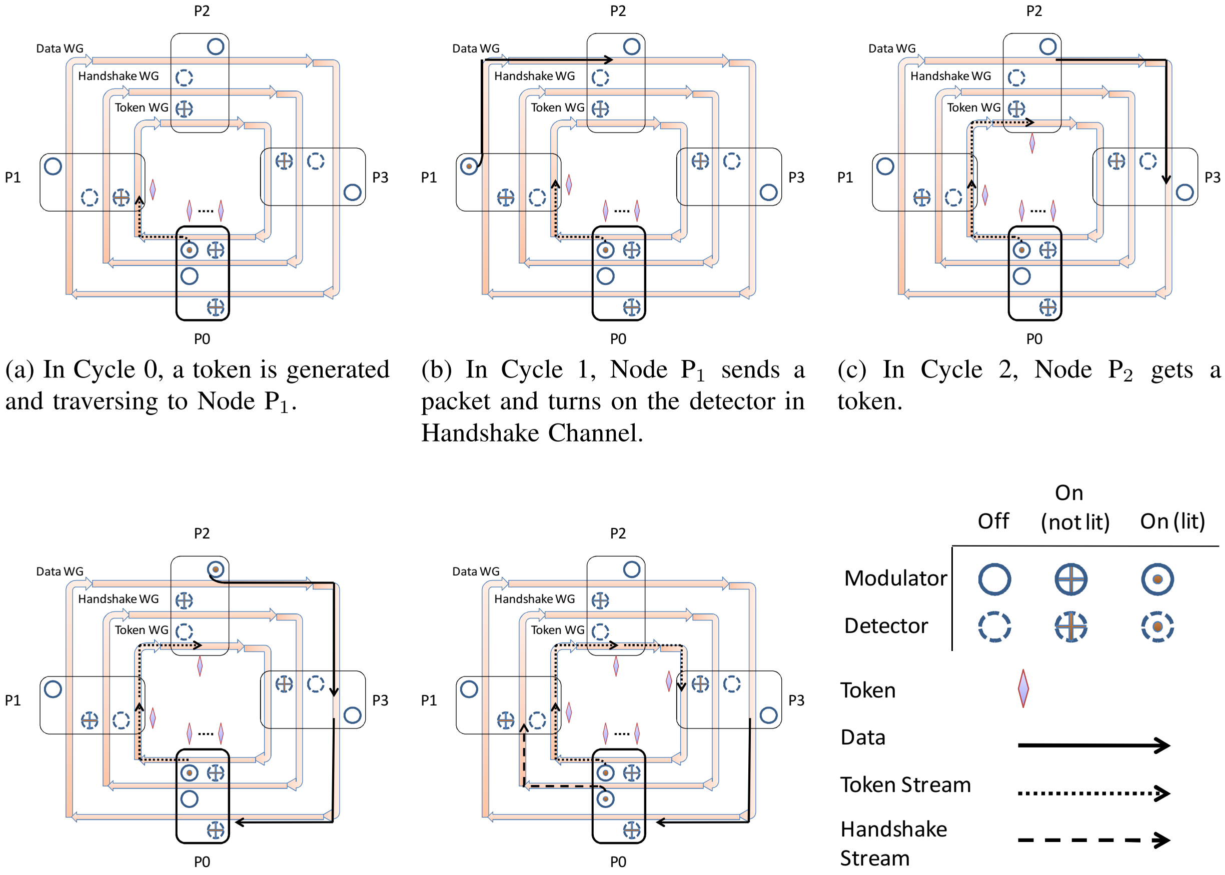
arrives at Node P_1, which removes the token. In Cycle 1, Node P_1 starts to send a packet, and turns on the detector in Handshake Channel. Meanwhile, a new token from the home node is generated and arrives at Node P_1. Since there is no new request from Node P_1, the token will keep traversing to Node P_2. In Cycle 2, the data packet from Node P_1 passes Node P_2, and the token arrives at Node P_2. Node P_2 takes the token, and will send a packet in the next cycle. In Cycle 3, Node P_2 sends a packet, following the previous packet from Node P_1 that arrives at the home node. In Cycle 4, the home node, P_0, sends a handshake message to Node P_1 after checking the buffer status.

GHS and DHS allow senders to send packets without knowing the buffer status of destination nodes, decouple the channel arbitration with flow control, and consequently reduce the credit synchronizing time ideally to zero. However, basic GHS and DHS suffer from the Head-Of-Line (HOL) blocking problem. Before receiving an *ACK* message, senders cannot drop the packet that was sent, which makes the packet stay in the head of the input queue for at least a round-trip time[8]. The pending packet will block the following packets in the same input buffer. To overcome this, we adopt a setaside buffer technique in GHS and DHS. Setaside buffers are small number of buffer slots that are collocated with input queues. Pending packets are removed from the input buffer and wait for the handshake message in the setaside buffer. Therefore, the next packet becomes the head of the queue and is ready for transmission. The size of setaside buffers may affect the network performance, which is discussed in Section V.

