Adaptive Flow Control for Robust Performance and Energy

Syed Ali Raza Jafri*, Yu-Ju Hong*, Mithuna Thottethodi, and T. N. Vijaykumar (* both primary student authors)
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN, USA
Email: {sjafri,yujuhong,mithuna,vijay}@purdue.edu

***Abstract*—As chip multiprocessors scale the number of on-chip cores, the superior scalability of multihop networks compared to buses and crossbars makes multihop networks the choice interconnection strategy. However, a significant part of the networks' energy is consumed in the buffers used to handle link contention via backpressured routing. Recent work proposes to apply well-known backpressureless routing techniques, which eliminate buffers, and hence buffer power (static and dynamic), at the cost of some misrouting/dropping upon link contention (misrouted/dropped flits are eventually recovered/retransmitted). At low loads, misrouting (dropping) is rare and hence backpressureless routing performs well. Unfortunately, backpressureless routers incur significant misrouting/dropping under high loads and saturate at lower throughputs than backpressured networks, resulting in poorer performance and energy. We make the key observation that because load varies significantly across applications, backpressureless and backpressured networks are not robust in performance-energy across the spectrum of high and low loads. That is, at high loads backpressureless networks suffer considerable performance and energy disadvantage compared to backpressured networks; and the energy disadvantage reverses at low loads. To address this robustness issue, we propose a novel adaptive flow control (AFC) router which dynamically adapts between backpressured and backpressureless flow control. AFC employs three novel mechanisms, namely *local contention thresholds*, *gossip-induced mode-switch*, and *lazy VC allocation*. The first mechanism maximizes performance (and minimizes energy) in the common case, and the second mechanism ensures correctness in corner cases. The third mechanism exploits flit-by-flit routing in AFC's backpressured mode to simplify VC allocation and reduces the buffer requirements by a factor of two in AFC's backpressured mode. Simulations using commercial workloads and SPLASH-2 confirm AFC's robustness by showing that AFC achieves performance and energy that are closer to that of the *better* of backpressured and backpressureless networks.**

***Keywords*-Interconnection networks; flow control;**

I. Introduction

As the microprocessor industry moves towards 16+ cores per chip, the adoption of multi-hop networks as the interconnection fabric is inevitable because neither buses nor crossbars scale adequately. Ideally, such multi-hop networks must provide (1) low-latency because all L1 cache misses are exposed to network latencies, and (2) high bandwidth to support the larger number of cores.

Traditionally networks have been designed to handle link contention by using input buffering (to stall all but one of the contending flits) and backpressure (to prevent stalled/buffered flits from being overwritten by other incoming flits). Unfortunately, buffers consume a significant part of on-chip network energy (e.g., 30-40%). Circuit techniques such as buffer resizing or fine-grained gating incur implementation severe complexities (see Section III-A). Accordingly, recent work addresses the problem of buffer dynamic energy by employing buffer bypass techniques [1], [2]. Approaches that target both static and dynamic buffer energy by using backpressureless routing and eliminating the use of buffers have also been proposed [3], [4]. Such backpressureless routers[1] handle link contention either by using well-known deflection/hot-potato routing [6] or by dropping packets/flits [4] upon contention. The first variant – deflection routing – ensures that all incoming flits leave on *some* outgoing link, even if it is a misroute (from which the flit will eventually recover). The second variant employs the strategy of dropping all but one of the contending flits instead of misrouting them. Such dropped flits are later retransmitted. At low network loads, backpressureless routing is efficient because link contention, and hence misrouting (or dropping of flits), is rare.

Unfortunately, backpressureless routers suffer from poor performance and energy at higher loads because the misrouting/dropping caused by link contention leads to increased link utilization, which creates a positive feedback cycle because increased link utilization further increases link contention. Consequently, backpressureless networks saturate at lower throughputs than backpressured networks.

Moscibroda *et al.*, in their case for backpressureless routers, have argued that the network load offered by typical workloads is indeed low [3]. However, their measurements were conducted on multiprogrammed sequential workloads. We show that the network load is not always low for commercial benchmarks running on multi-threaded cores. We make the key observation that load varies significantly across applications and, to a lesser extent, over time and space within the network (e.g., hotspots, program phases). A consequence of this observation is that backpressureless and backpressured networks are not robust in performance-

[1]Note, such routers have traditionally been referred to as bufferless routers. However, the term "bufferless" router is not precise because the lack of buffers is neither necessary (hot potato routers may use buffers to temporarily avoid misroutes [5], [3]) nor sufficient (traditional backpressured routers can avoid the use of buffers, at the cost of performance). The true distinguishing feature of such routers is the absence of backpressure. Consequently, we use the term "backpressureless" for such routers.

energy across the spectrum of high and low loads – i.e., at high loads backpressureless networks suffer considerable performance and energy disadvantage compared to backpressured networks; and the energy disadvantage reverses at low loads. Performance-energy robustness is important especially for multicores which target general-purpose computing where often applications exert vastly diverse loads on the network.

To address this robustness issue, we propose Adaptive Flow Control (AFC) – which dynamically adapts between backpressureless and backpressured flow control – to approach the best of both worlds, thereby improving performance-energy robustness. Individual AFC routers dynamically switch between backpressured and backpressureless modes of operation. AFC does not use global, network-wide mode switching because a distributed, synchronized operation to ensure that all routers are switched, while applications are running, may be impractical. As such, at any instant of time some routers may be in backpressureless mode and the others in backpressured mode; and, a given router may switch modes over time.

There are two key challenges for AFC, one in the common case of unvarying (high or low) load and the other in corner cases of varying load. In the first case, the routers should be in the appropriate mode of operation to achieve good performance-energy. To avoid per-application tuning of the modes, we propose *local contention thresholds*, our first novel mechanism, derived statically at design-time based solely on network loading and independent of other application characteristics. Routers where the locally-measured link contention exceeds the thresholds switch to backpressured mode, and vice versa.

Second, AFC must ensure the correctness in the corner cases of flow control interactions between adjacent routers in different modes of operation (e.g., in transient conditions when the load is changing or under high spatial variation in load). Depending on the direction of communication, there is an easy case and a difficult case. In the easy case of a backpressured-mode router communicating with a backpressureless-mode router, no additional safeguards are needed because backpressureless-mode routers are prepared to accept flits on every cycle. However, communication from backpressureless-mode routers to backpressured-mode routers is difficult because backpressured-mode routers, by definition, cannot always accept flits that a backpressureless-mode router may send. To address this concern, we propose *gossip-induced mode switches*, our second novel mechanism, wherein backpressureless-mode routers are forced to switch to backpressured-mode because of contention at a neighboring (backpressured) router even though the backpressureless router may not observe local contention. AFC infers contention at the neighboring nodes from locally-visible credit-backflows that are used for backpressured flow control.

