2014 IEEE 28th International Parallel & Distributed Processing Symposium

Energy-Efficient Time-Division Multiplexed Hybrid-Switched NoC for Heterogeneous Multicore Systems

Jieming Yin⋆, Pingqiang Zhou†, Sachin S. Sapatnekar⋆ and Antonia Zhai⋆

⋆University of Minnesota, Twin Cities
Minneapolis, Minnesota 55455, USA
{jyin, zhai}@cs.umn.edu, sachin@umn.edu

†ShanghaiTech University
Shanghai 200031, China
zhoupq@shanghaitech.edu.cn

***Abstract*—NoCs are an integral part of modern multicore processors, they must continuously support high-throughput low-latency on-chip data communication under a stringent energy budget when system size scales up. Heterogeneous multicore systems further push the limit of NoC design by integrating cores with diverse performance requirements onto the same die. Traditional packet-switched NoCs, which have the flexibility of connecting diverse computation and storage devices, are facing great challenges to meet the performance requirements within the energy budget due to latency and energy consumption associated with buffering and routing at each router.**

In this paper, we take advantage of the diversity in performance requirements of on-chip heterogeneous computing devices by designing, implementing, and evaluating a hybrid-switched network that allows the packet-switched and circuit-switched messages to share the same communication fabric by partitioning the network through time-division multiplexing (TDM). In the proposed hybrid-switched network, circuit-switched paths are established along frequently communicating nodes. Our experiments show that utilizing these paths can improve system performance by reducing communication latency and alleviating network congestion. Furthermore, better energy efficiency is achieved by reducing buffering in routers and in turn enabling aggressive power gating.

***Keywords*-interconnection network, energy-efficiency**

I. Introduction

Concomitant with diminishing performance improvement of complex uniprocessors, integrating multiple energy-efficient accelerator cores with a traditional superscalar cores onto the same die emerged as way to achieve the desired performance goal within a stringent power budget. Recently, heterogeneous multicore systems, such as Intel's Sandy Bridge [1], and AMD's Fusion [2], have become prevalent and provided feasibility for balancing single threaded performance and high throughput requirements. In heterogeneous systems, on-chip communication exhibits variant traffic patterns: CPU-like superscalar cores generate moderate coherence and core-to-core data sharing traffic, while GPU-like data parallel cores, on the contrary, have a high throughput requirement and generate throughput-intensive streaming traffic. Consequently, the Network-on-Chips (NoCs) must be designed to handle both types of traffic efficiently in terms of performance and energy.

A packet-switched network offers the flexibility and scalability for connecting a large number of diverse devices; however, the buffering and routing necessary for each message introduces considerable delay and energy overhead. Prior works showed significant energy consumption associated with buffering at the routers [3], [4]. On the other hand, a circuit-switched network is cost-efficient for throughput-intensive traffic; however, infrequent use of the circuit-switched paths leads to under-utilization of on-chip resources. A hybrid-switched NoC that supports both packet and circuit switching can potentially facilitate the design of NoCs that are both energy-efficient and flexible. In hybrid-switched NoCs, circuit-switched paths act as dedicated express channels between nodes, and packets taking advantage of these channels can avoid performance and energy overhead associated with routing. Thus, such networks can be potentially suited for heterogeneous multicore systems that consolidate diverse computation and caching components on a single die. How can these two switching mechanism be consolidated onto a single network?

Jerger et al. [5] propose a space-division multiplexing (SDM) based hybrid-switched NoC. In this network, links are physically partitioned into planes. Each individual plane is allocated to a given circuit-switched connection. The SDM-based hybrid-switched NoC demonstrates performance improvement for coherence-based traffic where most network messages are short and the number of circuit-switched paths is essentially small. However, an SDM network serializes packets as they are forced to use a single plane even though the other planes are idle, resulting in packet serialization delay and intra-router contentions [6]. These limitations become significant performance bottlenecks when handling throughput-intensive traffic in heterogeneous systems. Furthermore, the number of circuit-switched paths is fundamentally limited by the number of planes, which cannot be increased arbitrarily. As NoCs scale up in size, or as many-to-few-to-many traffic pattern increases with the integration of data-parallel many-core accelerators [7], the number of planes becomes insufficient for reserving circuit-switched paths. Thus, we must seek an alternative sharing mechanism for hybrid-switched NoCs in the presence of heterogeneous many-core systems.

To address the limitations of SDM-based NoC, we propose a hybrid-switched NoC in which packet- and circuit-switched messages share the communication fabric through time-division multiplexing (TDM). In a TDM-based hybrid-

1530-2075/14 $31.00 © 2014 IEEE
DOI 10.1109/IPDPS.2014.40

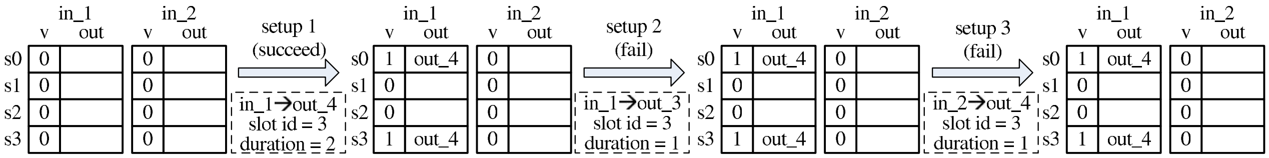


Figure 1. Slot table state transition of a single router, in response to 3 setup messages. In a setup message, slot_id specifies the initial time slot when a reservation begins; duration specifies the number of consecutive slots the reservation requires.

switched network, time is discretized into recurrent time-slots. In a given time-slot, the available link bandwidth is either allocated to the packet-switched network, or it is exclusively dedicated to a given circuit-switched path. Compared to an SDM-based NoC, TDM is able to avoid serialization observed in SDM-based NoC because each flit can utilize the full link bandwidth when transmitting. Consequently, each message results in fewer flits, and fewer flits per message can reduce the risk for congestion. Furthermore, by partitioning the network bandwidth in time domain, the number of possible circuit-switched paths is theoretically unlimited. In reality, with minimal additional hardware, a TDM-based NoC can support a large number of circuit-switched paths. We provide quantitative performance comparison between SDM- and TDM-based hybrid switching in Section IV using synthetic traffic.

The differences between packet- and circuit-switched interconnection architecture provide us with opportunities to optimize NoC performance as well as reduce energy consumption. In heterogeneous systems, data parallel cores generate throughput intensive traffic. Circuit switching this traffic can significantly reduce energy overhead. Superscalar cores, on the contrary, have moderate throughput requirements. Thus, circuit switching this traffic has only a moderate energy benefit compared to that of data parallel cores. However, superscalar cores can benefit from the latency reduction associated with circuit switching.

Within the context of heterogeneous multicore systems, this paper makes the following contributions:

- Introducing a hybrid-switched NoC that enables packet- and circuit-switched messages to share the same underlying communication fabric through Time-Division-Multiplexing (TDM).
- Demonstrating how canonical routers can be extended to support hybrid switching, and how hybrid switching can be tightly coupled with the system.
- Demonstrating the energy efficiency and scalability of TDM-based hybrid-switched NoC for supporting heterogeneous multicore systems with both synthesized and realistic workloads.

II. TDM-BASED HYBRID-SWITCHED NETWORK

A hybrid-switched network allows both packet- and circuit-switched messages to traverse through the network concurrently. In particular, packet-switched messages are buffered, routed and then forwarded at each router; while circuit-switched messages follow dedicated paths without incurring additional buffering/routing overhead. These dedicated paths are setup with explicit configuration messages.