C. Distributed Handshake with Circulation

While the setaside buffer technique tackles the HOL problem with additional buffer space, we propose another technique called *circulation* to remove this extra buffer

[8] The time is equal to the traversal time of the data packet from source to destination plus the traversal time of the handshake message back from destination to source.



(a) In Cycle 0, a token is generated and traversing to Node P_1.

(b) In Cycle 1, Node P_1 sends a packet and turns on the detector in Handshake Channel.

(c) In Cycle 2, Node P_2 gets a token.

(d) In Cycle 3, Node P_2 sends a packet following the previous packet that arrives at the home node.

(e) In Cycle 4, a handshake message is sent to Node P_1.

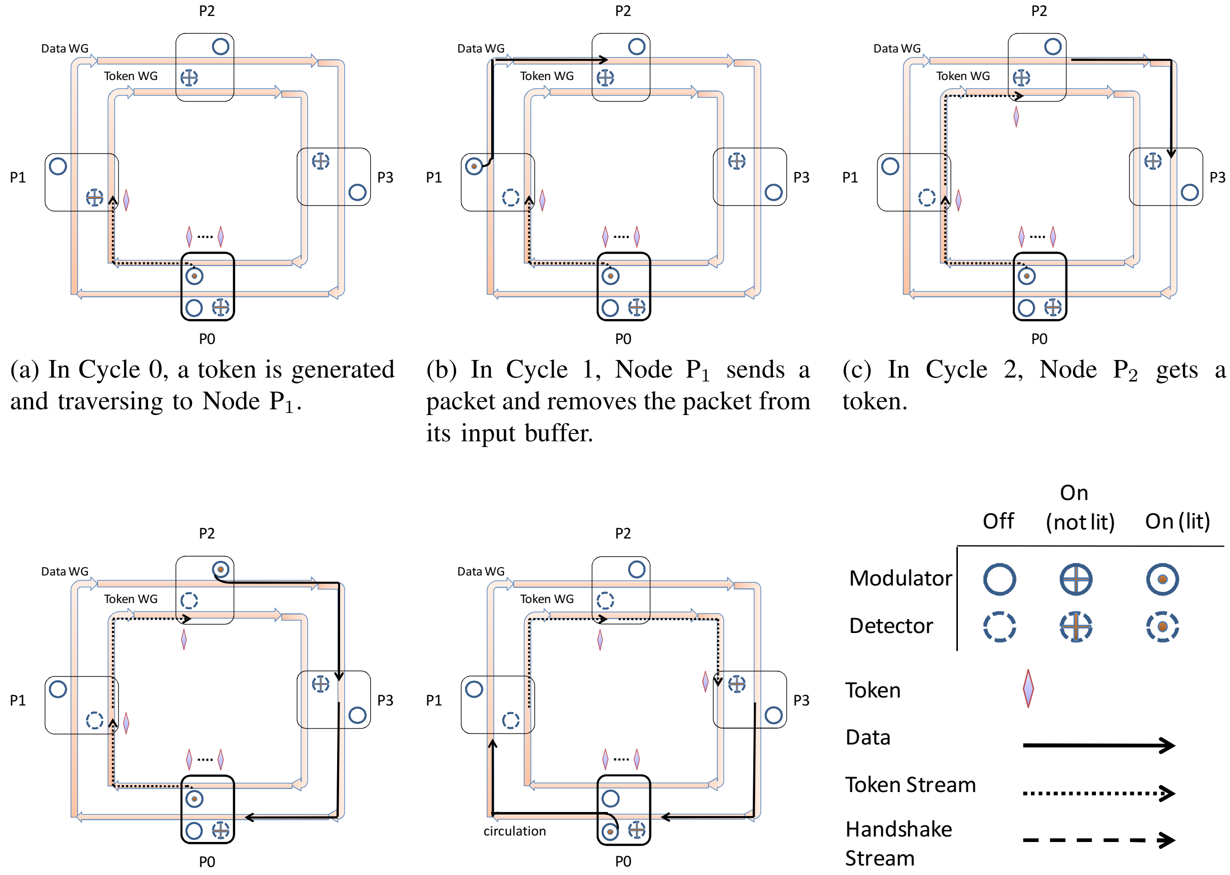
Figure 5. A Distributed Handshake Example.

overhead. The basic idea is that instead of packet dropping, receivers reinject packets into the same data channels if they run out of buffer space. The reinjected packet will circulate in the optical ring until the buffer is available at the receiver. Without packet dropping in the receiver side, circulation enables senders to remove packets from the head of input buffers immediately after sending them out, which gets rid of the HOL blocking problem. Considering no packet retransmission, there is no need for receivers to send acknowledgments, and thus handshake waveguides can be removed.