Finally, because AFC (like backpressureless routing) may route flits of a single packet independently, AFC incurs the area and dynamic energy overhead (compared to a backpressured router) of wider flits requiring wider buffers, crossbars and links to carry per-flit routing information. At low loads, the energy overhead is more than compensated by AFC's ability to eliminate both static and dynamic buffer energy due to its backpressureless mode where *all* its buffers are power-gated [7]. Such coarse-grained gating does not incur the implementation complexities of fine-grained gating mentioned above. At high loads, where dynamic energy dominates, naively using traditional backpressured mode incurs the full energy overhead. Instead, AFC compensates for the area and energy overhead of the wider flits by leveraging flit-by-flit routing to optimize the backpressured mode. Traditional virtual channel (VC) flow control allocates and releases VC buffers at per-packet granularity to ensure that flits of a packet are always routed together because individual flits do not contain routing information. In contrast, because a packet's flit may be routed independently by a backpressureless-mode router to a backpressured-mode router, AFC must support flit-by-flit routing even in its backpressured mode. Because the purpose of VCs is to prevent intermingling of flits from multiple packets, flit-by-flit routing simplifies VC allocation. AFC leverages this observation both (1) to improve performance by increasing the number of VCs *while* shortening the router pipeline via *lazy VC allocation* at the next router, which is our third novel mechanism, and (2) to reduce energy by shrinking the per-VC buffer and the total buffer (despite having more VCs).

Neither of these two optimizations is possible in traditional backpressured networks. Such lazy VC allocation is not possible due to basic correctness requirements of VC flow-control. Such buffer reduction is also not possible because conventional VC allocation does not scale to a large number of VCs and because reducing per-VC buffer depth has a significant impact on performance. AFC's shallower buffers recapture a significant fraction of the energy overhead of wider flits and more than compensate for the area overhead of supporting both flow-control mechanisms.

In summary, the paper's contributions are:

- We demonstrate backpressured and backpressureless networks are not robust in performance and energy across the spectrum of high and low loads.
- To address this robustness issue, we propose an adaptive flow control (AFC) router which employs *local contention thresholds*, *gossip-induced mode-switch*, and *lazy VC allocation*. The first mechanism maximizes performance (and minimizes energy) in the common case, and the second mechanism ensures correctness in corner cases. The third mechanism exploits flit-by-flit routing in AFC's backpressured mode to simplify VC allocation and reduces the buffer requirements by a factor of two in AFC's backpressured mode.
- Simulations using commercial workloads and SPLASH-2 confirm AFC's robustness by showing

that AFC achieves performance and energy that are closer to that of the *better* of backpressured and backpressureless routers.

The rest of the paper is organized as follows. Section II analyzes the impact of flow control on performance and energy. Section III describes adaptive flow control. Section IV describes our evaluation methodology. Section V discusses experimental results. Related work is described in Section VI. Finally, Section VII concludes this paper.

II. Impact of flow control on performance and energy

Backpressured and backpresureless flow control differ primarily in how they handle link contention. From this difference, a number of other secondary differences emerge. To illustrate the differences, consider how the following scenario is handled by the two flow control techniques: two flits at two different input ports of a router contend for the same output port.

In traditional backpressured networks, one flit is allowed to traverse the desired output link while the other is buffered locally. To prevent subsequent flits from overwriting the stalled flit, backpressure is implemented via credit tokens which let neighboring routers know whether buffers are free. To ensure that the stalled flit does not prevent subsequent flits from utilizing idle links, the input queues employ multi-flit buffers. To ameliorate head-of-line (HOL) blocking in such input-queued routers, routers employ multiple VCs[2] per physical channel. The operation of a canonical backpressured router may be viewed as four key steps (not necessarily corresponding to pipeline stages, as explained later). In the first step, the header flit of a packet is routed (R) to a set of output ports. The second step is the VC allocation (VCA) stage where the header flit requests a VC from among the free VCs on the output ports of interest. In the switch arbitration (SA) step, flits with an allocated VC compete for output ports. In the fourth step the flit traverses the switch (ST) and links (LT) (which may take multiple cycles) to be deposited at the input buffers of the neighboring routers.

The above steps may not correspond to pipeline stages because of several performance/energy optimizations including (a) lookahead routing (LAR) [8], wherein the routes are computed one hop earlier, (b) speculative overlapping of dependent functions, (e.g., [9]) and (c) aggressive router microarchitectures that can exploit other buffer/crossbar bypass paths to minimize router delay and energy [2]. Table I shows an ideal two-stage backpressured router in which we charitably assume that VCA occurs in zero cycles. Realistically, VCA delay can be hidden only by successful speculation, which is more likely at low loads.

[2]More importantly, VCs have a primary role in deadlock avoidance when used with routing algorithms that allow cyclic dependences among physical channels/links. However, in this paper, we consider provably deadlock-free dimension-ordered routing (DOR), where VCs have no deadlock-avoidance functionality. Instead, VCs offer performance benefits because of reduced HOL blocking.

Backpressureless flow-control, on the other hand, allows one contending flit to progress on the desired port. The other flit is either dispatched to an output port that may potentially take it farther from its destination (i.e., deflection) or dropped altogether. In either case, the routers are backpressureless since they are always ready to accept new flits because the old flits are either deflected or dropped[3].

Misrouting vs. Dropping flits: In this paper, we focus on the flit-by-flit deflection routing variant of backpressureless routing [3] because the variant that drops packets saturates at lower loads, even according to the original paper [4]. Because deflection routers ensure that each incoming flit is dispatched on an output port in each cycle [6], no flit is ever blocked. Consequently, deflection routers effectively avoid deadlocks and HOL blocking without the use of VCs.

Key complexities of deflection-based backpressureless routing: Recent critiques of backpressureless routing [4], [10] have focused on two key complexities in backpressureless routing, concluding that they must be overcome before such routing becomes practical. We observe that the complexities are not fundamental and arise because of specific design choices – there exist alternative backpressureless designs that avoid these complexities altogether.

The first perceived complexity is that deflection routers require worst-case buffering at *each node* for reordering and reassembly to handle the possibility of out-of-order flit delivery. Specifically, this complexity may be divided in to two categories: buffering for expected packets (the easy case) and buffering for unexpected packets (the difficult case). In the easy case, reordering and reassembly does not impose any additional cost for expected packets because expected packets are those for which a coherence request has been sent, which implies that there exists a (local) MSHR entry to receive the packet. MSHRs provide such receive-side buffering functionality even in traditional backpressured networks where flits from different packets may be intermingled because they may arrive on different physical/virtual channels, as also noted by Moscibroda *et al.* [3]. One may think that backpressureless will further complicate receive-side buffering because flits of the *same* packet may arrive in arbitrary order in backpressureless routers, whereas backpressured networks can only see arbitrary intermingling of flits across different packets. However, there is no additional complexity as both cases require a single random access memory array for MSHR buffers.

The difficult case involves unexpected packets which may occur due to dirty-writebacks. Such unexpected packets do not have pre-allocated MSHR entries to serve as receive-side buffers. In this case, a naive backpressureless implementation will indeed require worst-case buffering at *each node* for as many write-buffer entries in the *entire system*. However, such worst-case buffering can be avoided by using coherence

[3]Though there is no backpressure on the network ports, backpressureless routers do exert backpressure on the injection ports where new flits are not accepted unless an output port is free after accounting for network flits.

