Variable-Width Datapath for On-Chip Network Static Power Reduction

George Michelogiannakis, John Shalf
Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720
Email: {mihelog, jshalf}@lbl.gov

***Abstract*—With the tight power budgets in modern large-scale chips and the unpredictability of application traffic, on-chip network designers are faced with the dilemma of designing for worst-case traffic loads and incurring high static power overheads, or designing for average traffic and risk degrading performance. This paper proposes adaptive bandwidth networks (ABNs) which divide channels and switches into lanes such that the network provides just the bandwidth necessary in each hop. ABNs also activate virtual channels (VCs) individually and take advantage of drowsy SRAM cells to eliminate false VC activations. In addition, ABNs readily tolerate silicon defects with just the extra cost of detection. For application benchmarks, ABNs reduce total power consumption by up to 45% with comparable performance compared to single-lane power-gated networks, and up to 33% compared to multi-network designs.**

I. Introduction

Current and future large-scale chips are increasingly constrained by power [15]. Modern on-chip networks contribute significantly to the chip's power, area, and performance characteristics [6], [21], [1], [13]. A challenge in reducing network power is designing the network independently of the system and applications, given that communication demands can vary substantially across different applications. Also, applications tend to load the network unevenly in both space and time [35], [9], [19], [3]. For example, channel utilization ranges from near zero to 43% in PARSEC benchmarks [19], [3], [5].

Designing the on-chip network to handle worst-case loads increases both area and static power compared to designing for average traffic. Static power is mainly composed of leakage power and the power to toggle clocking inputs, with leakage typically dominating [30]. Leakage power can constitute 90% of network power with light-traffic applications, or 30% to 50% with heavy-traffic benchmarks [13], [18], [24]. To make matters worse, leakage power is projected to increase in future near threshold voltage technologies even up to 90% of total power under higher loads [6], [23], [9].

In this paper, we propose adaptive bandwidth networks (ABNs) which continuously adapt to traffic load by activating the proper amount of bandwidth individually at each channel, and the proper number of virtual channels (VCs) in each input port. ABNs accomplish this by dividing channels into lanes. Lanes are activated individually according to local traffic demands. Inactive lanes are power gated, consuming near zero static power. ABNs also power gate individual VCs [30]. However, unlike past work, ABN use drowsy SRAM cells which enable ABNs to make activation decisions in the upstream router's VC allocator, thus avoiding mispredictions which can cause more VC activations than necessary [30]. ABNs also power gate router switches by adding multiple lanes for every input and output [18]. ABNs hide activation delays using a single look-ahead signal per flit for both VC and lane activations [28], [29]. Finally, ABNs readily apply to fault tolerance by deactivating only lanes that contain defects, instead of whole channels.

In our experiments, ABNs reduce total power by 15% for uniform random (UR) traffic and up to 45% by average across application benchmarks with comparable performance, compared to single-lane power-gated networks [28], [29], [35], [9]. Compared to state-of-the-art multi-network designs [7], [13], ABNs reduce total power by up to 33% for application benchmarks and increase throughput by 8% for UR traffic, due to the flexibility flits have to switch lanes at each hop, instead of only at injection time. ABNs also provide more fine-grain fault tolerance than multi-network designs.

II. Background and Related Work

Power gating techniques typically disconnect cells from power or ground lines in a coarse- or fine-grain manner using high threshold voltage (low leakage) connector transistors [29], [37], [9]. Such work activates channels or routers in time for flit traversal, or enables detours around inactive resources and guarantees full connectivity [29], [35], [9], [18]. Power gating of input buffers is possible at the granularity of entire buffers [29], [18], VCs [30], or individual entries [24], [32]. Power gating has also been applied to switches and allocators [18], [38]. Other related work scales the voltage or clock frequency of channels and routers [31], [27], [1].

To hide the latency of waking up resources, past work uses look-ahead signals [28], [30]. However, look-ahead signals can cause false activations if they are eligible to activate multiple resources, such as one of multiple VCs [30]. This can occur when a packet A requests an output port that another packet B already has reserved a VC in. In this case, the router may conservatively activate a second VC in anticipation of packet A's arrival, because packet B's completion time is unknown.

To eliminate false VC activations, ABNs adopt drowsy SRAMs [16]. Drowsy SRAMs can be activated in a single cycle and still hold data when drowsy. However, when deactivated, drowsy cells consume more leakage power than power-

This work was supported by the Director, Office of Science, of the U.S. Department of Energy under Contract No. DE- AC02-05CH11231.

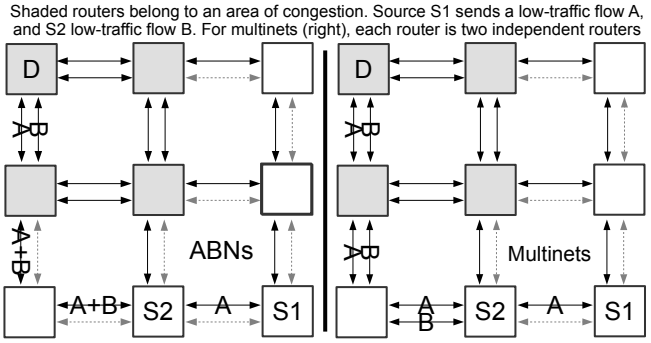

Fig. 1: With ABNs, packets A and B can be placed in the same channel lanes in low-traffic regions.

gated cells, and require more energy to be activated. Drowsy SRAMs were briefly investigated in on-chip networks [10].

Past work, related to ABNs, also adjusts channel bandwidth dynamically but does so by using channels in a bidirectional manner [19], [11], [26]. Further work provides duplicate physical channels between routers instead of VCs, where each channel can reverse direction or be disabled individually [14].