To integrate the circulation technique, basic DHS should be modified. To avoid channel collision, when a home node needs to reinject packets into the data channel, tokens are not generated in the same cycle. It looks like that the home node virtually consumes a token and gets the permission to use the channel. Figure 6 describes the operation of DHS with the circulation technique. Unlike DHS, the circulation technique cannot be applied to GHS. Note that GHS generates only one channel arbitration token that is relayed among senders. Before the token returns to the home node, the home node cannot grant itself the permission of using the channel, and thus no packets are allowed to be reinjected from the home node.

D. Fairness

One major problem of token-related protocols is fairness. Note that a home node acts as a global controller to generate tokens for every sender. Nodes close to the home node have higher priority over farther downstream nodes in obtaining tokens. Basic GHS and DHS partially solve the fairness issue due to the HOL blocking. Without receiving a handshake message from a home node, senders cannot remove the packets from input buffers, which prevents the following packets from requesting new tokens and potentially yields a newly generated token to downstream nodes. However, with the setaside buffer and circulation techniques, nodes close to home nodes can starve the farther downstream nodes. A similar problem has been addressed in [11], which proposes Fair Token Channel and Fair Slot with well served nodes



(a) In Cycle 0, a token is generated and traversing to Node P_1.

(b) In Cycle 1, Node P_1 sends a packet and removes the packet from its input buffer.

(c) In Cycle 2, Node P_2 gets a token.

(d) In Cycle 3, Node P_2 sends a packet following the previous packet that arrives at the home node, which has no free buffer slots at that time.

(e) In Cycle 4, the home node reinjects the packet into the data channel. Meanwhile, the home node stops generating a token for that cycle.

Figure 6. Distributed Handshake with Circulation.

sitting on their hands for a while and yielding the chance to other nodes. In this work, we adopt the same methods to provide fairness in GHS and DHS.

IV. Optical Handshake Architecture

In this section, we present the network architecture with handshake schemes.

A. Network Architecture

Figure 7 shows the optical network architecture with the handshake schemes. Routers are attached to global optical rings that are composed of different channels, including data channels, token channels and handshake channels. A channel consists of multiple waveguides carrying wavelengths. To support handshake schemes, extra components are added to the conventional virtual channel (VC) router, which are labeled as *Output* and *Input* modules. In the *Output* module, an output queue is used to buffer the packets before Electronic/Optical (E/O) conversion. To avoid the HOL blocking, setaside buffers can be added in parallel with output buffers. Each setaside buffer is one flit long and connected to an output mux. A handshake receiver processes *ACK* or *NACK* messages, and selects a flit to enter E/O conversion. In the *Input* module, the detector checks the status of the global optical ring. When a flit arrives, the flit will be stored into the router input buffer after Optical/Electronic (O/E) conversion. In basic GHS and DHS, if there are no free slots in the input buffer, flits will be dropped. With the circulation technique, router buffer status is recorded in the circulation controller, which decides the packet path, entering input buffers or reinjection.

B. Router Pipeline

An electrical NOC router processes packets with four pipeline stages, which are routing computation (RC), VC allocation (VA), switch allocation (SA), and switch traversal (ST). In optical on-chip networks, routers are attached to the global ring, thus making any two routers neighbors, which

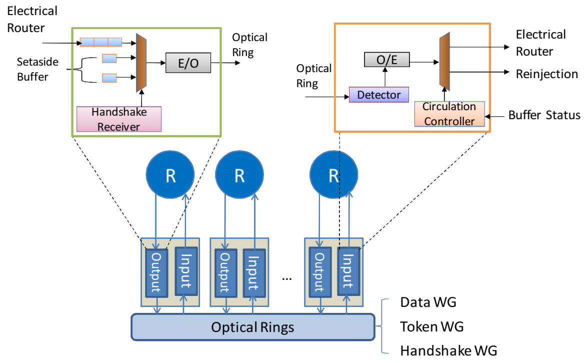


Figure 7. The Optical Network Architecture with the Handshake Schemes.

increases the overhead of maintaining the VC status for every neighboring router. Note that optical interconnects can provide a wide link width, which is feasible for a single-flit packet design. Flit interleaving in a network cannot occur with only single-flit packets. Therefore, the VA stage can be removed from the router pipeline, simplifying the electrical router logic. Since a router is shared by multiple cores and packets from different cores may have the same destination, the SA stage is still required. In this work, we adopt a two-stage electrical router, with RC and SA in one stage and ST in the other.

C. *Hardware Overhead*

The handshake schemes introduce handshake messages (*ACK* and *NACK*) into normal optical communication, which incurs extra hardware overhead. We analyze the hardware overhead in a network with 256 cores connected to 64 nodes. We advocate using a single bit for a handshake message. Note that in a segment of the channel only one node can get the arbitration token every cycle, and the round-trip time for an optical ring is fixed. After sending a packet, the sender will receive a handshake message in a fixed amount of time. For example, if the round-trip time for the optical ring is 8 cycles, then a sender will receive the handshake message in 9 cycles. A sender only needs to turn on its handshake detector 9 cycles after sending a packet, while at other times it keeps the detector off and passes the handshake messages for other senders. That's why using a single bit, which indicates whether it is an *ACK* or a *NACK*, for the handshake message is feasible. If we use one wavelength, modulated as 1 bit, for the handshake message of a node, 64 wavelengths are required in a 64-node network. Note that an optical waveguide can carry 64 wavelengths. Thus, only one waveguide is added to support the handshake schemes in a 64-node network. Since each wavelength requires 64 micro-rings to function as modulators or detectors, this extra waveguide needs total 4K (0.4%) micro-rings. In basic GHS and DHS, a home node only reads packets from its dedicated data channels[9], while with the circulation technique the home node needs to reinject packets, adding extra 16K (1.5%) micro-rings in the whole network. Table I lists the budget of optical components for each handshake scheme.