Table I
ROUTER PIPELINE STAGES

Flow Control	Router stage 1	Router stage 2	Link traversal
Backpressured	SA (PV→P) LAR in parallel VCA (0-cycle)	ST + partial LT	partial LT + input BW
Backpressureless	R + SA (P→P)	ST + partial LT	partial LT + latch-write
AFC (backpressureless mode)	Same as backpressureless		
AFC (backpressured mode)	SA (PV→P) LAR in parallel	Same as backpressured	Same as backpressured Lazy VCA

protocol variants that (1) pre-allocate an MSHR entry at the destination, and (2) hold writeback buffer data till such pre-allocation is possible. Note, in the absence of such restrictions even backpressured networks will see an increase in receive-side buffering, albeit less than backpressureless networks.

A second perceived complexity of deflection routing in [4], [10] is that it is fundamentally slow because it requires implementation of hardware priorities to ensure livelock freedom. This complexity is specific to implementations that use hardware priorities in order to guarantee that output ports are assigned to older (higher priority) flits before being assigned to younger (lower priority) flits. Such priorities ensure that the oldest flit at each router is never misrouted, hence guaranteeing forward progress. However, there are alternative implementations that avoid the use of priorities (which are necessary only for deterministic livelock freedom) and instead, rely on randomization and probabilistic guarantee of livelock freedom for the backpressureless router, as done in the Chaos router [5]. We emphasize that the probabilistic nature does not make the guarantee weak because the probability of a flit not reaching its destination diminishes with each hop and can eventually be made arbitrarily small (i.e., smaller than *any* adversarially-chosen ϵ ($0 \leq \epsilon \leq 1$)).

Summary: For the backpressured and backpressureless implementations described above, we have the following performance energy expectations. On the performance front, backpressureless networks are comparable to backpressured networks at low loads, but are significantly worse at high loads due to excessive misrouting and early saturation. On the energy front, backpressured networks achieve lower energy consumption that backpressureless networks at high loads. However, backpressureless networks achieve lower energy consumption than backpressured networks at low loads, by completely avoiding both static and dynamic buffer energy. The above observations, in conjunction with the fact that there are significant variations in network load across (and to a lesser extent, within) applications, directly make a case for an adaptive flow-control mechanism that captures the best of both worlds.

III. ADAPTIVE FLOW CONTROL

We describe the operation of AFC in terms of the following three questions. First, Section III-A answers the question: *What are the mechanisms that enable each AFC router to operate in either backpressured or backpressureless mode?* The next two subsections deal with the policy questions to achieve good performance and energy in the common case of uniform (high or low) load: *When do the* **forward mode-switch** *from backpressureless mode to backpressured mode and the* **reverse mode-switch** *from backpressured mode to backpressureless mode occur?* Our policies use our first mechanism *local contention thresholds*. Section III-D answers the question: *How does AFC achieve correctness in the corner cases of interactions between routers in different modes?*. Our second mechanism *gossip-induced mode-switch* ensures correctness in the corner cases. Section III-E focuses on the pipeline implementation of AFC, with a focus on our third mechanism – *lazy VC allocation*. Finally Section III-F discusses deadlock- and livelock-freedom for AFC networks.

A. Router organization

We describe the AFC router organization by focusing on key similarities and differences of three router parameters with respect to the basic backpressureless and backpressured routers.

First, the flits of the AFC router are wider because they have to include control information for both backpressured and backpressureless routers. Because AFC has to operate in the backpressureless mode at low loads, the AFC inherits hardware support for flit-by-flit routing from backpressureless routers (Section II). This inheritance implies that links, buffers, and crossbars are wider to include sequence/packet numbers (for reassembly) and destination-node (for routing) in each flit. Similarly, each flit also carries a VC-identifier as required by backpressured routers. However, as we describe later in Section III-E, our *lazy VC allocation* reduces the number of bits that need to be propagated. AFC routers operating in the backpressureless mode continue to propagate the VC information along the next hop even though the router itself does not use the information (Section III-E).

Second, the AFC router inherits input buffers from backpressured routers. One may think that the inclusion of buffers results in AFC suffering energy/area penalty over backpressureless routers. However backpressureless routers incur significant performance and energy degradation at high loads. AFC's energy overhead is minimal at low loads because the buffers are bypassed when the AFC router is in the backpressureless mode. Further, AFC buffers use power gating [7] when in the backpressureless mode to avoid leakage energy. Note, such power-gating is practical in AFC because we power-gate at the granularity of an entire physical port's buffers. One may think that traditional backpressured routing can capture similar benefits by partially powergating buffers at low loads (which is precisely

when AFC is able to do powergating, as well). However, because VC buffers are implemented as circular buffers, different contiguous sets of buffer-entries may be active in different VCs. Therefore, per-flit gating will be needed, which is impractical. Further, such per-flit gating (or gating entire VC buffers) complicates neighboring credit management (or VC allocation). Moreover, reducing buffering in backpressured networks with multi-cycle links introduces credit-management pipeline bubbles. Finally, an AFC router is likely to be smaller than a full-blown backpressured router due to the smaller buffers afforded by our lazy VC allocation (Section III-E).

Third, the AFC router includes credit propagation mechanisms to track per-VC buffer availability in neighboring nodes, as required by backpressured routing. Credit backflows would unnecessarily add to the energy cost when the AFC router is in backpressureless mode where credits are meaningless. To avoid this energy overhead, we include a special control line (one bit) to indicate to adjacent nodes to stop/start credit tracking when the router switches to backpressureless/backpressured mode.

B. Forward mode-switch using local contention thresholds

Ideally, the forward mode-switch must occur when load levels are high enough that backpressureless routers are near saturation. The actual load at which this switch occurs may be dependent on *global* application traffic characteristics and hence independent of *local* injection rate. One option to detect near-saturation loads is to track the number of cumulative misroutes of flits and monitor if those exceed thresholds. However, that approach has two problems. First, it adds the overhead of modifying flits as they progress through each router. Second, high contention may be detected in an incorrect network region because a flit that passes through a high contention region may have undergone a number of misroutes, but the number may exceed the threshold only after it has already passed through the high-contention region and has reached a low-contention region. Therefore, AFC fundamentally requires local measures of contention.

AFC measures contention via local traffic intensity. AFC uses the number of network flits traversing through the router averaged over the previous 4 cycles, and further smoothed using an exponentially weighted moving average (EWMA) as a metric of recent traffic intensity. Smoothing using EWMA was necessary to avoid frequent (and unnecessary) mode switches due to transient bursts of network activity. The measured traffic intensity is compared to an experimentally-determined *local contention threshold*, which is our first mechanism. The forward mode-switch is illustrated as the top transition in Figure 1. Because routers at edges and corners in a mesh have fewer ports, their thresholds are scaled accordingly. One may think that network traffic intensity could trigger false mode-switches because routers may observe high flit throughput without any link contention for "easy" traffic patterns (e.g., only near-neighbor communication). However, partly because real application traffic is not "easy" and partly because deflection induces further randomness, we found that the local contention thresholds were effective in detecting local load levels. Thus, our threshold is independent of application characteristics and is dependent only on the network configuration.

Once triggered, the mode-switch is realized over $2L$ cycles (where L is link latency). A mode-switch that begins in cycle T, will continue to receive any incoming flits in the input latches of the backpressureless mode. A notification to neighboring routers (via the credit-count start notification, as described in Section III-A) lets them know that they should start counting credits in cycle $T+L$ *even if the neighbors are in backpressureless mode*. Because all flits received in the $(T+2L-1)^{th}$ cycle are guaranteed to have been dispatched in the $(T+2L)^{th}$ cycle in the backpressureless mode, the backpressured mode can safely start from the $(T+2L)^{th}$ cycle onwards (starting with the routing stage). Thus, any incoming flits that are received on or after the $(T+2L)^{th}$ cycle are directed to the input buffers of the backpressured mode. Note that flits coming from previous backpressureless routers will still carry some VC information (Section III-A) even though the routers do not allocate any VCs for the benefit of any backpressured routers downstream.