In a TDM-based network, links are shared through the use of time-slots. The assignment of time-slot is kept in slot tables that are maintained at each router. Each slot table entry contains a valid bit and an output port id. For each incoming flit, the router looks up the slot table and determines whether the flit should be packet- or circuit-switched. Slot table entries are updated by explicit path configuration messages. Figure 1 demonstrates the path configuration process as seen by a single router. For simplicity, only two input ports are shown in this example. Configuration messages arrive at either of these two ports.

1) Initially, no path is reserved and all slot table entries are invalid.
2) A setup message $setup1$ arrives at input port in_1 and requests output port out_4, reserving two consecutive cycles starting from time-slot $s3$. Since all of the tables are empty, $setup1$ is successful and the corresponding slot table is updated. Notice that slot reservation is performed in modulo S fashion (S is the size of the slot table), so $s3$ and $s0$ are reserved for $setup1$.
3) $Setup2$ reserves time slot $s3$ from in_1 to out_3. This setup would fail because the slot has already been allocated. The slot tables remain unchanged and a setup failure acknowledgement is sent back.
4) $Setup3$ reserves time slot $s3$ from in_2 to out_4. This setup would fail because out_4 is already reserved for in_1 at slot $s3$. In this case, setup failed due to a conflict at the output port.
5) Teardown messages are used to remove circuit-switched paths when they are no longer needed. When a teardown message arrives, the valid bits of the corresponding entries are reset so that the slots can be reused by other paths.

No circuit-switched messages should arrive if the time slot is not reserved. However, if a time-slot is reserved for circuit-switched flits and no circuit-switched flit arrived in this cycle, a packet-switched flit will steal this slot. We refer to this technique as *time-slot stealing*. Detailed discussion can be found in Section II-D. In the following sections, we elaborate four design choices to improve network utilization.

A. Switching Decision

To improve network performance and reduce network energy consumption, deciding whether a message should be forwarded with packet or circuit switching is multifaceted. From energy perspective, as more messages traverse the network through circuit switching, less energy is consumed for buffering and routing; on the other hand, maintaining the slot tables incurs energy overhead. From performance perspective, circuit-switched messages spend less time traversing the network; however, messages might stall and wait for a circuit-switched time slot before transmitting. In the proposed design, a circuit-switched path is only reserved for source-destination pairs communicate frequently to ensure a high utilization of the path. For example, when a core generates a large number of memory accesses, the NoC will reserve circuit-switched paths between the core and the memory components. Once a path is established, not all messages are circuit-switched. Consider the case in which a circuit-switched message is stalled, waiting for its time slots. Such phenomenon leads to increase in network latency when messages only traverse short distances within the network. This issue can be mitigated by allowing a message to be packet-switched if the established path corresponds to a time slot that requires stalling. In general, switching decision is based on its impact on system performance. Further description can be found in Section V-A2.

B. Path Configuration

Once the source node has decided to construct a dedicated connection, circuit switching path configuration is performed. In the proposed hybrid-switched network, circuit-switched paths are set up and torn down through explicit configuration messages, which are sent through the packet-switched network. There are three types of configuration messages: $setup\_msg$, which creates a circuit-switched connection between the source and the destination; $teardown\_msg$, which destroys an existing connection; and ack_msg, which indicates a setup success/failure.

A $setup\_msg$ contains the source and destination node id, and a slot id indicating the time slot to reserve. Every router along the path of the $setup\_msg$ checks the availability of the output port at given slots. If it is free, the output port is reserved, and the $setup\_msg$ is forwarded with slot id incremented by 2 since the circuit-switched network is two-stage-pipelined (the increment is in modulo S fashion, where S is the number of entries in a slot table). Otherwise, the setup is aborted and an acknowledgement is sent back to the source node, indicating a setup failure. If the $setup\_msg$ successfully reaches the destination, an acknowledgement will also be generated with an ack_msg sent back to the source node, indicating a path setup success. After ack_msg reaches the source node, depending on the path setup success/failure, either: (a) the source-destination connection is registered at the source node, and packets can be sent in circuit-switched fashion; or (b) a $teardown\_msg$ is generated to destroy the corresponding connections.

A $teardown\_msg$ contains the same information as a $setup\_msg$. It traverses through the same path as the corresponding $setup\_msg$ by referring to the slot tables. Therefore, the $teardown\_msg$ will eventually arrive at the node where the setup failed, and invalidate all reserved slots. Acknowledgement is not required for a $teardown\_msg$. Since there is no guarantee that all source-destination connections can be set up successfully in a single try, we allow re-sending the failed setup with a different slot id. However, if the slot tables are too small to hold all possible source-destination reservations, source nodes will repeatedly and unsuccessfully try to set up a connection. We do not implement complex path setup algorithms since the re-send mechanism introduces minimal performance penalty. By selecting the slot table size properly, the overhead can be minimal. In our experiment, configuration messages correspond to less than 1% of total traffic.

A reserved connection can be repeatedly used by the same source-destination pair. Once a connection has been idled for a long period, it becomes the candidate to be destroyed when new setup requests come in. If all entries inside a slot table are reserved by circuit-switched connections, packet-switched messages might suffer starvation. Although unlikely to happen, in order to prevent starvation, slot allocations are prohibited when the percentage of reserved entries exceeds a threshold. In this work, we arbitrarily set the threshold to 90%. It is worth pointing out that in the proposed network, packet transmission does not wait for a successful circuit-switched path setup, which means a message can be sent through packet-switched network WHILE its path setup is performed in parallel. Therefore, the performance overhead caused by path setup is negligible.

Reserving consecutive slots: When reserving slot table entries, we can either reserve a single slot, or multiple slots consecutively. In this work, consecutive slot reservation is preferred because the entire message can reach its destination within a few cycles. The reservation duration depends on the data length, which is 4 slots in our setup since a cache line is 64-byte long and the flit width is 16 bytes. A duration field is added into the construction messages indicating the number of slots each reservation requires.

Path selection: During circuit-switched path setup, judiciously selecting paths can potentially reduce congestions and balance workload across all routers. To achieve the goal, we apply adaptive routing algorithm to configuration messages. Details of the algorithm can be found in [8].

C. Time-division Granularity

Time-division granularity corresponds to the percentage of bandwidth reserved for each connection, and in turn determines how often a circuit-switched message for a particular reservation can travel on the reserved path. It also determines the number of slot table entries. Smaller slot tables correspond to coarser granularity and a smaller interval between consecutive circuit-switched messages on the same path. However, smaller slot tables also imply fewer circuit-switched paths. Larger slot tables, on the other hand, provide finer granularity and can hold more reservations. Nevertheless, messages might stall for more cycles before transmission. Furthermore, larger slot tables also lead to more energy consumption.