Table I
COMPONENT BUDGETS FOR THE HANDSHAKE SCHEMES IN A 64-NODE NETWORK.

Schemes	**Data WG**	**Token WG**	**Handshake WG**	**Micro-rings**
Token Slot	256	1	0	1024K
GHS	256	1	1	1028K
DHS	256	1	1	1028K
DHS-cir	256	1	0	1040K

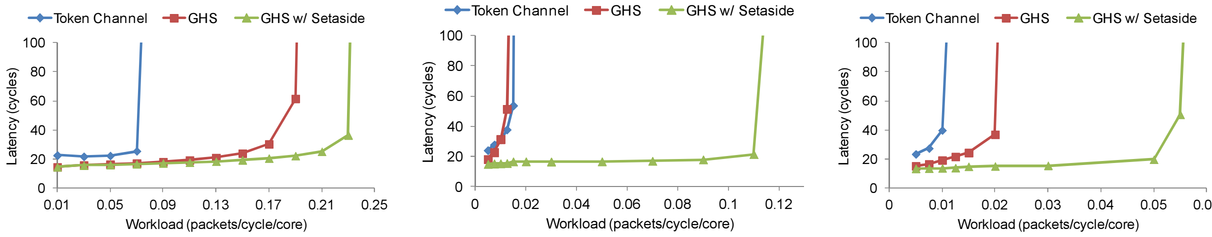
V. EXPERIMENTAL EVALUATION

In this section, we first describe our evaluation methodology. Then, the performance of the proposed handshake schemes is analyzed. We explore the schemes' sensitivity to a variety of network design parameters. Based on the power model in [12], [13], we estimate the power consumption in the handshake schemes.

A. *Methodology*

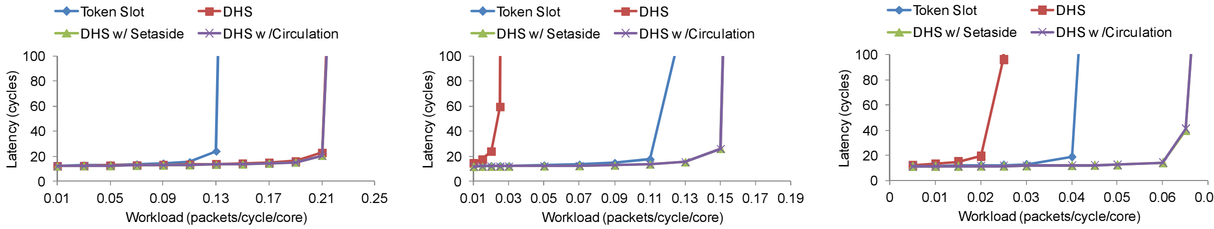
Our evaluation methodology contains two parts. First, we use Simics [14], a full system simulator configured as a SunFire multiprocessor system with UltraSPARCIII+ processors running Solaris 9 operating system, to extract trace information from real applications. We develop a customized timing-model interface modeling out-of-order cores with 4 MSHRs per each processing core to implement a self-throttling CMP network [15]. The CMP system contains 128 out-of-order processing cores and 128 L2 cache banks in a single chip, connected as 64 nodes with 4-way concentration, modeling static non-uniform cache architecture (S-NUCA) [16]. Next, we evaluate performance and power consumption using a cycle-accurate on-chip network simulator that models a 2-stage pipelined router architecture. The total latency of E/O or O/E conversion is around 75ps [17] and is modeled as part of the nanophotonic link traversal time. Assuming a die size of 400mm^2 with a 5GHz clock, the nanophotonic link traversal time amounts to be 1 to 8 cycles based on the distance between the sender and the receiver. The workloads for our evaluation consist of synthetic workloads and traces from real applications. Three different synthetic traffic patterns, Uniform Random (UR), Bit Complement (BC) and Tornado (TOR), are used. The real applications considered in this work are fma3d, equake, and mgrid from SPEComp2001 [18]; blackscholes, freqmine, streamcluster, and swaptions from PARSEC [19]; FFT, LU, and radix

[9] In MWSR, the home node is the single reader while other nodes are all writers.