C. Reverse mode-switch

Just as the forward mode switch occurs under high-load conditions, AFC attempts the reverse mode-switch when the measured load falls below a different (lower) threshold. We use the two thresholds as a hysteresis mechanism to avoid frequent mode-switches in the case of a single threshold when the load hovers around the single threshold. Instead, AFC maintains the previous mode when the load is between the high and low thresholds.

However, while performance/energy may indeed be maximized by switching to backpressureless mode as soon as load falls below a threshold, correctness requires that the reverse mode-switch cannot be initiated unless the buffers are empty. Without such a condition, flits remaining in buffers could be indefinitely trapped, leading to starvation. The reverse mode-switch is shown in the bottom transition in Figure 1. Once local buffers are empty, the router automatically starts operating in backpressureless mode in the very next cycle by bypassing incoming flits to the pipeline latches (for deflection) rather than to the input buffers. Subsequently, the switched router notifies its neighboring routers to stop accounting for credits. Upon receiving the notification, the neighbors simply set the buffer occupancy of the switched router to empty without waiting for actual credits. There is a time gap between when the switched router switches its mode and when the neighbors receive the notification (i.e., know that the switch has occurred). In this gap, the neighbors may have sent some flits to the already-switched router and decreased the router's credits not knowing that the switch has already occurred and that credit accounting has become unnecessary. However, because the discrepancy

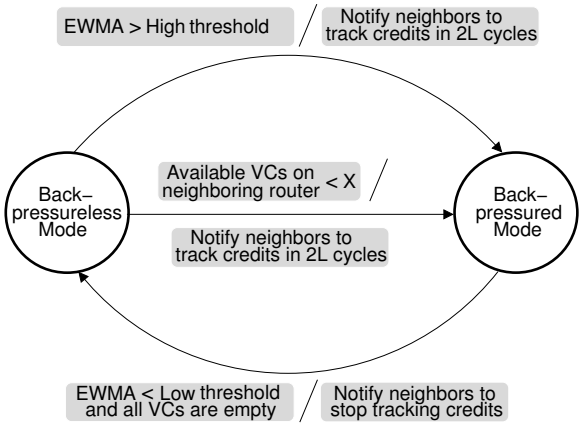


Figure 1. AFC Mode Transitions

leads only to unnecessary accounting for a brief time period, no correctness issues arise.

D. Handling interaction among modes using gossip-induced mode-switch

Given that each AFC router switches modes independently, adjacent routers may be operating in different modes. Such a situation may occur as a stable state when there are large spatial disparities in load. Even when there are no steady-state spatial load variations, the situation may occur when there are transient hot spots.

Communication from a backpressured router to a backpressureless router is the easy case, because a backpressureless router is always willing to accept flits from neighboring routers. The difficult case arises when a backpressureless router communicates with a backpressured router because the backpressureless router may *always* dispatch flits on all of its output links, but the backpressured router must be able to prevent incoming flits so that buffered flits are not overwritten.

AFC responds to this mismatch with a "scalpel and sledgehammer" approach wherein (1) AFC initially attempts to tolerate the mismatch with a lightweight response (the scalpel) which attempts to handle hot spots locally without spilling over to neighboring, low-contention regions; and (2) if the light-weight response fails, AFC responds with a heavyweight response (the sledgehammer) that guarantees correctness by expanding the backpressured region when the effects of contention cannot be contained. As part of the lightweight response, AFC ignores the mismatch as long as the downstream backpressured router has spare buffer capacity (as tracked locally using credits). Recall that the backpressureless routers begin to track credits as soon as their neighboring routers switch to backpressured mode (Section III-B). If the credits indicate that a downstream router's buffers are being exhausted (say, only X free buffers remain), then AFC employs the heavyweight response of a *gossip-induced* mode-switch, our third mechanism, wherein the neighboring backpressureless router is forced to switch to backpressured mode using the forward mode-switch (Section III-B). The threshold X must be at least $2L$ (we use $2L$) to allow sufficient time for a forward mode-switch (Section III-B). The gossip-induced mode-switch is shown as the middle-transition in Figure 1.

E. Lazy VC Allocation and its impact on the router pipeline

Recall from Section II that existing methods for achieving shallow pipeline depth in backpressured routers at low loads fundamentally rely on speculation. Such pipelines degrade to deeper router pipelines (3 stages, assuming lookahead routing) at higher loads because of the need for VC allocation on a per-packet granularity to ensure that flits of packet stay together. However, with AFC's use of flit-by-flit routing, VC allocation in the backpressured mode can be vastly simplified to operate at the flit-level and thus can be absorbed into other pipeline stages. Further, we leverage the simplification offered by lazy VC allocation to reduce overall buffer requirements which compensates for a significant fraction of the energy overhead of AFC's wider links.

To understand AFC's lazy VC allocation, we first examine the purpose of VCs and then examine the impact of flit-by-flit routing on VC allocation/deallocation. VCs serve two key purposes.

First, VCs are used to achieve deadlock-freedom by introducing VC traversal restrictions (rather than the more-limiting physical channel traversal restrictions used by earlier router designs) that can prevent deadlock cycles. However, AFC employs dimension-ordered routing (DOR) which also provides deadlock-freedom, and hence AFC does not use VCs for this purpose. Nevertheless, AFC must still respect higher-level deadlock avoidance rules when allocating VCs. For example, if coherence requests and responses are in two different virtual networks, VC allocation must respect such restrictions. In such cases, we can view the overall virtual channel as a two-tuple consisting of a virtual network and the virtual channel within the virtual network.

Second, traditional VC flow-control prevents intermingling of flits of different packets which is necessary when a multi-flit packet is the smallest independently-routed unit. To prevent flit intermingling, VC allocation has to be globally coordinated to ensure that the following two rules are observed. (R1) No packet may be assigned a VC that has previously been assigned to another packet (but not yet freed). (R2) No two packets may be assigned the same VCs in the same cycle. When implemented as above, packets in different VCs are allowed to overtake/bypass one another, thus reducing HOL blocking.

Unlike traditional packet-switched, backpressured routers, AFC's backpressured mode uses flits (which may be viewed as single-flit packets) as the smallest independently-routed unit because packets may be broken up into flits at another backpressureless router. Because individual (independently routed) flits can be freely intermingled in input queues, VC allocation can be simplified by ignoring the two rules mentioned above. The first rule (R1) is impossible to violate because the busy state of a VC is used solely to prevent flit

intermingling in multi-flit packets which is a non-issue for AFC. Further, rule (R2) can be ignored and any VC may be allocated to any flit (which may result in multiple flits being allocated the same VC in the same cycle). Such duplicate VC allocation is acceptable (from a correctness perspective) in AFC because the multiple flits will be serialized by the crossbar-switch anyway. Thus VC allocation can be precomputed locally (which may use simple round-robin or randomized allocation) thus completely eliminating VC allocation as a separate pipeline stage. However, duplicate VC allocation may cause unnecessary HOL blocking at the next (downstream) router because flits within a VC have to be routed in order. While the above optimization effectively eliminates VC allocation as a separate step, the possibility of duplicate VC allocation remains a performance problem.