In a hybrid-switched NoC, slot table size is an important design parameter and can be workload dependent. We propose to dynamically determine the slot table size. In particular, we start with activating only a small portion of the slot tables and power-gating the rest, and then double the

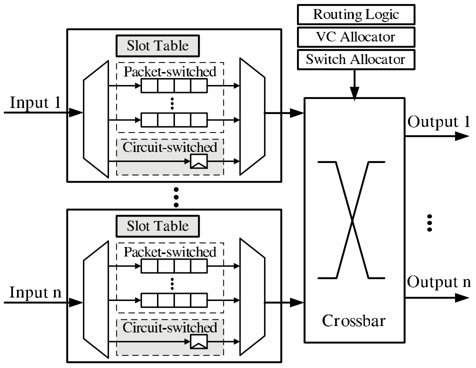


Figure 2. Hybrid-switched router structure

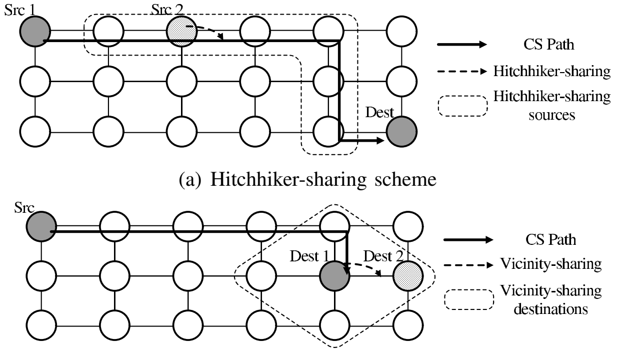
number of active entries as needed until all entries are activated. In other words, the slot table size is a function of the network size as well as the number of circuit-switched paths. While the former can be statically determined, the latter is workload dependent. More slot table entries are activated when path allocation continuously fails. Once the capacity of the slot table is increased, all slot tables are reset, and the path setup procedure restarts. Since program behaviors are relatively stable, we expect slot table reconfiguration to rarely occur.

D. Hybrid-Switched Router Architecture

We extend the virtual-channeled wormhole router to support hybrid switching. The router architecture is shown in Figure 2. The following components are added to a classical router architecture [9]: slot tables, circuit-switched latches, and demultiplexers which forward incoming messages to either the packet- or the circuit-switched pipeline.

At time T, an arriving flit is forwarded to either the packet- or the circuit-switched pipeline, depending on the entry in the slot table corresponding to time T. Before a circuit-switched flit's arrival, the crossbar is configured in advance by retrieving the output port information from the slot table, and the reserved output is guaranteed to be available in T. Thus, the flit simply proceeds through the router in a single cycle without buffering. It reaches the downstream router in time $T + 2$ after the link transmission stage, which takes another cycle. Packet-switched flits, on the other hand, traverse through the router pipeline.

In a reserved time-slot, it is possible that no circuit-switched flit is presented, leaving the crossbar idle. When this happens, we allow a packet-switched flit to utilize the crossbar, referred to as *time-slot stealing*. To enable time-slot stealing in time T, the router must be informed at time $T - 1$ whether a circuit-switched flit will arrive in the next cycle. A designated one-bit signal wire can propagate this information from the upstream router to the downstream router. If a circuit-switched flit is arriving, the signal will be enabled when the crossbar from the upstream router finishes its connection allocation. Otherwise, time-slot stealing can be performed in the downstream router.



(a) Hitchhiker-sharing scheme

(b) Vicinity-sharing scheme

Figure 3. Circuit-switched path sharing mechanism

III. Optimizations for Hybrid Switching

In this section, we describe two optimizations to enhance the resource utilization and energy-efficiency of the hybrid-switched network.

A. Circuit-switched Path Sharing

In the hybrid router described in Section II, each slot table entry is reserved for a particular source-destination pair. When the communication path of a source-destination pair partially overlaps an existing path, sharing the slot table entries of the existing path, rather than reserve a new path, can potentially improve the utilization of the slot table. In this section, we propose hitchhiker-sharing that allows intermediate nodes along a circuit-switched path to share the connection if the message is sent towards the same destination as the reserved slots; and vicinity-sharing that allows path sharing if the destination of a message is in the vicinity of the destination node of some reserved path. Compared to time-slot stealing, which enables packet-switched traffic to utilize an unused time-slot reserved for circuit-switched traffic, the proposed path sharing mechanisms allows multiple circuit-switched paths to share entries in the slot tables.

1) Hitchhiker-sharing: As shown in Figure 3(a), a circuit-switched path is already constructed from source node $Src1$ to destination node $Dest$. If another source node, for example $Src2$, resides along the circuit-switched path, it is allowed to share the circuit-switched connection with $Src1$ when the corresponding time slots are not occupied by messages from $Src1$. To make decision of whether hitchhiker-sharing is feasible, $Src2$ has to know the destinations of circuit-switched connection presented in its slot tables, and the time slots reserved for these connection. This information is stored in a *Destination Lookup Table* (DLT) in $Src2$, and is updated when a new connection is setup in the router of $Src2$. When sending a message, if the circuit-switched path originated from $Src2$ does not exist but hitchhiker-sharing can be performed, the message is still considered to be circuit-switched. If the path sharing fails due to contention, this message will be sent in packet-switched manner. If path sharing towards a particular destination fail continuously, a circuit-switched path setup request is thereby generated from $Src2$. In Figure

3(a), nodes circled by dashed line are candidates that can potentially share the circuit-switched path with $Src1$.

Hitchhiker-sharing scheme does not require modifications to the slot table. Instead, the DLT is required for each node to store the destination and its corresponding time-slot information. The size of the DLT is dependent on the network size and slot table size. For example, in a k-by-k network, $2\lceil log_2 k \rceil$ bits are required to represent the destinations. Meanwhile, for slot tables with S entries, $\lceil log_2 S \rceil$ bits are needed to indicate the time-slot for each destination. Moreover, we introduce a 2-bit saturation counter to keep track of the failure of circuit-switched path sharing. If the counter becomes '10', a circuit-switched path setup is generated and the entry is removed from the table. The size of an 8-entry DLT in the evaluated system described in Section V-A is less than 16 bytes.

2) Vicinity-sharing: Figure 3(b) is an example of vicinity-sharing scheme. A circuit-switched path is set up from Src to $Dest1$. If a message is sent from Src to another destination, for example $Dest2$, but dedicated circuit-switched path has not been constructed. Instead of setting up a new connection, this message is allowed to use the existing circuit-switched path between Src and $Dest1$, as long as $Dest2$ and $Dest1$ are adjacent. After reaching $Dest1$, the message will be sent towards it destination through packet-switched network. Similar to hitchhiker-sharing scheme, when contention occurs at the source node, path sharing fails and the message is again packet-switched. A circuit-switched path setup requirement is generated if path sharing fail continuously. In Figure 3(a), vicinity-sharing destination candidates are circled by dashed line.

To support vicinity-sharing, a 2-bit saturation counter is required for each reservation to keep track of the circuit-switched path sharing failure. Moreover, comparison logics are used to compute the vicinity-sharing candidate nodes. A header flit is required in addition for vicinity-sharing, because messages go through packet-switched network after hop-off. Therefore, when reserving circuit-switched paths, one additional time slot is required.

Notice that hitchhiker-sharing and vicinity-sharing can be combined and applied to the same circuit-switched path. In other words, messages can hop-on at intermediate nodes and get off at nodes close to their destination. With circuit-switched path sharing deployed, most of the frequent connections can still be satisfied even with smaller slot tables.

B. Aggressive VC Power Gating

Since the circuit-switched network alleviates the burden on packet-switched counterpart, we can turn off under-utilized router buffers to save static energy while still satisfying the performance requirement. Increasing the number of VCs allows the network to support a higher injection ratio without significant increase in network latency. However, when the buffer pressure is alleviated by circuit switching a portion of the traffic, fewer VCs are needed. Thus, we propose a dynamic VC tuning policy where the number of activate VCs is periodically adjusted based on VC utilization (μ) in comparison to two thresholds, $Threshold_{High}$ and $Threshold_{Low}$. If μ exceeds $Threshold_{High}$, one set of virtual channel will be activated; if μ is below $Threshold_{Low}$, one set of virtual channel will be turned off. The VC must be evacuated before adjusting, and the downstream routers are updated with the new VC count.