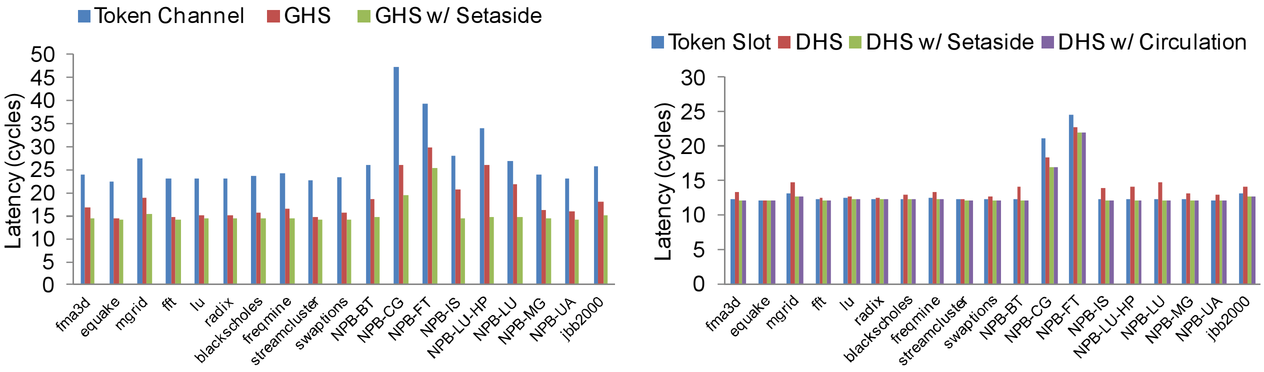
(a) UR. (b) BC. (c) TOR.

Figure 8. Performance Evaluation of Global Handshake.



(a) UR. (b) BC. (c) TOR.

Figure 9. Performance Evaluation of Distributed Handshake.



(a) Global Handshake. (b) Distributed Handshake.

Figure 10. Performance Evaluation with Real Applications.

from SPLASH-2 [20]; NAS parallel benchmarks [21] and SPECjbb2000 [22]. Considering the high bandwidth of optical channels, we use a one-flit packet in our experiments.

B. Performance

Given that GHS and token channel use global arbitration while DHS and token slot adopt distributed arbitration, we separate the performance evaluation into two groups. GHS related schemes are compared with token channel, while DHS related schemes are compared with token slot.

Synthetic Workloads: We first evaluate average packet latency and saturation bandwidth with synthetic workloads. Figures 8 (a), (b) and (c) show the results of the schemes using global arbitration, in which only one arbitration token is circulating for each destination. The total amount of credits or buffer slots provided by each destination is 8. The trend from the three traffic patterns is consistent. Because token channel [11] suffers from the long token waiting time, especially after senders consume all the credits stored in the token, GHS produces better performance than token channel. Since the number of tokens depends on the number of credits at the destination, limited buffer space restrains the token slot performance. Unlike token slot, there is no credit-based flow control in the handshake schemes where tokens are generated every cycle maximizing the transmission opportunity for senders, which is more efficient than token slot especially when destinations get free buffer space while packets are already in the middle of traversal. Packet dropping and retransmission are the potential overhead introduced by handshake schemes. Simulation results show that even with high injection rates, the packet dropping and retransmission rates are below 1%, which also implies that the packet circulation occurs rarely.

The HOL blocking problem affects the performance of basic GHS and DHS. Although there are free tokens, the following flits cannot seize a token because the flit in the head of the queue is waiting for the acknowledgment. This situation becomes more obvious in the peer-to-peer communication patterns such as BC. In Figure 9 (b), we can see that token slot outperforms basic DHS. With the setaside buffer technique, flits can wait for the acknowledgments in the setaside buffer, yielding the head position to following flits, which brings significant throughput improvement. The setaside buffer and circulation techniques have almost the same effect on relieving the HOL blocking. Compared with the setaside buffer technique, the circulation does not require additional buffer space, and is a more promising design.

Real Applications: Figure 10 shows the performance results with real applications. Since the packet injection rate of each node in these real applications is very low, the performance improvement is not comparable to synthetic workloads. It is clear that the handshake schemes outperform previous schemes, especially in NAS parallel benchmarks. Compared with token channel, GHS reduces communication latency by an average of 42%, while DHS achieves an average of 4% latency reduction over token slot. Suffering from the HOL blocking problem, basic GHS and DHS cannot perform as well as GHS and DHS with the setaside buffer and circulation techniques. However, in most of the selected benchmarks, basic GHS and DHS outperform token channel and token slot, respectively.

To study the effect of the handshake schemes on the system performance, we evaluate the IPC. In this experiment, we select the handshake schemes with the setaside buffer technique to compare with token channel and token slot separately. GHS improves the IPC by an average of 15% compared with token channel, while DHS gets 1.3% IPC improvement over token slot.

Sensitivity Study: We present variations that provide insight into the performance of the handshake schemes in different environments. We select UR as our traffic pattern and set the injection rate as 0.11. First, we evaluate the handshake schemes with various numbers of credits. Because in the handshake schemes, a token is used only for channel arbitration and no credit information is piggybacked, the performance of the handshake schemes is nearly independent of the number of credits, as shown in Figures 11 (a) to (e), which makes the handshake schemes feasible in a large network. Next, we analyze the performance of the handshake schemes with different sizes of the setaside buffer. Figure 11 (f) shows that the handshake schemes can produce comparable performance with only a small size setaside buffer.