We are now faced with a dilemma. On one hand, we want purely local, pre-computable VC allocation to eliminate the pipeline stage. On the other hand, we want to avoid the inadvertent assignment of the same VC to multiple flits to avoid HOL blocking which fundamentally requires global coordination. AFC's lazy VC allocation resolves the above dilemma by assigning VCs at the downstream router by exploiting our observation that any VC allocation is legitimate. Lazy VC allocation views the K-flit input buffer SRAM structure as having K VCs per physical channel with a single flit buffer per VC. Further, unlike traditional credit-backflow where credits at the upstream router are tracked on a per-VC level, AFC tracks credits on a per virtual-network level. As long as a virtual network is not full, there exists at least one VC with an unoccupied flit buffer. The upstream router dispatches flits to the downstream router without a VC allocation with only the virtual network identifier, which remains unchanged from one router to the next. Upon receipt, the flit is placed into one of the free flit-buffer entries (say, the i^{th} entry), thus lazily allocating the i^{th} VC to the flit. Note, free slots may be pre-discovered using simple daisy-chaining mechanisms and adds no latency to the critical path. Because all flits are placed in different VCs, the design avoids artificial HOL blocking (independent flits with the same VC allocation) altogether.

AFC's lazy VC allocation captures one additional advantage beyond the twin benefits of eliminating the VC allocation stage and minimizing HOL blocking. Because VC allocation is greatly simplified by flit-by-flit routing, AFC can increase the number of VCs without slowing down the VC allocation stage as would occur in traditional backpressured routers. To offset the energy increase of more VCs, AFC employs shallower buffers which suffice due to the following reason. While backpressured per-packet routing allocates an entire buffer for a packet so that some of the buffer slots are empty while the packet's flits are processed spread over time, flit-by-flit routing avoids this underutilization by allocating only one slot for a flit, thereby enabling shallower buffers. We show later in Section V that AFC reduces the total buffer size by a factor of 2 while matching the performance of a tuned traditional backpressured router. AFC's shallower buffers mostly compensate for the energy overhead of its wider links.

Table II
SYSTEM CONFIGURATION

System	1 chip, 9 cores
Network	3x3 mesh, each node is a core and an L2 cache bank; flit width is 32-bit, 2 virtual control networks and 1 data network (2+2+4=8 VCs) with 8-flit deep buffers; 2-cycle link latency
Cores	4-way SMT, 4-issue out-of-order with 40-entry instruction window
Private L1 Caches	split I & D, each 64KB, 4-way set associative, 64-byte blocks, 2-cycle latency, 16 MSHRs
Shared L2 Cache	unified 18 MB with 9 banks, 16-way set associative with LRU, 12-cycle latency, 16 MSHRs
Memory	8 GB DRAM, 250-cycle off-chip access time, 2 DIMMs per channel, 2 ranks per DIMM, 8 banks per rank, 32 bank queue entries

The activity in each of AFC's backpressured mode pipeline stages are summarized in the last row of Table I. Because lazy VC allocation can easily fit (due to simple pre-computation) in the buffer-write stage, the AFC router can realistically operate in 2 cycles (as opposed to the generous 0-cycle VCA assumption for backpressured routers in Section II).

F. Deadlock and livelock freedom

One may think that a combination of backpressureless routing (where DOR rules are ignored) and backpressured routing (where we rely on DOR for deadlock freedom) can lead to deadlocks. However, we can prove the deadlock-freedom of AFC using the following two observations. First, deadlocks can occur in AFC only when all the flits in a deadlock "knot" are in routers that are backpressured. (If a single router is backpressureless, those flits are not blocked and hence can escape.) Second, given that all the routers are in backpressured mode, using DOR for the backpressured routers, we guarantee that backpressured routers are deadlock-free (i.e. escape paths always exist). (A similar property can be proved for non-DOR routers which use deadlock avoidance since escape paths will exist on some VCs.) The fact that those flits may have originally been deflected is immaterial.

Livelock-freedom is another important issue to address given AFC's use of backpressureless deflection routing at low loads, especially since AFC does not use (impractical) priorities to guarantee livelock freedom. Even in the absence of such priorities, deflection routing has been shown to be probabilistically livelock free [5]. Further, because AFC uses deflection routing only at low loads, the likelihood of a continuous chain of misroutes is even less likely. Thus, the probabilistic guarantees may be strengthened further. Recall from Section II that probabilistic guarantees are indeed strong guarantees.

IV. EXPERIMENTAL METHODOLOGY

We evaluate AFC using Wind River's Simics 3.0 [11] full system simulation platform and the GEMS [12] timing

models, which include an SMT-processor model (Opal), a memory system model (Ruby) and an interconnection networks model (Garnet) [13].

Simulated Machine Configuration: Table II summarizes the key parameters of our simulated machine. Simulating a 16-core system (with multi-threading, as exemplified by many recent "CMP of SMTs" [14], [15], [16]) proved infeasible because of long simulation times as simulator state spilled out of memory to swap-space. We employ conservative scaling to simulate a 3x3 network with 9 nodes. The scaling makes our results conservative because the saturation throughput for backpressureless routers is higher when the network is smaller.

Based on the arguments in Section II and Section III-E, all routers are simulated with 2-cycle pipelines (see Table I). Our configuration (number of VCs and buffer-depths) is energy-optimized for the backpressured base case. Adding more VCs (or increasing buffer-depths) resulted in no significant performance improvement.

We used 32-data bit flits in each direction as we found that they were energy-delay-squared optimal. The control link widths were chosen so that VCs, destination nodes, flit-numbers, and global MSHR identifier could be encoded. Such encoding required 9 bits for backpressured networks, 13 for backpressureless networks, 17 for AFC. Thus, the total flit width, including data and control lines, were 41 (backpressured), 45 (backpressureless) and 49 (AFC) bits. These flit widths are reasonable because (a) they correspond to fairly wide 80–94 bit buses for full-duplex communication, (b) they are similar to the on-chip network in Intel's Teraflops research chip [17], (c) wider widths will reduce AFC's overheads, and (d) wider widths will cause super-linear growth in crossbar area.

AFC Parameters: AFC uses 8 VCs for each of the two virtual control networks and 16 VCs for the virtual data network with 1-flit deep buffers in its backpressured mode. The per-physical-channel buffer size is therefor 32 ($= 8 \times 2 + 16$) flits in comparison with the baseline packet switched network which uses a 64-flit buffer (=$4 \times 8 + 2 \times 2 \times 8$). Recall, the reduction in buffer-size is enabled by lazy VC allocation. The local contention thresholds for the forward (reverse) mode switch are set to 1.8 (1.2) for the corner routers, 2.1 (1.3) for the edge routers and 2.2 (1.7) for the other routers. The weighting factor for EWMA is 0.99 (i.e., the moving average update computation is $m_{new} = 0.99 * m_{old} + 0.01 * l$ where l is the average load over the past four cycles).