IV. Evaluation on Synthetic Workload

Compared to real applications, synthetic traffic makes it possible to study the NoC behavior under a wider range of on-chip traffic. In this section, we evaluate the performance and energy consumption impact of the proposed NoC with three traffic patterns: 1) uniform random(UR): destinations are randomly selected; 2) tornado(TOR): messages from (x, y) are sent to $(x + \frac{k}{2} - 1, y)$, where k is the number of nodes in both x and y dimensions; and 3) transpose(TR): messages from (x, y) are sent to (y, x) [10]. In addition, we also compare the performance of our proposed work against [5].

A. Evaluation Infrastructure

The interconnection network and its power model are based on Garnet [9] and Orion 2.0 [11], respectively. We update the router power model with the hybrid-switched architecture. We consider a 36-node mesh with router parameters shown in Table I. The network is warmed up with 1000 packets and simulated for 100,000 packets.

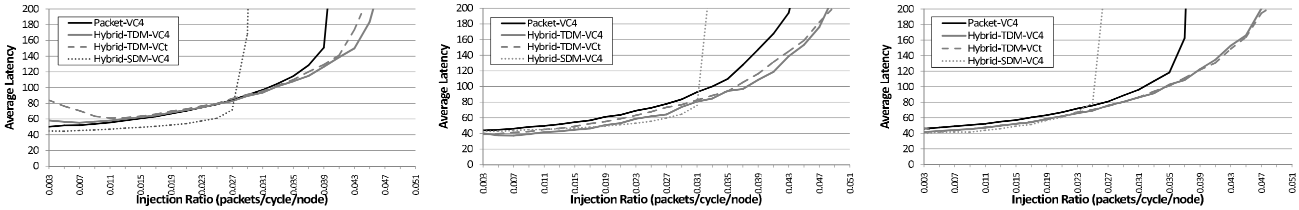
To compensate for overestimation in router area and inaccuracy in router power in Orion 2.0, as described in [12], [13], we revise the technology parameter in Orion 2.0 and augment our router with an RTL implementation. The SRAM bit cell spacing has been updated based on [13]. To avoid the inaccuracy in the multiplexer-based crossbar model, we assume a matrix crossbar in our evaluation. Furthermore, we use an RTL model to adjust the router area [14]. Both packet- and hybrid-switched routers are synthesized using Nangate Open Cell Library for 45nm technology. The total area of a packet-switched router is $0.177mm^2$; and $0.188mm^2$ for a hybrid-switched router, leading to 6.2% area overhead.

Table I
Router Parameters

Topology	36-node, 2D-Mesh
Technology	45nm technology at 1.0V, 1.5GHz
Routing	Minimal Adaptive Routing (configuration packet) X-Y Routing (other packet)
Channel Width	16 Bytes
Packet Size	1 flit (configuration message) 4 flits (circuit-switched packet) 5 flits (packet-switched packet, circuit-switched packet when vicinity-sharing applied)
Slot Tables	128 entries
Virtual Channels	4/port
Buffer size per VC	5 in depth

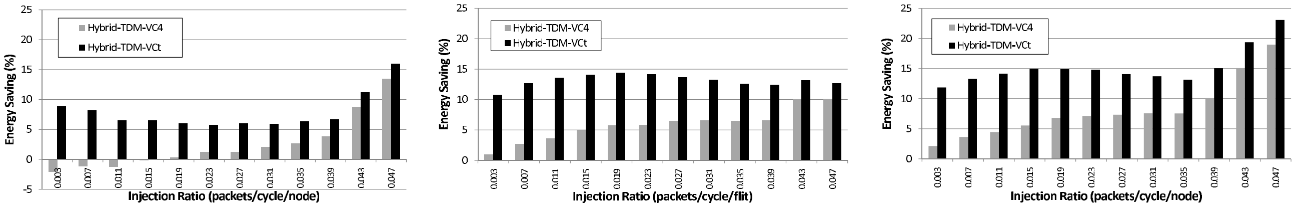
B. Impact on Performance

Figure 4 shows the load-latency curves with different traffic patterns for baseline packet switching with 4VCs(Packet-VC4), SDM-based hybrid switching with



(a) Uniform Random (b) Tornado (c) Transpose

Figure 4. Average latency as a function of injection rate with synthetic traffic patterns.



(a) Uniform Random (b) Tornado (c) Transpose

Figure 5. Network energy saving as a function of injection rate with synthetic traffic. Results are compared to network with 4-VC packet-switched routers.

4VCs(Hybrid-SDM-VC4) [5], TDM-based hybrid switching with 4VCs(Hybrid-TDM-VC4), and TDM-based hybrid switching with aggressive power gating(Hybrid-TDM-VCt). For the evaluated traffic patterns, TDM-based hybrid-switched routers improve the throughput by 14.7%, 9.3%, and 27.0%, respectively. Forwarding messages through circuit switching alleviates the buffer pressure and router congestion, which in turn improves the throughput. Exception is found in UR traffic, where TDM-based NoC suffers longer latency. This is because the TDM network required large slot tables to capture as many communication pairs as possible. Therefore, a circuit-switched packet has to wait for long time before its time-slot arrives, resulting in longer delay.

Compared to the TDM-based NoC, the SDM-based NoC reduces latency for all traffic patterns under low/moderate injection rates, while the proposed TDM-based NoC is able to achieve a higher level of throughput. The TDM-based NoC is competitive in latency reduction in TOR and the TR traffic, but not in UR traffic for reasons explained above. However, the SDM-based NoC is unable to scale as the injection rate increases. This is because in wormhole switching, the SDM-based NoC must serialize packets as they pass through a single plane. This serialization increases the number of flits per packet and increases the risk for congestion and intra-router contention. Eventually, this serialization manifests as a penalty in throughput at high injection rates.

It is worth noticing that the performance of the SDM-based NoC can improve with increased link bandwidth, because packet serialization becomes less serious with wider links for a fixed packet size. On the other hand, as the network increases in size, the performance gap between the TDM- and the SMD-based NoC will widen. This is because more circuit-switched paths are expected in larger networks, but SDM-based NoC switching is unable to support a large number of paths. Overall, we believe that TDM-based hybrid switching is more desirable for many-core systems in the presence of throughput-intensive traffic.

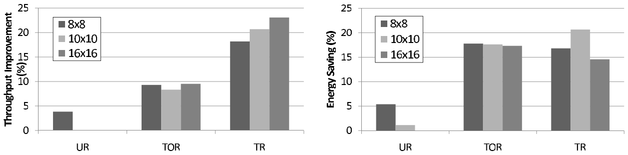
C. Impact on Energy

Figure 5 shows the network energy saving compared to the baseline *Packet-VC4*. SDM-based NoC is not shown because it increases the network energy consumption. The energy saving under uniform random traffic is relatively small. Negative energy saving is observed under low injection rate. This is again because large slot tables are required to capture all possible communication pairs in uniform random traffic. Under low injection rate, the energy saving from circuit switching messages does not offset the overhead caused by large slot tables. In other words, when the number of source-destination pairs is reasonable so that most circuit-switched reservations can be held in slot tables, the hybrid-switched network can efficiently reduce the network energy consumption.