C. Power

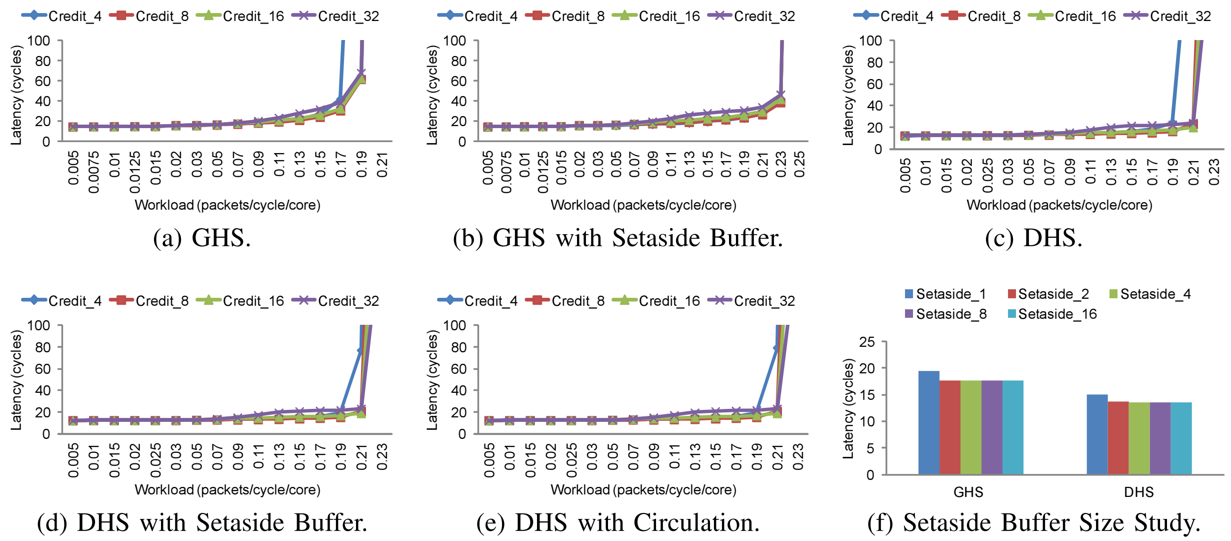
The power dissipated in nanophotonic on-chip networks is composed of electrical router power, modulation/demodulation power, laser power and ring tuning power. Laser power and ring tuning power are also known as static power which dominates the overall power consumption. Modulation/demodulation power is determined by the number of E/O and O/E conversions. We use 158 fJ/b [12] as the energy cost for each signal conversion. To calculate the laser power, we consider the E/O conversion losses as well as transmission losses in the waveguide. Waveguide loss is length-dependent. A non-linearity limit of 30 mW at 1 dB loss is assumed for waveguides. We assume 10 μW for the sensitivity of photodetectors [10]. Additionally, all rings in the system must be thermally tuned to maintain their resonance under on-die temperature variations. We assume 1 μW tuning power per ring per K, and a temperature range of 20K [13]. We use Orion 2.0 [23] power model to estimate the power consumption of an electrical router.

Figure 12 (a) shows the power comparison among different schemes. As expected, laser power and ring heating power are dominant in all the schemes. Because the schemes with global arbitration, such as token channel and GHS, have only one shared token circulated in the network, which incurs more optical loss, they consume more laser power than the schemes with distributed arbitration such as token slot and DHS. Given that the token in GHS does not carry credit information, GHS has less laser power consumption than token channel. Among all the schemes, token slot has the lowest power consumption because the handshake schemes add additional handshake waveguides. However, the power overhead introduced by additional handshake waveguides is negligible as shown in Figure 12 (a). Figure 12 (b) indicates the average energy consumption for delivering a packet. With the passive writing nature of nanophotonics, where the modulation is done by imprinting messages onto a laser beam rather than driving a whole channel, the circulation technique has nearly no energy overhead for delivering a packet.

VI. Related Work

Handshake has been widely used in the Internet. TCP/IP protocols adopt three-way handshake for reliable data transfer (RDT) [24]. On-chip interconnect, which is considered to be reliable, also hires acknowledgment-based transmissions. In a circuit switching network, to set up a transmission circuit from source to destination, a routing probe is injected and traversing to the destination, which will send back an acknowledgment to notify the successful circuit set-up. Previous studies [25], [26] introduce an ACK network to reduce the buffer requirements of the on-chip network.