Energy Modeling: The Garnet network timing model is integrated with callbacks to the Orion [18] network energy model to report energy dissipation. We model all the key additional hardware for backpressureless routers and for AFC including flit-latches and additional control links. Because the receive-side buffering for flit reassembly is required for both backpressured and backpressureless routers [3], and because they are associated with MSHRs we exclude the receive-side buffers from network energy. The MSHR buffer sizes do not vary with flow-control mechanism since they are provisioned for the worst case (e.g., in the worst case, all-but-one flit corresponding to each outstanding MSHR entry arrives at the node). We used the parameters for 70nm technology with V_{dd} of 1.0V and 3GHz frequency. We assume 2.5mm links. We realistically assumed 90% effective power gating when AFC power-gates buffers in its backpressureless mode.

Table IV
SIMULATION PARAMETERS FOR COMMERCIAL WORKLOADS

Workload	System Warmup (transactions)	Cache Warmup (transactions)	Measurement (transactions)
Apache	2 million	20,000	600
OLTP	0.1 million	5,000	50
SPECjbb	1 million	50,000	3,000

Finally there are previously proposed optimizations that target crossbar dynamic energy. For example, an aggressive variant of express virtual channels [2] proposes to use wires that bypass the crossbar switch for packets that traverse express virtual channels that proceed along the same dimension. Note, such an orthogonal optimization can be grafted on to any of backpressureless, backpressured or AFC router by adding bypass paths between appropriate pairs of ports and letting flits that traverse the corresponding ports to use the bypass paths instead of the crossbar. As such, we do not consider this optimization in our comparisons.

Workloads: While open-loop simulations have some value, relying solely on them is problematic because they set injection rates to arbitrary values which may or may not correspond to real workloads. Trace-driven evaluations do not include the feedback effect of the network on execution time. To avoid these problems, the majority of our experiments use execution times on multi-threaded applications to evaluate AFC. We do not use multi-programmed, sequential workloads because they lack coherence interactions which fundamentally change the network traffic. Our benchmarks include three high-load/commercial and three low-load/scientific multi-threaded applications (see Table III). Table III shows the injection rate achieved by each benchmark (in flits/node/cycle). We run the commercial benchmarks for a fixed number of transactions after adequate cache warmup (see Table IV). We scale scientific workloads from SPLASH-II benchmark suite [19] to run to completion. We also include an experiment with synthetic random traffic to highlight key performance/energy characteristics of AFC. We repeat all simulations multiple times to account for statistical variations.

V. RESULTS

Recall that, for the low-load applications, backpressureless networks are expected to consume less power than the backpressured networks. In contrast, for high load applications backpressured networks are expected not only to consume less energy than backpressureless networks but also to outperform backpressureless networks.

Table III
WORKLOADS: DESCRIPTION AND CHARACTERISTICS

Commercial Workloads
Apache: version 2.2.9, a static web server workload with repository of 20,000 files (˜500 MB). SURGE is used to generate web requests by simulating 4500 clients, each with 25ms think time between requests. *Inj. Rate = 0.78*
Online Transaction Processing (OLTP): models database transactions of a wholesale parts supplier. We use PostgreSQL 8.3.7 database system and DBT-2 test suite which implements TPC-C benchmark. We reduced number of items and districts per warehouse and customers per district to allow a larger number of warehouse. We use a database of 25,000 warehouses (˜5GB). We simulate 300 concurrent database connections. *Inj. Rate = 0.68*
SPECjbb: version 2005, Java-based 3-tier client/server system workload with emphasis on the middle tier. Java server VB version 1.5 with parallel garbage collection. We simulate a system with 90 warehouses. *Inj. Rate = 0.77*
SPLASH-2 Workloads
Barnes: implements the Barnes-Hut method to simulate an N-body problem. We use 8 threads with a problem size of 512 particles. *Inj. Rate = 0.1*
Ocean: simulates ocean movements based on eddy and boundary currents with contiguous partitions to enhance data locality. We use 8 threads with a problem size of 34x34 grid. *Inj. Rate = 0.19*
Water-nsquared (Water): simulates water molecules by solving Newtonian equations using a predictor-corrector method in each time step. We use 8 threads with a problem size of 64 molecules for one time step. *Inj. Rate = 0.09*