Further related work reduces static power dynamically by using multiple subnetworks [7], [13]. In those designs, traffic sources either inject to an active network or activate a power-gated network using oblivious [7] or adaptive [13] metrics. Compared to multi-network approaches, ABNs provide flits the flexibility to switch lanes in each hop, instead of just at injection time. This reduces the number of lane activations and the number of cycles channel wires are active for, especially under uneven network load. As an example, consider the case where packets need to be placed in separate subnetworks such that congestion is avoided in a high-traffic region. This placement, however, is not optimal for low-traffic regions the packets may traverse, resulting in more channel activations in the low-traffic regions. This is illustrated in Fig. 1.

Packet placement in subnetworks affects performance in addition to energy because packets encountering congestion are unable to utilize another subnetwork's bandwidth. Also, packet injection decisions are inherently less accurate than per-hop decisions because global and accurate knowledge of current and future network state is impossible at injection time.

ABNs also more readily apply to fault tolerance since a defect in a single channel wire shuts down only the affected lane of that channel. In multi-networks, a single fault would disable an entire subnetwork, without the complexity to enable packet detours [36]. Finally, the radix of the network interface at each endpoint increases with the number of subnetworks.

On the other hand, an ABN with two lanes and the same bisection bandwidth as a multi-network design with two subnetworks has half the number of switches but of twice the radix each. Due to the quadratic cost of switches with radix, this results in half the switch area and energy for multi-networks compared to ABNs, and simpler switch allocators.

Mechanisms to dynamically detect silicon defects have been proposed [17]. Faulty channels can be disabled which forces packets to route around faults [34]. Alternatively, channels can include spare wires to replace faulty wires [12], [39], flits

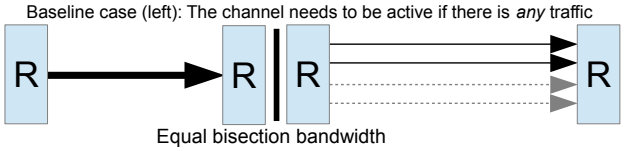

Fig. 2: ABNs divide channels into lanes. Each lane is an independent power gating domain.

can be serialized through the functioning wires [33], [8], or channels can be used in a bidirectional manner [36], [14].

ABNs advance the state of the art by using the low activation delay of drowsy SRAMs to activate VCs without false activations, adjusting the bandwidth in every hop to match traffic demands without the drawbacks of multi-network approaches, and using the same lane and VC activation mechanism for channel fault tolerance in addition to reducing static power.

III. Adaptive Bandwidth Networks

A. Multi-lane Channels

Fig. 2 illustrates how ABNs divide channels into lanes without affecting the bisection bandwidth (which is an orthogonal option). Each lane is an independent power gating domain and is activated according to local traffic demands.

We use channel power gating as described in [29], [10], which disconnects cells from ground using high voltage threshold (low leakage) connector transistors. We use a 65nm technology library and modify the models of [2] to estimate area, power, and wakeup latencies. Using those models we pessimistically estimate 3ns for the channel activation latency ($Lane_{ActLat}$), which matches the upper bound reported by [29], [10] for another 65nm process. Power-gated channel bits consume 0.5% of their leakage power ($Lane_{inact}$), due to the connector transistors. Channel lanes are activated by the router that is driving data on them. Therefore, each lane requires an extra wire to control the high voltage threshold connector transistors and therefore the lane's status. Finally, the activation energy penalty is the equivalent of eight clock cycles of leakage power at 1GHz ($Lane_{ActPen}$). This covers the activation penalty, the propagation of the control bit, as well as the gradual increase and decrease of leakage power.

We restrict packets to using one lane per hop per cycle. In other words, multiple flits of the same packet may not be transmitted in the same cycle using different lanes. This decision was made in the interest of static power and resembles the operation of alternative techniques such as multi-networks [7], [13] which assign packets to a single subnetwork. Without this restriction, a packet could activate all channel lanes, which defeats the purpose of channel lanes. However, this restriction increases serialization latency similar to multi-networks. Still, execution time never increases more than 1% in two-lane ABNs for our application benchmarks.

B. Router Datapath

For input buffers, we model drowsy SRAM cells [10]. Each VC can be activated independently from other VCs and in a single cycle (VC_{ActLat}). When inactive, drowsy SRAM cells consume 15% of their active leakage power (VC_{inact}). We

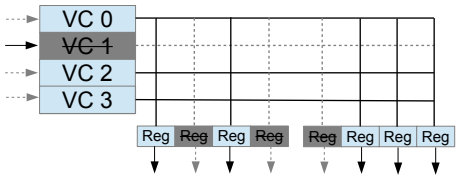

Fig. 3: One input and two outputs are shown. The switch connects to each VC and each output lane. These connections are power gated according to the state of VCs and output lanes.

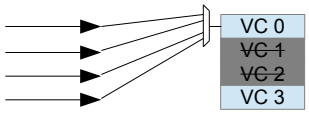

Fig. 4: A multiplexer for each VC is required because flits from any lane may be destined to any VC. Look-ahead signals minimize the timing impact of these multiplexers.

pessimistically estimate the energy penalty for activating a VC to equal sixteen cycles of leakage power at 1GHz (VC_{ActPen}). This covers both the activation energy and the leakage power during the cycle that a VC is activating or deactivating.

Even though channels may deliver one flit per lane per cycle, those flits will be destined to different VCs since packets are restricted to send only one of their flits per cycle. Therefore, VCs can be placed in separate SRAMs to make managing VCs as independent power gating domains more straightforward.