VC tuning can reduce energy consumption without affecting the throughput significantly. By selecting $Threshold_{High}$ and $Threshold_{Low}$ properly based on network throughput characteristics, sufficient VCs can be guaranteed to prevent saturation. *Hybrid-TDM-VCt* outperforms *Hybrid-TDM-VC4* in terms of energy saving. However, as injection ratio increases, the gap between the two reduces. This is due to the fact that when more flits are injected, VC buffers become busier. More VCs have to be activated to satisfy the throughput requirement, resulting in fewer opportunities for aggressive VC power gating. Overall, *Hybrid-TDM-VCt* achieved additional energy saving over *Hybrid-TDM-VC4* by 2.4%-10.9% under uniform random traffic; 2.6%-10.0% under tornado traffic; and 4.1%-9.7% under transpose traffic.

D. Scalability

To study the scalability of the proposed NoC, we scale the network size from 64 nodes (8-by-8) to 256 nodes (16-by-16), and evaluate the maximum throughput improvement and energy savings of *Hybrid-TDM-VCt* compared to *Packet-VC4*. We also increase the slot table size to 256 for the larger network because there are more source-destination



(a) Max Throughput Improvement (b) Network Energy Saving

Figure 6. Throughput improvement and network energy saving of *Hybrid-TDM-VCt* in larger networks. Energy result is sampled at 75% capacity before *Packet-VC4* saturates.

pairs. Figure 6(a) and 6(b) show the throughput improvement and energy saving, respectively. For most workloads, TDM-based hybrid-switched network can achieve the same throughput improvement and energy saving as the network scales in size. However, for the uniform random workload, the performance and energy benefit from hybrid switching is relatively low for a small network, and the benefit become negligible as network size increases. This is because the number of frequent communication pairs increases quadratically as the network size grows. Moreover, communication pairs in uniform random traffic are distributed across the network with equal packet injection possibilities. Therefore, only a small fraction of the traffic benefit from circuit switching given limited size slot tables. However, we expect realistic workload to exhibits some form of regularity and can benefit from hybrid switching.

V. Evaluation on Realistic Workload

In this section, we present a detailed evaluation of the proposed TDM-based hybrid-switched network using realistic workloads.

A. *Evaluation Infrastructure*

We build a heterogeneous multicore simulator that integrates both superscalar-based CPU cores and data-parallel accelerators. CPU cores and the memory hierarchy is modeled after the Simics-based [15] GEMS [16] simulator. The accelerators are modeled with GPGPU-Sim [17]. The system configuration can be found in Table II.

1) Application Workload: We use applications from the SPEC OMP 2001[18] benchmark suites as CPU workloads, and applications from [17] and Rodinia [19] benchmark suites to simulate accelerator workloads. We consider 8 CPU benchmarks and 7 GPU benchmarks, where CPU benchmarks include: Ammp, Applu, Art, Equake, Gafort, Mgrid, Swim, and Wupwise; and GPU benchmarks include: Blackscholes, Lps, Lib, Nn, Hotspot, Pathfinder and Sto. Heterogeneous CPU-GPU workload is created by executing a multi-threaded CPU benchmark on the CPU cores and one GPU kernel across all the GPU cores. In each simulation, we sample a period of execution that corresponds to 500 million CPU instructions. We enumerate all possible combinations of the CPU and GPU workloads and evaluate 56 workload mixes.

2) Circuit Switching Decisions: To achieve optimal utilization of the network and minimize network energy consumption, we packet switch all CPU traffic while hybrid switch only GPU messages, since GPU traffic has a higher throughput requirement compared to CPU traffic. However, not all GPU messages are circuit-switched. A GPU message is considered to be circuit-switched only when no performance penalty is caused. Circuit switching a message can potentially increase or decrease transmission latency. While in GPUs, a delayed message may not necessarily cause performance degradation. The number of available warps in an SM can be used as an indicator to imply whether circuit switching a message causes performance penalty [8]. In our work, we estimate the GPU message slack by referring to the number of available warps. If the slack is greater than the overall circuit-switched transmission latency, we deliver the message through circuit-switched network. More sophisticated decision process can lead to better performance, however, a detailed investigation of the policy it is beyond the scope of this paper.

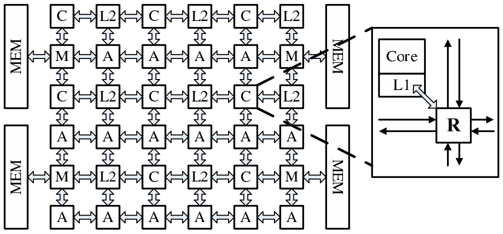


Figure 7. Evaluated multicore system

Table II
Baseline System Configuration

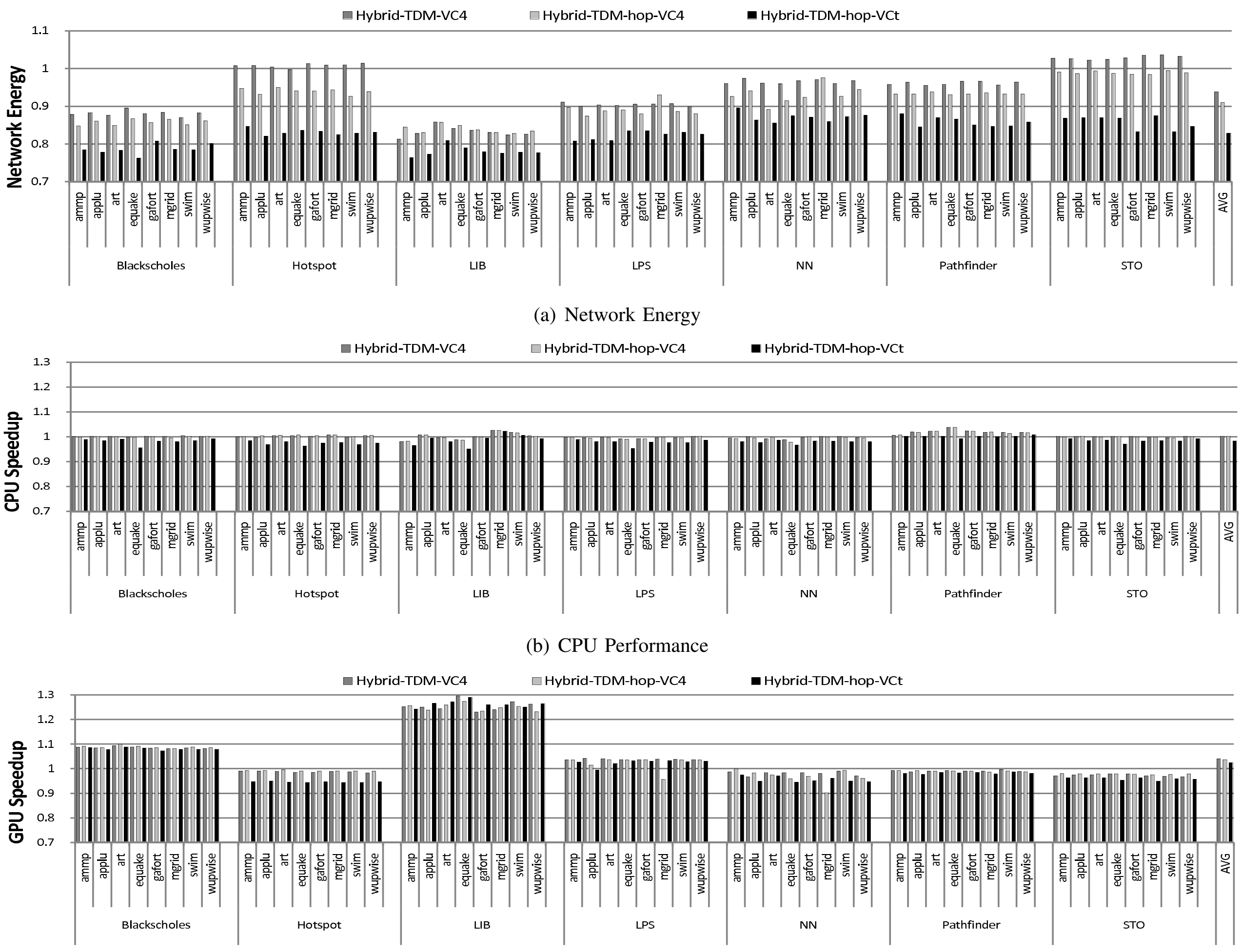
CPU Configuration	
Processor	Four-way out-of-order, 6 integer FUs, 4 floating point FUs, 128-entry ROB
L1 Cache	Split private I/D caches, each 64KB, 2-way set associative, 64B block size, 1-cycle access latency
L2 Cache	16M banked, shared distributed, 4-way set associative, 64B block size, 8-cycle access latency
Accelerator Configuration	
Accelerator	32-wide SIMD pipeline, 1024 threads, 32KB shared memory
Memory Configuration	
Memory	4GB DRAM, 200 cycle access latency, 4 memory controllers