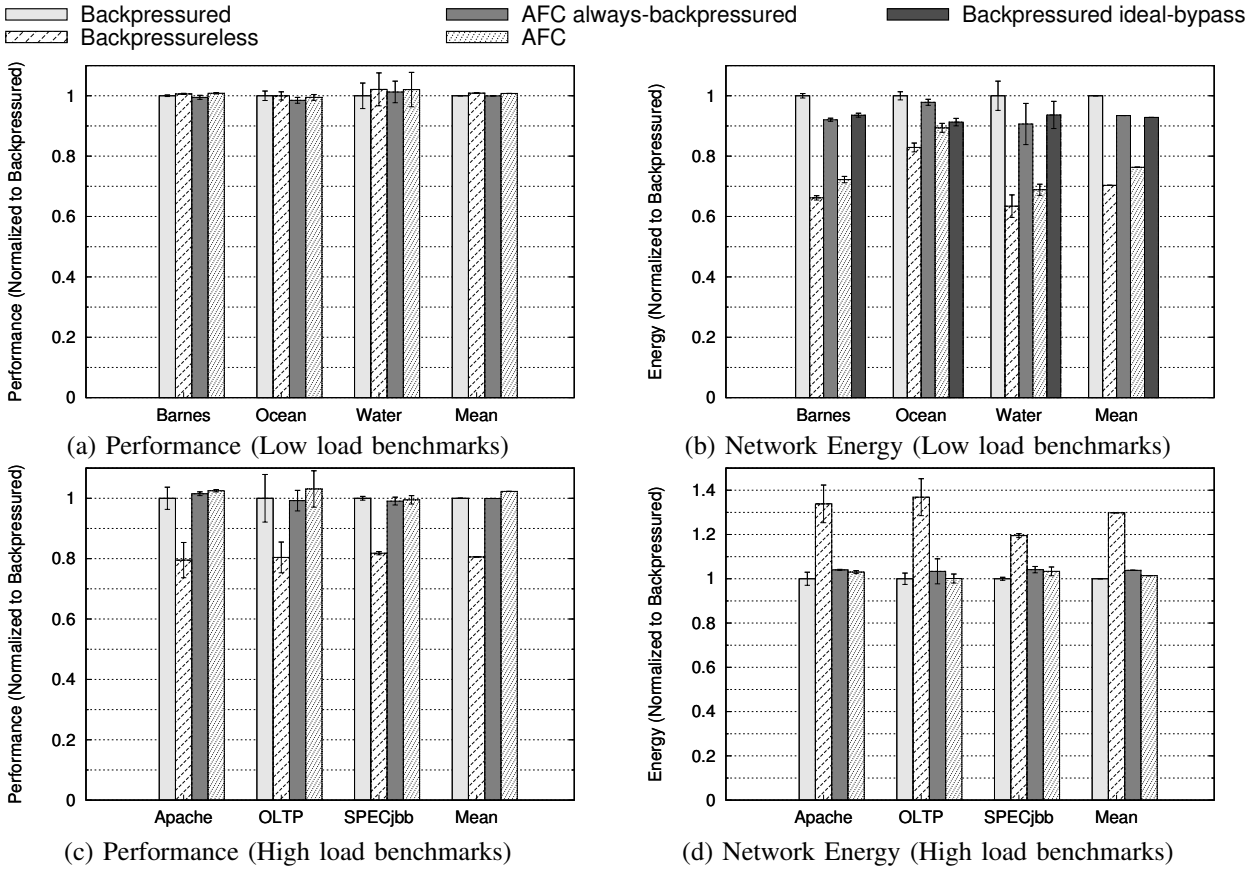


Figure 2. Performance and Energy Robustness

A. Performance and energy

Recall, AFC's key goal is to match the performance and energy of the better flow control mechanism at both low and high network loads. The four figures in Figure 2 plot the normalized performance (left) and network energy (right) at low loads (top) and high loads (bottom). For the performance graphs, higher is better and for the energy graphs, lower is better. Each graph includes the set of benchmarks (groups of bars) on the X-axis with the appropriate metric (performance or energy) on the Y-axis. The Y-axis numbers are normalized to that of the baseline backpressured network. Three of the bars within each group correspond to the three flow-control mechanisms being compared. We also show one other comparison. AFC combines mechanisms for adaptively switching between flow-control modes as well as mechanisms for optimizing the flit-by-flit backpressured mode with lazy VC allocation. To isolate the effects of the two sets of mechanisms, we include an AFC router which is always in the backpressured mode (called "AFC always-backpressured"). Finally, rather than compare AFC against the each of the many proposed buffer energy optimizations, we show a packet-based router in which *all* buffer dynamic energy is eliminated (called "Backpressured ideal-bypass"). This serves as a lower bound on energy for techniques that

elide buffer reads (but not writes) [1] as well as those that elide a fraction of both buffer reads and writes (Express virtual channels [2]). We show this bound only for the low-load, energy graph, which is where it is relevant. Each graph also includes the geometric mean (rightmost bars). The variance bars indicate the standard deviation of multiple runs of the benchmark.

We make four key observations. First, at low loads, flow control has no meaningful impact on performance (see Figure 2(a)). This is not surprising given our expectation that lack of contention implies that there is little misrouting in backpressureless routing. AFC, which operates in backpressureless mode at low loads achieves similar performance. The backpressured router and AFC's always-backpressured router are also similar in performance.

Second, flow control does have a big impact on energy (see Figure 2(b)). Backpressureless, which eliminates buffers and thus, all buffer energy, consumes the least energy. AFC, which is largely in backpressureless mode, achieves within 9% of backpressureless. This gap mostly comes from our assumption that power-gating the buffers eliminates only 90% of their static power. The basic backpressured router, without any buffer energy optimizations, is the most energy consuming (42% more than backpressureless). More interestingly, even backpressured-ideal-bypass, where *all* dynamic buffer energy is elided, is significantly worse (32%) than backpressureless. This result strengthens the argument that dynamic buffer power optimizations have fundamental limitations at low loads, where static power dominates. Further, the skew in favor of static power will only worsen as we move to future technology generations.

Third, at high loads, backpressureless routing suffers a significant degradation in performance relative to backpressured routing (19% on average, see Figure 2(c)). This degradation is due to excessive misrouting. AFC, which is largely in the backpressured mode, achieves comparable performance (within 2%). Not surprisingly, AFC-always-backpressured is also very similar. Note that the backpressured router, which assumes a fixed 2-cycle pipeline with 0-cycle VCA subsumes previous pipeline optimizations for backpressured routers (Section II).

Fourth, on the energy front, the behavior in terms of performance is mirrored in energy. Backpressureless, which has the worst performance, also dissipates the most energy, being 35% higher than backpressured, which is the least-energy configuration. In contrast, AFC incurs a modest energy overhead (2% on average, 3% worst-case) compared to backpressured.

In summary, AFC matches the performance of better of both backpressureless and backpressured flow control at both high and low loads. It approaches the better of the two in terms of energy as well (within 3% at high loads and within 9% at low loads). In contrast, non-adaptive backpressureless flow control incurs a 19% performance penalty and a 35% energy penalty at high loads. Conversely, at low loads, non-adaptive backpressured flow control incurs an energy penalty of 32% (on average) even with ideal buffer bypassing.

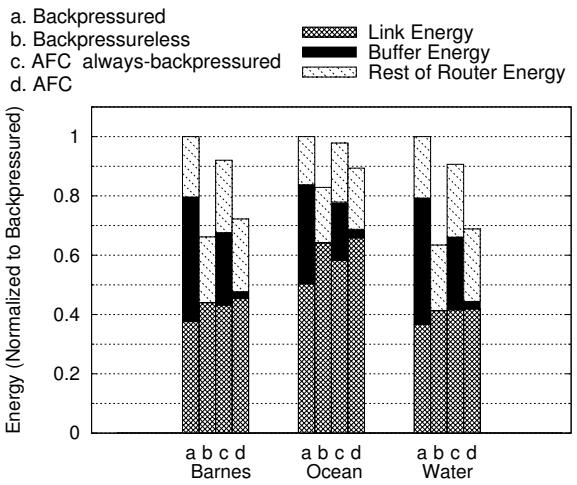
Energy breakdown: Figure 3(a) and Figure 3(b) plot the normalized energy (normalized to backpressured router's energy, Y-axis) for our low load and high load benchmarks (X-axis), respectively. Each flow-control mechanism (bars within group) is shown for each benchmark. Further, the overall network energy is shown partitioned into buffer energy, link energy and other router energy (which includes crossbar energy and arbiter energy).

For low-load applications (Figure 3(a)), all three benchmarks exhibit largely similar energy profiles. The breakdown for backpressured routers indicates that buffer energy is significant, even in the case with the smallest proportion of buffer energy (`ocean`). In contrast, backpressureless routers eliminate all buffer energy for a modest increase in link energy. Because AFC largely stays in backpressureless mode, it too eliminates most buffer energy. Finally, though AFC-always-backpressured reduces some buffer energy because it uses half as much buffer space as the backpressured router, buffer energy remains a significant fraction.

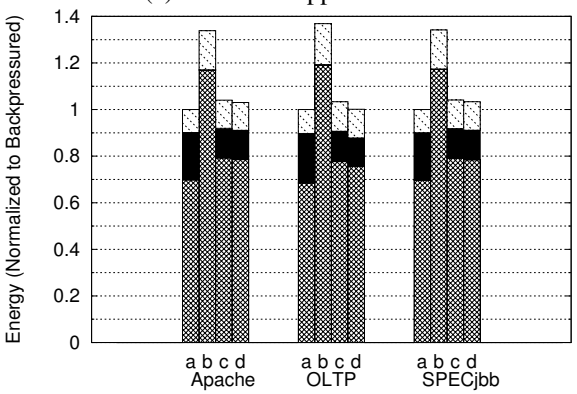
On the other hand, at high loads, backpressured mode achieves the lowest energy across all benchmarks. Backpressureless routers incur a significant link energy penalty due to excessive misrouting as mentioned before. There is little difference between AFC and AFC-always-backpressured because AFC largely stays in the backpressured mode. We observe that the overall energy penalty of AFC (relative to backpressured) is the difference between increased link energy (due to wider flits) and reduced buffer energy (due to lazy VC allocation).