To avoid making the input side of router switches a bottleneck, router switches need to connect to every input VC with separate switch lanes. At the output side, switches connect to each lane of each output channel. Therefore, switches have $(InputPorts \times VCs)$ inputs and $(OutputPorts \times ChannelLanes)$ outputs. This way, routers can transmit a flit to each lane of each output port in every cycle, and each input VC can transmit a flit independently of other input VCs. In a single-lane network with the same bisection bandwidth switches have $InputPorts$ inputs and $OutputPorts$ outputs, but the width of each of these switch input and output ports equals the width of $ChannelLanes$ switch ports in ABN switches. Therefore, the output side of ABN switches has the same number of bits as a single-lane network with the same bisection bandwidth, whereas the input side is no wider as long as $ChannelLanes \leq VCs$.

In addition to VCs and channel lanes, ABNs also apply power gating to switches [18]. At the input side, the switch connection to a VC is only active when the VC is active. At the output side, switch lanes are only active when the corresponding channel lanes are active. This is shown in Fig. 3.

Because flits in any channel lane may be destined to any input VC, input buffers require a multiplexer for each input VC, as shown in Fig. 4. To mitigate the timing overhead to the last pipeline stage of the channel, the multiplexer's control inputs for each input VC arrive one cycle before the flit, using the look-ahead signal for that flit (explained in Section III-D).

Still, with a large number of channel lanes, this multiplexer may pose a noticeable timing overhead even with preset control inputs. We can simplify this multiplexer by mapping a subset of VCs to each lane such that choosing an output VC restricts the allowed output channel lanes. While this will result in more channel lane activations due to choice restriction, it also simplifies switch allocation by reducing the possible input VC–output port combinations. Finally, if the number of VCs equals the number of lanes, reserving an output VC essentially also reserves a lane in the output channel. Therefore, VC allocation is no longer required. We call this option *ABN simple* and quantify its efficiency in Section V. Since packets choose a VC before lanes, ABNs use VCs similarly to networks without lanes for deadlock avoidance.

C. ABN Complexity

ABNs increase switch allocator complexity because multiple grants may be generated for each output (one for each lane), and each input VC may be granted independently of other VCs of the same input. With ABNs, switch allocators perform an $(InputPorts \times VCs) \times (OutputPorts \times ChannelLanes)$ allocation, where the same input can receive multiple grants to different VCs in the same cycle. However, in the typical case where $ChannelLanes \leq VCs$, the switch allocator is no more complex than the VC allocator, which performs an $(InputPorts \times VCs) \times (OutputPorts \times VCs)$ allocation where the same input can also receive multiple grants to different output VCs. Past work reports that in a typical mesh with 2 VCs, extending the radix of the switch allocator to become equivalent to that of the VC allocator extends the switch allocator's minimum timing path by 10% for separable allocators [4]. Even in a high radix flattened butterfly (FBFly) topology, the switch allocator's path is only extended by 15% [4]. However, given than VC and switch allocation are typically performed in separate pipeline stages and the switch allocator is no more complex that the VC allocator if $ChannelLanes \leq VCs$, this timing overhead is unlikely to extend the router critical path.

In addition, increasing the switch allocator radix in a mesh with 2 VCs to match that of the VC allocator would increase area by approximately 30% and power by 35% for separable allocators [4]. However, the VC allocator occupies approximately 5000 μm^2 and consumes 2 to 10 mW, both of which are very small percentages of the router [4], [21]. For example, the Intel Teraflop chip consumes 7% of the network's power for allocation and *all* other router logic [21].

As stated in Section III-B, ABNs do not increase router switch radix compared to a single-lane network with the same bisection bandwidth as long as $ChannelLanes \leq VCs$. Moreover, ABNs have the same overhead for power gating compared to past work because ABNs use the same models of power-gated transistors [18], [29], [7], [13]. Finally, past work also uses look-ahead signals for wakeup. ABNs reduce the overhead of look-ahead signals by using drowsy SRAMs which allows a single look-ahead signal per flit.

D. Router Pipeline and Control

The router pipeline is illustrated in Fig. 5. For each flit receiving a switch allocator grant, a look-ahead signal is sent one cycle before the flit enters the output channel that:

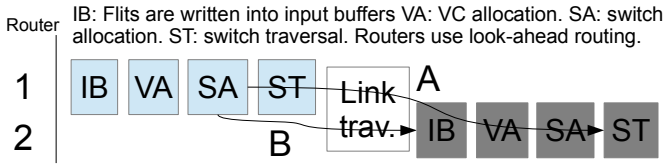

Fig. 5: The pipeline for two consecutive routers. A look-ahead signal is created for each flit winning switch allocation.

- *A*: Alerts the downstream router that a flit will be arriving for a certain output. The router can then activate more switch and output channel lanes.
- *B*: Alerts the input buffer of the downstream router of the VC the flit will be arriving for. If that VC is inactive, it will be activated. The multiplexer at the input side of that input VC will be set at the beginning of the next cycle.

Look-ahead signals are generated for each flit in a packet. That is necessary because switch allocation may delay body flits long enough for resources to power down. Each look-ahead signal contains the input VC and output port the flit will use in the downstream router, obtained in the upstream router using look-ahead routing [28]. Therefore, look-ahead signals are $\log_2 Outputports + \log_2 InputVCs + ValidBit$ bits wide. One look-ahead signal is required per channel lane.