B. *Experimental Results*

Figure 7 shows a 36-tile system connected with routers to a 6-by-6 mesh network. A tile consisting of a CPU core as well as an L1 cache is denoted by *C*. A tile that consists of a bank of shared L2 cache is denoted by *L2*, and a tile in which an accelerator resides is denoted by *A*. Off-chip memories are connected via memory controllers labeled with *M*. Each tile is equipped with a hybrid-switched router *R*. Heterogeneous traffic is generated from different nodes, and such structure can be easily extended to a larger system with more tiles.

1) Overall Energy Efficiency: Figure 8 shows the network energy saving, which is normalized to the baseline 4-VC packet-switched network; CPU and GPU application speedup, which is computed against the same baseline 4-VC packet-switched network. X-axis is the workload mix, grouped by GPU benchmarks. The last set of bars *AVG* is the geometric mean across all workload mixes.

Observing from Figure 8(a), the proposed hybrid-switched NoC reduces the network energy consumption significantly.



(a) Network Energy

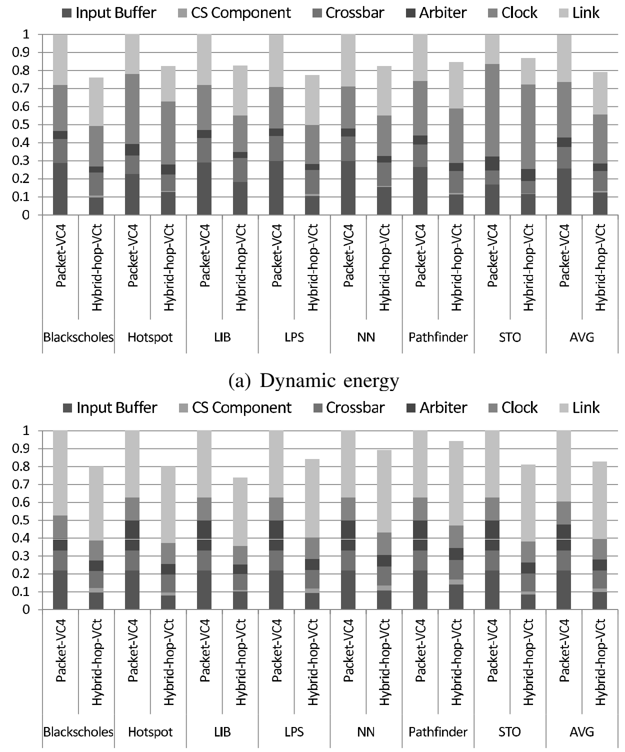
(b) CPU Performance

(c) GPU Performance

Figure 8. Network energy and performance of hybrid-switched network. *Hybrid-TDM-VC4* corresponds to network using 4-VC hybrid-switched routers; *Hybrid-TDM-hop-VC4* corresponds to network using 4-VC hybrid-switched routers with circuit-switched path sharing; *Hybrid-TDM-hop-VCt* corresponds to network using 4-VC hybrid-switched routers with both circuit-switched path sharing and aggressive VC power gating techniques deployed. Energy results are normalized to the baseline network with 4-VC packet-switched network, which is not shown. Speed up is computed against the same baseline.

Different CPU workloads have minimal impact on network energy saving, because CPU packets only take up a small portion of the entire on-chip traffic, while the variation in energy saving corresponds to GPU workload. On average, the basic hybrid-switched scheme reduces network energy consumption by 6.3%. With path sharing and aggressive VC power gating techniques deployed, 9.0% and 17.1% of energy saving is achieved, respectively. In particular, up to 23.8% of energy saving is achieved in BLACKSCHOLES. However, in STO, more energy dissipation is required for *Hybrid-TDM-VC4*. This is because STO has relatively low throughput requirement with only a small amount of messages sent through the circuit-switched network, thus dynamic energy saving is insignificant. Moreover, adding slot tables into routers introduces extra static energy consumption. Therefore, applications benefit from hybrid-switched router only when the reduction of dynamic energy offsets the increase of static energy overhead. Although the basic scheme is not efficient for applications with low injection rate, circuit-switched path sharing mechanism allows more packets to be sent through circuit-switched network with smaller slot tables; furthermore, aggressive VC power gating technique turns off inactive buffers so that the static energy consumption can be reduced significantly. Overall, by applying aggressive optimizations, considerable amount of energy saving is achieved in STO.

CPU application performance is presented in Figure 8(b). CPU performance is hardly affected in most applications. This is mainly because CPU messages are sent through packet-switched network, and are merely delayed slightly when competing with circuit-switched flits in the router. Since not all CPU messages are critical [20], such delay affects the performance trivially. GPU performance varies as shown in Figure 8(c). The worse case comes from STO, where all circuit-switched network schemes suffer from performance degradation by over 2%.



(a) Dynamic energy

(b) Static energy

Figure 9. Detailed network energy breakdown. Results are grouped by GPU benchmarks. Each bar is an average over all CPU applications.

Table III
GPU TRAFFIC INJECTION RATE AND PERCENTAGE OF FLITS THAT ARE CIRCUIT-SWITCHED FOR *Hybrid-TDM-VC4*

GPU benchmarks	Injection ratio (flits/node/cycle)	Circuit-switched flits percent (%)
Blackscholes	0.18	55.7
Hotspot	0.09	29.1
Lib	0.20	34.4
Lps	0.20	55.0
Nn	0.18	38.9
Pathfinder	0.13	49.1
Sto	0.05	18.5

Overall, the proposed hybrid-switched network is able to reduce the network energy consumption by 17.1% with only 1.6% of CPU performance degradation and 2.6% of GPU performance improvement, on average. Therefore, the hybrid-switched network is energy efficient in the evaluated heterogeneous system.