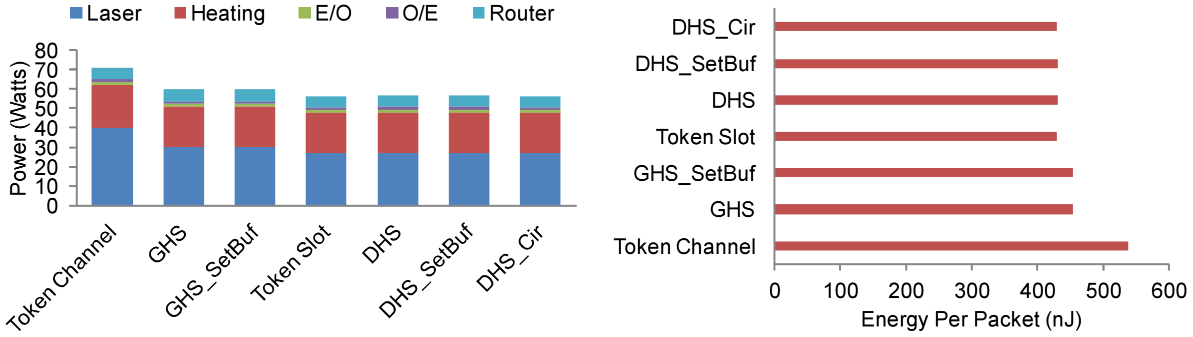
The emerging nanophotonic technology enables on-chip optical interconnect. Different on-chip network architectures have been proposed to exploit silicon nanophotonics. Kirman et al. [5] propose to use optical components to build on-chip buses. Shacham et al. [27] propose a circuit-switching photonic interconnect for data packets in parallel with an electric network. Nanophotonic switching is explored in the Phastlane [28]. The Corona [7] architecture implements a



(a) GHS. (b) GHS with Setaside Buffer. (c) DHS.

(d) DHS with Setaside Buffer. (e) DHS with Circulation. (f) Setaside Buffer Size Study.

Figure 11. Sensitivity Studies with Uniform Random Traffic.



(a) Total Power Breakdown. (b) Energy Consumption per Packet.

Figure 12. Power and Energy Analysis.

monolithic crossbar topology to support on-chip communication. Firefly [6] uses partitioned nanophotonic crossbars to connect clusters of electrically connected mesh networks. Joshi et al. [13] build a nanophotonics clos network, which provides uniform latency and throughput with low power.

Ha et al. [29] and Kodi et al. [30] advocate token-based protocols to arbitrate for optical off-chip interconnects. An optical arbiter can be found in [31]. Vantrease et al. [11] propose token channel and token slot for optical on-chip interconnects, which piggyback flow control information on the arbitration tokens. FlexiShare [10] reduces the number of channels across the network and proposes single-pass and two-pass token stream arbitration.

VII. Conclusions

Optical interconnects have been leveraged to build various on-chip networks. In this paper, we propose handshake schemes for nanophotonic interconnects, Global Handshake (GHS) and Distributed Handshake (DHS). By getting rid of the credit-based flow control, GHS and DHS reduce the average token waiting time and improve the network throughput. To overcome the HOL blocking problem existing in the basic handshake schemes, we propose the setaside buffer and circulation techniques. Our evaluation shows that the proposed handshake schemes improve network throughput by up to 62% under synthetic workloads. For real applications, the handshake schemes can reduce the communication latency by up to 59%. The basic handshake schemes add only 0.4% hardware overhead for optical components and negligible power consumption. In addition, the performance of the handshake schemes are independent of on-chip buffer space, which makes them feasible in a large scale nanophotonic interconnect design.

References

[1] J. D. Owens, W. J. Dally, R. Ho, D. N. Jayasimha, S. W. Keckler, and L.-S. Peh, “Research Challenges for On-Chip Interconnection Networks,” *IEEE Micro*, vol. 27, no. 5, pp. 96–108, 2007.

[2] R. Kumar, V. V. Zyuban, and D. M. Tullsen, “Interconnections in Multi-Core Architectures: Understanding

Mechanisms, Overheads and Scaling," in *ISCA*, 2005, pp. 408–419.

[3] A. Jose and K. Shepard, "Distributed Loss-Compensation Techniques for Energy-Efficient Low-Latency On-Chip Communication," *Solid-State Circuits*, vol. 42.

[4] M. F. Chang, J. Cong, A. Kaplan, M. Naik, G. Reinman, E. Socher, and S.-W. Tam, "CMP Network-On-Chip Overlaid with Multi-Band RF-Interconnect," in *HPCA*, 2008, pp. 191–202.

[5] N. Kirman, M. Kirman, R. K. Dokania, J. F. Martínez, A. B. Apsel, M. A. Watkins, and D. H. Albonesi, "Leveraging Optical Technology in Future Bus-based Chip Multiprocessors," in *MICRO*, 2006, pp. 492–503.

[6] Y. Pan, P. Kumar, J. Kim, G. Memik, Y. Zhang, and A. N. Choudhary, "Firefly: Illuminating Future Network-On-Chip with Nanophotonics," in *ISCA*, 2009, pp. 429–440.