Channel and switch lanes are activated to match the number of flits destined to each output port. Routers maintain a counter per output port. Look-ahead signals increment the counter for the output port they are requesting. Flits receiving a grant in switch allocation decrement the counter. Routers activate as many channel lanes and corresponding switch output lanes as the counter's value, but with a delay of $Lane_{ActWait}$ cycles. Specifically, lane X is activated if the counter's value for that output has been at least X continuously for the last $Lane_{ActWait}$ cycles. This ensures that ABNs do not over-react to short-lived congestion. ABNs can use more complex policies that consider past or neighboring state [1], [14], [19], [11], but this is left as future work. Routers deactivate lanes if they are inactive for $Lane_{DeactWait}$ cycles.

In our four-stage routers, look-ahead signals hide three cycles of activation latency for switch and channel lanes. That is because look-ahead signals arrive one cycle in advance of their corresponding flit, and flits need to go through VA and SA before traversing the switch. In our 65nm technology library and power gating models [29], [10], this is enough to hide the lane activation delay in full. In addition, since switch traversal precedes link traversal by a cycle, channel lanes need only be activated one clock cycle after the corresponding switch lanes.

Because of the proactive nature of lane activation, false lane activations are possible. A false lane activation is an activation without a subsequent flit traversal. Consider the case where packets A and B arrive for the same output port, but A is stalled waiting for credits until after B departs the router. In this example, two lanes are activated to guarantee that A will not stall waiting for a lane if a credit arrives. However, one active lane would suffice. We quantify the frequency of false lane activations in Section V-B.

Look-ahead signals arrive only one clock cycle in advance of the corresponding flits. One cycle suffices to hide the single-cycle activation delay of drowsy SRAMs. Power gated (non-drowsy) SRAMs which require more cycles (such as three cycles in [30]) would require an additional look-ahead signal before VC allocation in the upstream router's pipeline. The single-cycle activation delay enables the use of the upstream router's VC allocator to activate downstream VCs. This eliminates false VC activations [30] because only VCs that flits are actually assigned to are activated. However, downstream VCs are not activated until after the flit wins switch allocation. In addition, to reduce unnecessary VC activations, VC allocators prioritize active output VCs. Input VCs are deactivated if empty and have either been idle for $VC_{DeactWait}$ cycles or all channel lanes to that input are inactive.

E. Silicon Defect Tolerance

While the primary purpose and novelty of ABNs lie in static power consumption, ABNs also readily apply to isolating channel defects with only the additional cost for detecting faults. With ABNs, faulty VCs as well as channel or switch lanes can simply be disabled, but the remaining resources are still usable. Therefore, a single fault does not necessitate disabling entire input ports or channels and resort to extra complexity to enable detours [25], [34]. If the fault probability for a single channel bit is P (ranging from 0 to 1), channel width W, number of lanes L, the probability for a channel to fully fail is:

$$(\frac{W}{L} \times P)^L$$

$L = 1$ represents the baseline single-lane network assuming channel bit failures are independent events. Lanes are considered failed if they contain at least one faulty bit.

As the number of lanes increases, the probability that all lanes contain a fault decreases. Therefore, ABNs are more likely to maintain network connectivity with an equal number of faults compared to the baseline single-lane network. Multi-network designs will fail if a single channel in any subnetwork fails, without extra VCs and complexity to enable detours [36], [25], [34]. That is because flits already injected to the faulty subnetwork cannot switch subnetworks, and there is propagation delay to alert all traffic sources of the fault.

IV. Methodology

For our evaluations, we use a modified version of Booksim [22]. We present results for synthetic traffic and PARSEC benchmarks collected with Netrace traces [5], [20], which respect packet dependencies and therefore reflect the impact of the network to application execution time. For synthetic traffic we use a read-reply communication protocol. The traffic pattern decides the destination of read and write requests. Each request generates a reply. Read requests and write replies are 128 bits. Write requests and read replies are 640 bits. For our synthetic traffic we vary the injection rate of request packets.

We use the Netrace traces provided in the project's website. These traces were collected for a 64-core cache-coherent chip multiprocessor (CMP) with in-order ALPHA cores. L1 data and instruction caches are 32KBs each, 4-way set associative, and use MESI cache coherency. L2 caches are fully shared S-NUCA with 64 banks and 16MB, eight-way set associativity and 8-cycle bank access time. Finally, the memory has a 150-cycle access time and 8 on-chip memory controllers. We simulate 200,000 packets of the parallel region of seven PARSEC benchmarks using their medium size input sets. Longer simulations produce comparable results.

We compare the following networks:

- *Baseline network without power gating (baseline)*: This is a single network without power gating.
- *Single-lane power gating network (single-lane)*: This network represents the state of the art in single-network power gating [28], [30]. In order to isolate the gains from channel and switch lanes, in this network we still use drowsy SRAM cells [30], [10].
- *Flexible adaptive bandwidth network (ABN flexible)*: We keep bisection bandwidth constant compared to other networks. Therefore, with two lanes, flits are half the width compared to the networks above, and each input port has twice the VCs. Flits can choose any output VC.
- *Simple adaptive bandwidth network (ABN simple)*: Same as above, but we map lanes to only allow delivery to a subset of VCs. With four VCs and two lanes, each lane delivers to one request VC and one reply VC.
- *Multi-network designs (multinets)*: This represents the state of the art in static power reduction [7], [13]. Sources inject flits to the first subnetwork unless the count of available buffer space in all input buffers of the injection router is less than half of total buffer size [13]. In that case, sources consider the next subnetwork, and so on. If all subnetworks are congested, sources choose one at random. Bisection bandwidth and flit width equal ABNs for an equal number of lanes and subnetworks.