Dynamic energy saving: Figure 9(a) demonstrates the breakdown of dynamic energy dissipation, including input buffers, circuit switching (CS) components, crossbars, VC/SW arbiters, clock, and link energy. All hardware introduced for circuit switching are referred to as CS Component in this figure. Due to the avoidance of buffer read/write, hybrid-switched network reduces the buffer energy by 51.3%, on average. The overhead caused by circuit switching is 0.6%, on average. Both buffer energy saving and circuit switching overhead are dependent on the percentage of traffic sent through the circuit-switched network, as shown in Table III. Higher circuit-switched percentage corresponds to more energy saving while larger circuit switching overhead. Savings from crossbars, links and arbiters are negligible. This is because both circuit- and packet-switched flits have to pass through crossbars and link wires; and arbiters only correspond to a small portion of dynamic energy consumption. Overall, 20.8% of dynamic energy reduction is achieved with hybrid-switched routers.

Static energy saving: Dynamic energy is expected to reduce as technology advances due to smaller device size, but static energy is expected to increase. Our proposed router exploits the opportunities of turning off under utilized components as discussed in Section III-B, so that static energy dissipation can be reduced as well. Dynamic voltage-and-frequency scaling (DVFS) can be applied orthogonally to our technique to mitigate clock energy largely, but is beyond the scope of this paper. Figure 9(b) shows static energy breakdown. On average, 17.3% of static energy saving is achieved with 2.1% overhead caused by circuit-switched components. As expected, all the savings come from input buffers. Introducing slot tables enables bandwidth sharing, but brings in additional overhead. Lib suffers from insignificant overhead because it has fewer communication pairs compared to other GPU applications, hence smaller slot tables are sufficient to hold all circuit-switched paths. It is worth pointing out that compared to packet-switched network with VC power gating (not shown), 6.8% static energy saving is achieved, proving that hybrid-switched network can alleviate buffer pressure and enable further VC power gating.

2) Performance Analysis: Hybrid-switched network brings in minimal performance penalty even with aggressive energy saving techniques deployed. As shown in Figure 8(b), overall CPU performance impact is negligible. Reducing the number of VCs degrades the network throughput, which slightly affects the CPU application performance by 1.6%. For GPU performance in Figure 8(c), Nn, Pathfinder, and Sto suffer from slight performance degradation while Blackscholes and Lib achieve over 9% of speedup. Performance penalty is mainly due to the delay of critical messages. Circuit-switched network might affect message delay in two ways: Firstly, buffering circuit-switched messages at the source increases the queueing delay; secondly, when contentions occur in routers, packet-switched messages which have a lower priority always give way to circuit-switched messages, therefore the network latency of packet-switched messages increases. If the transmission of critical message is delayed, system performance will be harmed. On the other hand, critical messages can possibly reach their destination faster in a hybrid-switched network, in which case the system performance will be improved as shown in Blackscholes and Lib. Recall that in Section V-A2 only simple policy is used for deciding which messages should be circuit-switched, accurate performance monitors can be referred in order to avoid performance penalty.

3) Effectiveness of Circuit-switched Path Sharing: Compared to *Hybrid-TDM-VC4*, network energy is further reduced by 2.8% in *Hybrid-TDM-hop-VC4*, on average. While the performance impact on both CPU and GPU applications is negligible. By allowing circuit-switched path sharing, a message can be sent through circuit-switched network without reserving a dedicated connection, provided that its route is partially overlapped with existing circuits. Better energy saving can therefore be achieved. Especially when traffic injection rate is not high, most of the circuit-switched connections are not fully utilized. Path sharing enables smaller slot tables being used. Moreover, when slot tables become smaller, the waiting time before a circuit-switched flit can be sent is reduced. As a result, more flits can be considered to be circuit-switched as performance constraint is easier to satisfy. Meanwhile, aggressive VC power gating can potentially benefit from circuit-switched path sharing as VC buffers becomes less busy. Therefore, path sharing promotes aggressive network energy saving and is a critical design in the hybrid-switched network.

4) Effectiveness of Aggressive VC Power Gating: The aggressive VC power gating policy described in Section III-B can be applied to both packet- and hybrid-switched NoC. In a heterogeneous system with various on-chip traffic patterns, sufficient VCs are required to satisfy the throughput requirement while guaranteeing the desired latency. Energy and performance trade-off can be accomplished by reducing the number of VCs in a router. To evaluate how hybrid-switched network enables further VC power gating, we compare against packet-switched network with VC power gating deployed (not shown in Figure 8 due to the density of data points). Results show that the hybrid-switched NoC further reduces the energy consumption by 10% on average, while providing better speedup. The energy savings come from 1) dynamic energy reduction due to circuit switching, and 2) static energy reduction due to the fact that input buffer pressure is alleviated by circuit-switched network, more buffers can be turned off. For some applications, performance degradation is seen for both packet- and hybrid-switched network, because the network fails to provide sufficient bandwidth after reducing VC numbers.

Overall, aggressive VC power gating policy reduces energy consumption significantly in applications with moderate on-chip communication requirement. We believe activating and deactivating VCs based on more accurate metrics, for example, packet latency, will ensure better performance.

VI. Related work

Energy in NoCs has been shown to be a significant contributor to the on-chip energy consumption [3], [21], [11]. A variety of techniques have been deployed to reduce NoC energy dissipation. Dynamic Voltage and Frequency Scaling (DVFS) is an effective technique for reducing energy consumption [22], [23], [24]. However, DVFS becomes a performance bottleneck in heterogeneous systems with significant throughput requirement since reducing network operation frequency increases packet delay and degrades throughput [8]. Although flexible-pipeline routers [25], [26] can be deployed to reduce the latency penalty by dynamically combining the router pipeline stages, degradation in throughput remains an issue which harms the performance of throughput-intensive applications.

In virtual-channeled packet-switched network, buffers account for a significant portion of energy consumption [3], [21], [11]. Reducing buffer read/write operations or even removing buffers from the routers can largely eliminate the network energy consumption. Express virtual channels (EVC) is proposed to avoid the need for packets to stop and be buffered at intermediate nodes, therefore saves buffer energy and improves throughput [27]. While sharing the motivations, EVC incurs significant hardware overhead due to the complexity for credit management to ensure buffer availability. This complexity limits the length of the express links, and limits the benefit EVC can exploit. In SCARAB [28] and BLESS [29], energy reduction is achieved by removing input/output buffers. However, the performance of bufferless designs degrades under heavy NoC workloads, thus these designs are not suitable for heterogeneous systems with throughput-intensive cores.

TDM circuit switching is proposed to provide bandwidth and latency guarantees in Æthereal NoC [30], in which time slots are reserved for each guaranteed flow along the path. However, circuit switching transmission starts after acknowledgement is sent back, and guaranteed flows are not allowed to use excess bandwidth when the network is under-utilized. MANGO NoC [31] reserves virtual channels for guaranteed flows, therefore large number of buffers as well as costly switching modules are required, leading to significant energy dissipation. Our work utilizes TDM to facilitate resource-sharing between packet- and circuit-switched traffic to reduce NoC energy consumption, which requires different implementations and optimizations such as time-slot stealing, circuit-switched path sharing, dynamic time-division granularity adjusting, and aggressive power gating. Reconfigurable circuit-switched NoCs, such as [32], take advantage of the deterministic traffic patterns of certain applications and create circuit-switched paths for these applications. These paths only handle circuit-switched traffic for fixed sources and destinations, thus are inadequate for heterogeneous multicore systems that have non-deterministic traffic patterns. SDM hybrid-switched NoCs have been proposed to provide hard QoS support in SoC [33], [34], [35], [36], [37]. It is possible for SDM to work together with the proposed TDM mechanism. However, the hardware and performance overhead of such hybrid systems must be carefully evaluated.