[7] D. Vantrease, R. Schreiber, M. Monchiero, M. McLaren, N. P. Jouppi, M. Fiorentino, A. Davis, N. L. Binkert, R. G. Beausoleil, and J. H. Ahn, "Corona: System Implications of Emerging Nanophotonic Technology," in *ISCA*, 2008, pp. 153–164.

[8] X. Zhang and A. Louri, "A Multilayer Nanophotonic Interconnection Network for On-Chip Many-Core Communications," in *DAC*, 2010, pp. 156–161.

[9] L. Zhang, M. Song, T. Wu, L. Zou, R. G. Beausoleil, and A. E. Willner, "Embedded Ring Resonators for Microphotonic Applications," *Optics Letters*, vol. 33.

[10] Y. Pan, J. Kim, and G. Memik, "FlexiShare: Channel Sharing for an Energy-Efficient Nanophotonic Crossbar," in *HPCA*, 2010, pp. 1–12.

[11] D. Vantrease, N. L. Binkert, R. Schreiber, and M. H. Lipasti, "Light Speed Arbitration and Flow Control for Nanophotonic Interconnects," in *MICRO*, 2009, pp. 304–315.

[12] C. Batten, A. Joshi, J. Orcutt, A. Khilo, B. Moss, C. Holzwarth, M. Popovic, H. Li, H. I. Smith, J. L. Hoyt, F. X. Kärtner, R. J. Ram, V. Stojanovic, and K. Asanovic, "Building Manycore Processor-to-DRAM Networks with Monolithic Silicon Photonics," in *Hot Interconnects*, 2008, pp. 21–30.

[13] A. Joshi, C. Batten, Y.-J. Kwon, S. Beamer, I. Shamim, K. Asanovic, and V. Stojanovic, "Silicon-Photonic Clos Networks for Global On-Chip Communication," in *NOCS*, 2009, pp. 124–133.

[14] P. S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, G. Hållberg, J. Högberg, F. Larsson, A. Moestedt, and B. Werner, "Simics: A Full System Simulation Platform," *IEEE Computer*, vol. 35, no. 2, pp. 50–58, 2002.

[15] D. Kroft, "Lockup-Free Instruction Fetch/Prefetch Cache Organization," in *ISCA*, 1981, pp. 81–87.

[16] C. Kim, D. Burger, and S. W. Keckler, "An Adaptive, Non-Uniform Cache Structure for Wire-Delay Dominated On-Chip Caches," in *ASPLOS*, 2002, pp. 211–222.

[17] P. Kapur and K. C. Saraswat, "Comparisons between Electrical and Optical Interconnects for On-Chip Signaling," in *International Interconnect Technology Conference*, 2002, pp. 89–91.

[18] "Specomp 2001 benchmark suite," http://www.spec.org/omp/.

[19] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC Benchmark Suite: Characterization and Architectural Implications," in *PACT*, 2008, pp. 72–81.

[20] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The SPLASH-2 Programs: Characterization and Methodological Considerations," in *ISCA*, 1995, pp. 24–36.

[21] "Nas parallel benchmarks," http://www.nas.nasa.giv/Resources/Software/npb.html.

[22] "Specjbb 2000 benchmark," http://www.spec.org/jbb2000/.

[23] A. B. Kahng, B. Li, L.-S. Peh, and K. Samadi, "ORION 2.0: A Fast and Accurate NoC Power and Area Model for Early-Stage Design Space Exploration," in *DATE*, 2009, pp. 423–428.

[24] V. C. R. Kahn, "A Protocol for Packet Network Intercommunication," *Communications, IEEE Transactions on*, vol. 22, no. 5, pp. 637 – 648, 1974.

[25] M. Hayenga, N. D. E. Jerger, and M. H. Lipasti, "SCARAB: A Single Cycle Adaptive Routing and Bufferless Network," in *MICRO*, 2009, pp. 244–254.

[26] C. G. Requena, M. E. Gómez, P. J. L. Rodríguez, and J. Duato, "An Efficient Switching Technique for NoCs with Reduced Buffer Requirements," in *ICPADS*, 2008, pp. 713–720.

[27] A. Shacham, K. Bergman, and L. P. Carloni, "On the Design of a Photonic Network-On-Chip," in *NOCS*, 2007, pp. 53–64.

[28] M. J. Cianchetti, J. C. Kerekes, and D. H. Albonesi, "Phastlane: A Rapid Transit Optical Routing Network," in *ISCA*, 2009, pp. 441–450.

[29] J.-H. Ha and T. M. Pinkston, "A New Token-Based Channel Access Protocol for Wavelength Division Multiplexed Multiprocessor Interconnects," *J. Parallel Distrib. Comput.*, vol. 60, no. 2, 2000.

[30] A. K. Kodi and A. Louri, "A Scalable Architecture for Distributed Shared Memory Multiprocessors Using Optical Interconnects," in *IPDPS*, 2004.

[31] C. Qiao and R. G. Melhem, "Time-division optical communications in multiprocessor arrays," in *SC*, 1991.