We use an 8×8 2D mesh with dimension-order routing (DOR), 2mm channels, and the router pipeline of Fig. 5. The baseline network has 128-bit channels and two VCs per input, chosen as a good trade-off between performance and cost. VCs are equally divided among requests and replies. To keep total buffer size constant, we increase the number of VCs for networks with more than one lane, because such networks have narrower flits. Making VCs deeper instead typically does not justify the increased cost as long as the credit round-trip delay is covered. Therefore, two-lane ABNs have four VCs per input. The increased number of VCs in ABNs may affect router clock frequency if the VC allocator is in the critical path and the network needs to be clocked at maximum frequency. In that case, ABNs can use fewer VCs which will sacrifice performance, but also reduce cost. Increasing the radix of VC allocators also increases their power and area costs [4], but all router allocation and control logic is just 7% of network power in the Intel Teraflop processor [21]. Multinets also have more VCs in total but the VCs are distributed among subnetworks.

TABLE I: Network and model parameters.

Parameter	Value
$Lane_{ActLat}$	3 cycles
VC_{ActLat}	1 cycle
$Lane_{inact}$	0.5% of full leakage
VC_{inact}	15% of full leakage
$Lane_{ActPen}$	8 cycles worth of leakage
VC_{ActPen}	16 cycles worth of leakage
$Lane_{ActWait}$	15 cycles
$Lane_{DeactWait}$	3 cycles
$VC_{DeactWait}$	6 cycles
$Area_{Overhead}$	7%

For cost estimation, we use a 65nm technology library and a 1GHz clock frequency. We modify the area and power models of [2] to include power gating [29], [10], [18], drowsy SRAMs, and additional overhead such as the extra wires for the look-ahead signals. From these models, we pessimistically estimate the power gating area overhead to be 7% for buffers, switches, and channels ($Area_{Over}$). Drowsy SRAMs can be activated in a single cycle [16], whereas for channel and switch lanes the activation latency is 3ns [29]. Since ABNs hide three cycles of lane activation delay, ABNs fully hide the lane activation delay in this technology process. We derive the parameters shown in TABLE I based on our models and preliminary evaluations. While $Lane_{ActWait}$, $Lane_{DeactWait}$, and $VC_{DeactWait}$ depend on the probability of flits reusing lanes or VCs, which depends on the traffic pattern, we choose one set of numbers for all traffic patterns for simplicity.

We report static power which is predominantly composed of leakage power but also includes the power to toggle the capacitance of the clock input of cells and SRAMs. Static power also includes energy penalties from activating resources. Dynamic power includes look-ahead and wakeup signals.

V. Evaluation

A. Application Traffic

For our PARSEC simulations, we first replay the traces and respect packet dependencies. This produces an approximately 0.2% flit injection rate across benchmarks. We call this the low load testcase, which also evaluates application execution time. We then relax packet dependencies and increase the average flit injection rate to 2% (medium load) and 3.5% (high load) across benchmarks. Injection rates higher than 3.5% are susceptible to causing tree saturation in some benchmarks due to load imbalance. Medium and high loads are not used to measure execution time, but rather to load the network in a manner closer to an application with higher loads than our PARSEC benchmarks. The PARSEC benchmarks we choose exhibit a variety of communication patterns. Specifically, in blackscholes and fluidanimate cores transmit equally to each of two nearby cores, in canneal traffic resembles UR by average, and in the rest of the benchmarks cores transmit different amounts of data to a subset of nearby and distant cores [3].

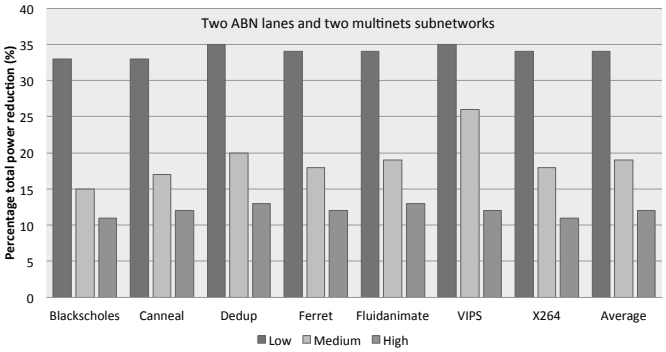

Fig. 6: Total power reduction of ABN simple with two lanes compared to multinets with two subnetworks.

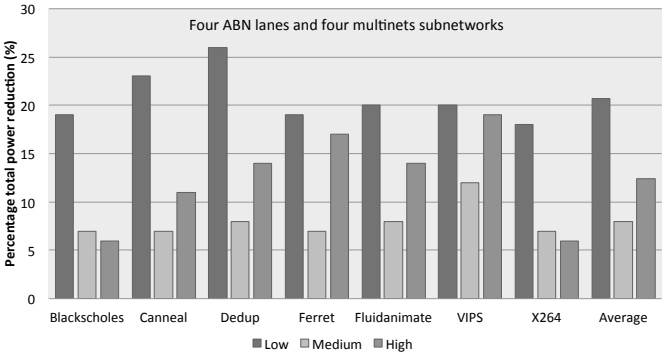

Fig. 7: Total power reduction of ABN simple with four lanes compared to multinets with four subnetworks.