Jerger et al. proposed a SDM-based hybrid-switched NoC design with a prediction-based coherence protocol, enabling significant latency reduction [5]. However, their work introduces packet serialization delay, which affects the performance when handling high throughput traffic. SMART selectively circuit-switch packets using asynchronous repeaters [38]. However, SMART is unable to optimize multiflit packets and only allows for short circuit-switched paths. Instead of using TDM, Kilo-NoC [39] uses both VC routers and elastic buffer routers in certain nodes of the network, to reduce energy consumption and provide scalability. Kilo-NoC mainly focus on enabling QoS and ensure fairness inside NoC in large-scale systems.

VII. Conclusion

In this paper, we design, implement and evaluate a hybrid-switched network that supports both packet-switched and circuit-switched messages by partitioning the network through time-division multiplexing (TDM). By evaluating both synthesized and realistic traffic, we found that the proposed network is able to improve the energy-efficiency as well as the performance of on-chip network. We demonstrated the feasibility of implementing TDM-based hybrid switching by extending an existing router design with hybrid switching capability.

In the evaluated heterogeneous multicore system with hybrid-switched NoC deployed, the network energy consumption is reduced by as much as 24% and system performance is improved by as much as 12%, compared to the network using canonical packet-switched routers. With an adequate policy for setting up circuit-switched paths, as well as selecting and forwarding messages that can benefit from circuit switching, TDM-based hybrid-switched networks can efficiently reduce the network energy consumption, and even improve overall system performance. Moreover, optimizations such as circuit-switched path sharing and VC power gating are essential to achieve further energy savings. In conclusion, in heterogeneous multicore and manycore systems, we believe that TDM-based hybrid-switched NoC provides a viable alternative for achieving low-latency and high-throughput with a stringent power budget.

References

[1] "Intel. Sandy Bridge," http://software.intel.com/en-us/articles/sandy-bridge.

[2] "AMD. Fusion," http://sites.amd.com/us/fusion/apu/Pages/fusion.aspx.

[3] S. R. Vangal *et al.*, "An 80-Tile Sub-100-W TeraFLOPS Processor in 65-nm CMOS," *In JSSC*, vol. 43, 2008.

[4] A. Kumar *et al.*, "A 4.6Tbits/s 3.6GHz Single-cycle NoC Router with a Novel Switch Allocator," in *In ICCD*, 2007.

[5] N. D. E. Jerger *et al.*, "Circuit-switched coherence," in *In NOCS*, 2008.

[6] R. Das *et al.*, "Catnap: Energy Proportional Multiple Network-on-Chip," in *In ISCA*, 2013.

[7] A. Bakhoda *et al.*, "Throughput-Effective On-Chip Networks for Manycore Accelerators," in *In MICRO*, 2010.

[8] J. Yin *et al.*, "Energy-efficient Non-minimal Path On-chip Interconnection Network for Heterogeneous Systems," in *In ISLPED*, 2012.

[9] N. Agarwal *et al.*, "Garnet: A Detailed Interconnection Network Model inside a Full-system Simulation Framework," Princeton University, Tech. Rep., 2008.

[10] A. Singh *et al.*, "GOAL: A Load-balanced Adaptive Routing Algorithm for Torus Networks," in *In ISCA*, 2003.

[11] A. Kahng *et al.*, "ORION 2.0: A Fast and Accurate NoC Power and Area Model for Early-Stage Design Space Exploration," in *In DATE*, 2009.

[12] A. B. Kahng *et al.*, "Explicit Modeling of Control and Data for Improved NoC Router Estimation," in *In DAC*, 2012.

[13] M. Hayenga *et al.*, "Pitfalls of ORION-Based Simulation," in *In WDDD*, 2012.

[14] D. U. Becker, "Efficient Microarchitecture for Network-on-Chip Routers," *PhD thesis, Stanford University, August 2012*.

[15] P. Magnusson *et al.*, "Simics: A Full System Simulation Platform," *Computer*, vol. 35, 2002.

[16] M. M. K. Martin *et al.*, "Multifacet's General Execution-driven Multiprocessor Simulator (GEMS) Toolset," *SIGARCH Computer Architecture News*, vol. 33, 2005.

[17] A. Bakhoda *et al.*, "Analyzing CUDA Workloads Using a Detailed GPU Simulator," in *In ISPASS*, 2009.

[18] "SPEC OMP2001," Available at http://www.spec.org/omp/.

[19] S. Che *et al.*, "Rodinia: A Benchmark Suite for Heterogeneous Computing," in *In IISWC*, 2009.

[20] R. Das *et al.*, "Aergia: Exploiting Packet Latency Slack in On-chip Networks," in *In ISCA*, 2010.

[21] A. Banerjee *et al.*, "A power and energy exploration of network-on-chip architectures," in *In NOCS*, 2007.

[22] L. Shang *et al.*, "Dynamic Voltage Scaling with Links for Power Optimization of Interconnection Networks," in *In HPCA*, 2003.

[23] S. E. Lee *et al.*, "A Variable Frequency Link for a Power-aware Network-on-Chip (NoC)," *Integration, the VLSI Journal*, vol. 42, 2009.

[24] A. K. Mishra *et al.*, "A Case for Dynamic Frequency Tuning in On-chip Networks," in *In MICRO*, 2009.

[25] H. Matsutani *et al.*, "A Multi-Vdd Dynamic Variable-Pipeline On-Chip Router for CMPs," in *In ASP-DAC*, 2012.

[26] P. Zhou *et al.*, "NoC Frequency Scaling with Flexible-pipeline Routers," in *In ISLPED*, 2011.

[27] A. Kumar, L.-S. Peh *et al.*, "Express Virtual Channels: Towards the Ideal Interconnection Fabric," in *In ISCA*, 2007.

[28] M. Hayenga *et al.*, "SCARAB: A Single Cycle Adaptive Routing and Bufferless Network," in *In MICRO*, 2009.

[29] T. Moscibroda *et al.*, "A case for bufferless routing in on-chip networks," in *In ISCA*, 2009.

[30] K. Goossens *et al.*, "Æthereal Network on Chip: Concepts, Architectures, and Implementations," *IEEE Des. Test*, vol. 22, 2005.

[31] T. Bjerregaard *et al.*, "A Router Architecture for Connection-Oriented Service Guarantees in the MANGO Clockless Network-on-Chip," in *In DATE*, 2005.

[32] P. Wolkotte *et al.*, "An energy-efficient reconfigurable circuit-switched network-on-chip," in *In IPDPS*, 2005.

[33] F. Palumbo *et al.*, "Concurrent Hybrid Switching for Massively Parallel Systems-on-Chip: the CYBER Architecture," in *In CF*, 2012.

[34] A. K. Lusala *et al.*, "Combining SDM-Based Circuit Switching with Packet Switching in a Router for On-Chip Networks," *International Journal of Reconfigurable Computing*, vol. 2012.

[35] M. Modarressi *et al.*, "A hybrid packet-circuit switched on-chip network based on sdm," in *In DATE*, 2009.

[36] S. Secchi *et al.*, "A Network on Chip Architecture for Heterogeneous Traffic Support with Non-Exclusive Dual-Mode Switching," in *In DSD*, 2008.

[37] A. K. Lusala, "A SDM-TDM Based Circuit-switched Router for On-chip Networks," in *In ReCoSoC*, 2011.

[38] T. Krishna *et al.*, "Breaking the on-chip latency barrier using SMART," in *In HPCA*, 2013.

[39] B. Grot *et al.*, "Kilo-NOC: a Heterogeneous Network-on-Chip Architecture for Scalability and Service Guarantees," in *In ISCA*, 2011.