Fig. 6 presents the percentage of total power reduction of ABN simple with two lanes compared to multinets with two subnetworks. We observe an approximately 33% average total power reduction for ABN simple for low loads. Application traffic is often bursty and produces unbalanced loads [35], [9], [19], [3]. Hotspots exacerbate the impact of the lack of flexibility flits have in choosing subnetworks after injection, as shown in Fig. 1. Bursty traffic also creates temporary congestion in routers which causes flits injected to that router to be sent to another subnetwork. Those flits may not switch subnetworks after the injection router, and therefore may not share active resources with other low traffic. This results in 43% to 55% more active channel and switch lane cycles by average across benchmarks for multinets compared to ABN simple. Compared to single-lane power-gated networks, ABN simple reduces total power by an average of 45%. Both ABNs and multinets cause an average slowdown of just 0.95%, with the maximum being 1.05% in the case of blackscholes. Static power reductions decrease with an increase in injection rate due to fewer power gating opportunities with more traffic.

B. Synthetic Traffic

We use synthetic traffic to gain more insight and to evaluate the worst-case traffic for ABNs since UR traffic is perfectly load balanced (load imbalance favors ABNs compared to multinets). Results are shown in Fig. 8. ABN simple saturates at an 8% higher injection rate than multinets because in multinets flits cannot escape transient congestion encountered in a subnetwork even with UR traffic, and perfect injection decisions are unrealistic. This also causes multinets to have a 34% higher average latency close to saturation (40% request packet injection rate). Baseline and single-lane each provide an 8% lower throughput than ABN simple because ABN simple has twice as many VCs. However, due to serialization latency, baseline and single-lane have a 10% lower average zero-load packet latency compared to ABNs and multinets.

ABN flexible and multinets have comparable power consumption across injection rates. They each have a 15% lower total power consumption than the single-lane network, and 24% compared to the baseline. In addition, multinets have 7% lower dynamic power compared to each other network, because multinets use twice the number of switches of half the radix, which therefore incur one quarter of the cost each. However, because flits pick a subnetwork at injection time with imperfect knowledge and are not able to switch subnetworks later, multinets have a 9% higher static power compared to ABN flexible due to a 36% higher channel static power and 48% higher buffer static power. This offsets the gains in dynamic power for multinets. ABN simple has no false lane activations because once a packet chooses an output VC, all flits have to use the lane that output VC is assigned to. ABN flexible has a 19% lower activation power overhead than multinets because it experiences 17% fewer activations, since it is free to make maximum use of already active lanes.

In the baseline single-lane network, channel static power is 51% of overall static power, while buffer static power is 20% and switch 23%. Static power is 62% of overall power under a 2% request packet injection rate. ABN flexible reduces channel static power including activation penalties by 53%, buffer static power by 73%, and switch static power by 56%.

Comparing ABN simple and flexible, ABN simple saturates at a 21% higher injection rate. This is because in ABN flexible flits may request any output lane. This increases the allocation problem and intensifies the inefficiencies of our separable single-iteration VC and switch allocators.

In summary, for UR traffic multinets have power consumption comparable to ABNs, but lower performance. UR traffic is the worst-case for ABN because traffic patterns with imbalance in time or space favor ABNs due to the flexibility in ABNs in choosing lanes in each hop. As discussed in Section V-A, ABNs lower the total power consumption by up to 33% compared to multinets for application traffic.

C. Increasing the Number of Lanes

In this Section, we divide the same bisection bandwidth to four lanes for ABNs and four subnetworks for multinets. As shown in Fig. 7, ABN simple reduces total power by 21% under low loads, 8% under medium loads, and 13% under high loads, compared to multinets. Power reductions are smaller compared to Section V-A because further subdividing into more subnetworks makes router switches in multinets more energy efficient due to their quadratic cost with radix. In addition, power gains under high loads are larger for ABN simple because the lack of flexibility of flits in multinets becomes

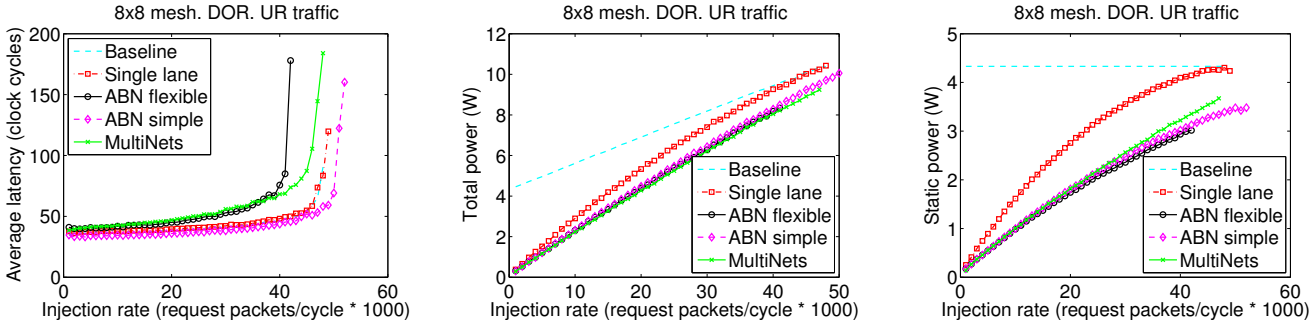

Fig. 8: ABNs have two lanes and multinets consist of two subnetworks. Static power includes activation penalty power.

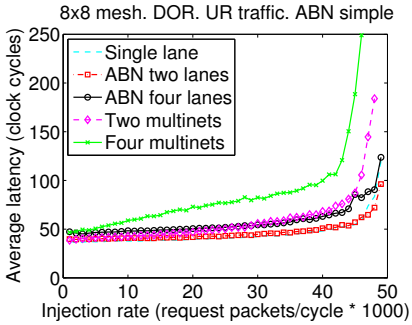

Fig. 9: Scaling of ABNs and multinets with UR traffic.

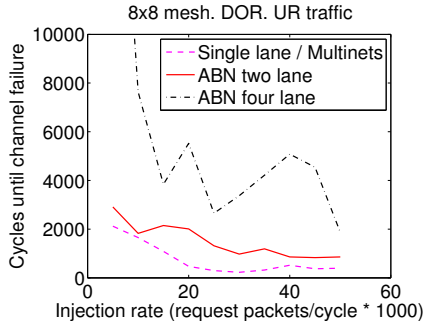

Fig. 10: Time until a packet chooses an output port leading to a failed channel, for single-lane, multinets, and ABN flexible.

more pronounced compared to having two subnetworks. ABNs also have a marginal (1%) benefit in execution time under low loads in five benchmarks compared to multinets.

Fig. 9 presents latency under UR traffic. As shown, multinets with four subnetworks saturate at a 5% lower injection rate and have a 45% higher average latency compared to ABN simple with four lanes. This is because the effect of flits being unable to switch to idling subnetworks is more pronounced with four subnetworks compared to two. Finally, ABNs with four lanes and multinets with four subnetworks have a 19% higher average zero-load latency than ABNs with two lanes.

D. Design Space Exploration

To motivate use of drowsy SRAM cells, we compare ABN flexible with drowsy SRAMs and power-gated (non-drowsy) SRAMs [30], [16] under UR traffic. At low loads, we observe an average 43% false VC activations with power-gated SRAMs (there are no false activations with drowsy SRAMs), and 15% more active VC cycles for power-gated SRAMs. However, due to the different energy overheads, non-drowsy SRAMs consume 22% less activation power, but 32% more static power due to the extra active cycles, resulting in 11% higher static power (including activation overhead) overall. At high loads, static power without activation overheads is comparable, but non-drowsy SRAMs incur 12% more activation power due to the 35% false VC activations. While these numbers depend on the traffic pattern, they show the benefits of drowsy SRAMs, while also simplifying the router pipeline since one wakeup signal suffices for both VCs and channel lanes.

We also evaluate a baseline network with four VCs and ABNs with eight VCs to test the sensitivity of our results to the number of VCs. With UR traffic, ABN simple has comparable (1% higher) performance than multinets with separable allocators, and 3% higher with wavefront allocators. However, with realistic application traffic, there is insignificant impact because the network is not close to saturation.

Finally, our results depend on the relative contributions of channels and switches. Topologies with higher-radix switches, such as the FBFly, favor multinets because router switches have a higher radix and therefore the benefit of reducing their radix in half in multinets is relatively larger. In contrast, topologies with longer channels, such as a mesh with longer channels than our mesh, favor ABNs because ABNs reduce channel leakage power compared to multinets.

E. Silicon Defect Tolerance

To measure the improved resiliency of ABNs, we simulate UR traffic and assign a 5×10^{-4} probability that any one channel bit line will fail in each cycle. We assume that channels have two spare bit lines [12], [39]. We report the time that a packet chooses an output port that leads to a failed channel (without detours). In the case of ABN flexible, this means all lanes have failed. Essentially, this is the time period that the network is no longer able to function correctly. Multinets have comparable probability as the single-lane network because when a channel in any subnetwork fails, flits already injected may not switch subnetworks to avoid the faulty channel, and there is propagation delay to alert sources.

Fig. 10 shows the increased resiliency of ABNs (2× higher for two lanes compared to single-lane and multinets). The variance in our results, especially for four lanes, stems from the many random choices in each experiment. ABN simple performs in between ABN flexible and multinets.

VI. Discussion

Our results illustrate the advantage of allowing flits to switch lanes in ABNs with local per-hop decisions to accommodate regions with different traffic conditions, compared to multinets where the decision is made once at injection time where perfect knowledge of current and future global state is impossible. This translates to performance and power benefits, especially with unbalanced traffic. However, dividing a single network into subnetworks with multinets makes router switches more energy and area efficient. Comparing ABN simple and flexible, ABN simple provides better performance but ABN flexible provides slightly lower static power.

Routers with shallow pipelines would activate VCs in a similar manner because of the single-cycle activation delay of drowsy SRAMs. However, routers with shallow pipelines or higher clock frequencies (which increase wakeup latencies in terms of clock cycles) may not be able to fully hide channel and switch lane wakeup latencies with look-ahead signals. Future technologies may also affect wakeup latencies. For example, past work reports 5.1ns for a 32nm technology library [13] and 4ns for 45nm [9]. If wakeup latencies cannot be fully hidden, networks can either use predictors similar to non-drowsy SRAMs [30], or leave some lanes constantly activated. In addition, ABNs have similar power gating granularity and require similar power distribution networks or power gating transistors than multinets or other past work [24], [32], [13].

Look-ahead signals do cause extra bits to be transported, but they are only a small fraction of the flit width and enable significant savings in leakage power which will be critical in future technologies [6], [23], [9]. Finally, upcoming technologies such as FINFETs may reduce the contribution of leakage power, but that is still projected to increase and remain important in future technologies [6], [23].

VII. Conclusion

This paper proposes ABNs. ABNs divide channels and switches into lanes each of which can be activated individually to match traffic demands. Unlike power-gating approaches with multiple subnetworks, flits are free to choose a different lane at each hop instead of committing to a set of lanes at injection time. At the input buffer side, ABNs take advantage of drowsy SRAM cells to activate VCs individually without the possibility of false activations. ABNs also readily apply to silicon defect tolerance. For application traffic, ABNs reduce total power consumption by up to 33% compared to multi-network designs and up to 45% compared to single-lane networks, with comparable or superior performance.

References

[1] A. Ansari *et al.*, "Tangle: Route-oriented dynamic voltage minimization for variation-afflicted, energy-efficient on-chip networks," ser. HPCA, 2014.

[2] J. Balfour and W. J. Dally, "Design tradeoffs for tiled CMP on-chip networks," ser. ICS, 2006.

[3] N. Barrow-Williams, C. Fensch, and S. Moore, "A communication characterisation of Splash-2 and Parsec," ser. IISWC, 2009.

[4] D. U. Becker and W. J. Dally, "Allocator implementations for network-on-chip routers," ser. SC, 2009.

[5] C. Bienia, "Benchmarking modern multiprocessors," Ph.D. dissertation, Princeton University, January 2011.

[6] S. Borkar, "How to stop interconnects from hindering the future of computing," ser. OIC, 2013.

[7] J. Camacho and J. Flich, "HPC-mesh: A homogeneous parallel concentrated mesh for fault-tolerance and energy savings," ser. ANCS, 2011.

[8] C. Chen, Y. Lu, and S. D. Cotofana, "A novel flit serialization strategy to utilize partially faulty links in networks-on-chip," ser. NOCS, 2012.

[9] L. Chen and T. M. Pinkston, "NoRD: Node-router decoupling for effective power-gating of on-chip routers," ser. MICRO, 2012.

[10] X. Chen and L.-S. Peh, "Leakage power modeling and optimization in interconnection networks," ser. ISLPED, 2003.

[11] M. H. Cho *et al.*, "Oblivious routing in on-chip bandwidth-adaptive networks," ser. PACT, 2009.

[12] W. J. Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks," ser. DAC, 2001.

[13] R. Das *et al.*, "Catnap: Energy proportional multiple network-on-chip," ser. ISCA, 2013.

[14] D. DiTomaso, A. Kodi, and A. Louri, "QORE: A fault tolerant network-on-chip architecture with power-efficient quad-function channel (QFC) buffers," ser. HPCA, 2014.

[15] H. Esmaeilzadeh *et al.*, "Dark silicon and the end of multicore scaling," ser. ISCA, 2011.

[16] K. Flautner *et al.*, "Drowsy caches: simple techniques for reducing leakage power," ser. ISCA, 2002.

[17] A.-A. Ghofrani *et al.*, "Comprehensive online defect diagnosis in on-chip networks," ser. VTS, 2012.

[18] K. C. Hale, B. Grot, and S. W. Keckler, "Segment gating for static energy reduction in networks-on-chip," ser. NoCArc, 2009.

[19] R. Hesse, J. Nicholls, and N. E. Jerger, "Fine-grained bandwidth adaptivity in networks-on-chip using bidirectional channels," ser. NOCS, 2012.

[20] J. Hestness, B. Grot, and S. W. Keckler, "Netrace: dependency-driven trace-based network-on-chip simulation," ser. NoCArc, 2010.

[21] Y. Hoskote *et al.*, "A 5-GHz mesh interconnect for a teraflops processor," *Micro, IEEE*, vol. 27, no. 5, 2007.

[22] N. Jiang *et al.*, "A detailed and flexible cycle-accurate network-on-chip simulator," ser. ISPASS, 2013.

[23] H. Kaul *et al.*, "Near-threshold voltage (NTV) design; opportunities and challenges," ser. DAC, 2012.

[24] G. Kim, J. Kim, and S. Yoo, "Flexibuffer: reducing leakage power in on-chip network routers," ser. DAC, 2011.

[25] M. Koibuchi *et al.*, "A lightweight fault-tolerant mechanism for network-on-chip," ser. NOCS, 2008.

[26] Y.-C. Lan *et al.*, "BiNoC: A bidirectional noc architecture with dynamic self-reconfigurable channel," ser. NOCS, 2009.

[27] S. E. Lee and N. Bagherzadeh, "A variable frequency link for a power-aware network-on-chip (NoC)," *Integr. VLSI J.*, vol. 42, no. 4, 2009.

[28] H. Matsutani *et al.*, "Run-time power gating of on-chip routers using look-ahead routing," ser. ASPDAC, 2008.

[29] ——, "Ultra fine-grained run-time power gating of on-chip routers for CMPs," ser. NOCS, 2010.

[30] ——, "Adding slow-silent virtual channels for low-power on-chip networks," ser. NOCS, 2008.

[31] A. K. Mishra *et al.*, "RAFT: A router architecture with frequency tuning for on-chip networks," *Journal on PDC*, vol. 71, no. 5, 2011.

[32] C. Nicopoulos *et al.*, "Variation-aware low-power buffer design," ser. ACSSC, 2007.

[33] M. Palesi, S. Kumar, and V. Catania, "Leveraging partially faulty links usage for enhancing yield and performance in networks-on-chip," *IEEE Transactions on CAD*, vol. 29, no. 3, 2010.

[34] S. Rodrigo *et al.*, "Addressing manufacturing challenges with cost-efficient fault tolerant routing," ser. NOCS, 2010.

[35] V. Soteriou and L.-S. Peh, "Exploring the design space of self-regulating power-aware on/off interconnection networks," *IEEE Transactions on PDS*, vol. 18, no. 3, 2007.

[36] W.-C. Tsai *et al.*, "A fault-tolerant NoC scheme using bidirectional channel," ser. DAC, 2011.

[37] K. Usami and N. Ohkubo, "A design approach for fine-grained run-time power gating using locally extracted sleep signals," ser. ICCD, 2006.

[38] H. Wang, L.-S. Peh, and S. Malik, "Power-driven design of router microarchitectures in on-chip networks," ser. MICRO, 2003.

[39] Q. Yu and P. Ampadu, "Transient and permanent error co-management method for reliable networks-on-chip," ser. NOCS, 2010.