Mode duty cycle and spatial variation: We measured the fraction of time spent by AFC in the two modes for all our workloads. Four of the six benchmarks were uniformly high or low load without any variation in time. For example, `water` and `barnes` were in backpressureless mode 99% of the time. Similarly, `specjbb` and `apache` were in the backpressured mode more than 99% of the time. The other two benchmarks exhibited a small amount of variation. For example, routers spent 7% of `ocean`'s execution time in backpressured mode and 5% of `oltp`'s execution time in backpressureless mode.

Interestingly, although our runs did not see any gossip-induced mode-switches, we did see them in an open-loop network experiment which created hotspots. Recall, gossip-induced mode switch is required for correctness and is therefore justified even if our runs do not exercise it. The lack of gossip-induced mode switching in our runs indicates that either transient hot spots did not develop because the network load was uniform, or if any transient hot spots developed, they were tolerated by the "scalpel" (Section III-D) wherein adjacent routers did not see any backpressure that would have forced them to switch modes (i.e., the hot-spot would spread).



(a) Low load applications



(b) High load applications

Figure 3. Network Energy Breakdown

B. Open-loop evaluation for spatial variation

Because of the near-perfect spatial uniformity of load in our runs, we used synthetic traffic with open-loop simulation to simulate spatial load variation. The configuration is designed to mimic a consolidation workload in an 8x8 multicore in which a different application runs in each quadrant. One quadrant of the network injected packets at a fixed high rate (0.9 flits/node/cycle) and the other three quadrants injected packets at fixed low rates (0.1 flits/node/cycle). The destinations were chosen such that traffic injected in a quadrant stayed within the quadrant (except possibly due to misrouting).

In the absence of variation, AFC can only approach the best of either backpressured or backpressureless routers. However, with the spatial variation described above, AFC was the best energy configuration because neither backpressured (9% more energy than AFC) nor backpressureless (30% more energy than AFC) flow control was robust in handling the load variation. We also observed that (1) backpressured and AFC achieved 33% lower latencies than backpressureless in the high-load quadrant, and (2) the high-load quadrant had an adverse impact on a neighboring low-load quadrant's latencies because of misrouting.

Other results: Experiments with open loop, uniform random traffic injected at various rates revealed that (1) all flow-control techniques achieve similar latencies at low loads (2) AFC and backpressured networks achieve near identical saturation throughput (whereas backpressureless saturates at lower offered loads).

VI. Related Work

There is a rich body of work on each of the two flow control mechanisms. Deflection routing, first proposed in [6], has seen several variants implemented in real machines and in research prototypes [20], [21], [5], [22]. Deflection routing has also been studied extensively [23], [24]. More recently, researchers have refocused on deflection routing as an attractive option for energy-constrained on-chip networks [25], [26], [27], [3]. All the above variants of deflection-based routing either drop packets or suffer from high latencies at saturation (which is typically at lower loads than with backpressured routers). In contrast, AFC adaptively changes the flow-control to enable backpressured mode of operation.

Similarly, there has been extensive research on backpressured networks with credit-based flow control. Since our focus is on optimizing energy and latency in backpressured networks at low loads, we discuss only the work relevant to those goals. For example, Wang *et al.*propose a technique to bypass buffer-reads under low loads when there is only one flit in the buffer [1]. While such techniques do help reduce dynamic energy, they are not as energy efficient as eliminating all buffer dynamic energy and most static energy (using power-gating) as in the backpressureless mode of AFC, even at low loads. There have been techniques proposed to target leakage power of buffers by placing buffers in inactive modes [28]. In general, they require fine-grained (flit-by-flit) power-gating which may be unviable, especially given our small buffers. Techniques that speculatively overlap key pipeline stages (e.g., VC allocation and switch arbitration in [9]) attempt to avoid the latency penalty of a backpressured-router's pipeline stages at low loads. Our lazy allocation goes one step further and removes the dependence between VC allocation and switch allocation.

While single-cycle routers have been proposed [29], [4], they will likely need a slow clock to accommodate both switch arbitration and switch traversal. For example, the router in [29] employs speculative switch arbitration in parallel with switch traversal. However, the router design assumes that mis-speculations can be caught and recovered-from in the same cycle. Effectively, this assumption implies that the router can ensure a conflict-free switch arbitration and switch traversal in the same cycle. In the case of SCARAB [4], switch arbitration and switch traversal are non-speculative and must fit in a clock cycle.

VII. Conclusion

As the microprocessor industry packs more cores into a chip, multi-hop interconnection networks are likely to

be used as the on-chip communication fabric. Network performance has a direct impact on overall system performance. In turn, flow-control mechanisms have a first-order impact on the performance and energy of networks. Two widely-studied flow-control mechanisms – credit-based backpressured flow control and backpressureless deflection flow control – have their own particular network load "sweet spots" where they operate well. Unfortunately, they incur significant performance/energy penalties at loads outside their sweet spots. For example, backpressureless networks achieve low energy at low network loads, but suffer from excessive misrouting at high loads, which leads to poor performance and energy. Similarly, backpressured networks are energy-efficient and achieve high throughputs at high network loads that backpressureless networks cannot reach. However, backpressured routers incur a energy penalty at low loads. If network loads for real applications were predominantly in either high- or low-load region, one of the flow control mechanisms would suffice. Unfortunately, workload characteristics are not limited to the "sweet spot" region of any single flow control mechanism.

We propose Adaptive Flow Control (AFC) – a robust flow control mechanism with a wide sweet spot that spans high and low loads. AFC routers operate in backpressureless mode at low loads and as backpressured routers at high loads. Consequently, AFC avoids the significant energy/performance penalties that each of the two flow-control policies incur when operating outside their sweet spots. Evaluation with a suite of multi-threaded commercial and scientific/engineering workloads reveals that AFC's performance and energy are close to those of the better of backpressured and backpressureless routers. As the number of cores continues to scale, and as the mix of applications grows more diverse, AFC's performance and energy robustness will be increasingly important.

Acknowledgments: We thank the anonymous reviewers for their feedback. This work is supported in part by National Science Foundation (Grant no. CCF-0541385).

REFERENCES

[1] H. Wang, L.-S. Peh, and S. Malik, "Power-driven design of router microarchitectures in on-chip networks," in *MICRO 36: Proc. of the 36th Annual IEEE/ACM Int'l Symp. on Microarchitecture*, 2003, p. 105.

[2] A. Kumar, L.-S. Peh, P. Kundu, and N. K. Jha, "Express virtual channels: towards the ideal interconnection fabric," in *ISCA '07: Proc. of the 34th Annual Int'l Symp. on Computer architecture*, 2007, pp. 150–161.

[3] T. Moscibroda and O. Mutlu, "A case for bufferless routing in on-chip networks," in *ISCA '09: Proc. of the 36th Annual Int'l Symp. on Computer Architecture*, 2009, pp. 196–207.

[4] M. Hayenga, N. E. Jerger, and M. Lipasti, "SCARAB: a single cycle adaptive routing and bufferless network," in *MICRO 42: Proc. of the 42nd Annual IEEE/ACM Int'l Symp. on Microarchitecture*, 2009, pp. 244–254.

[5] S. Konstantinidou and L. Snyder, "The chaos router: a practical application of randomization in network routing," in *SPAA '90: Proc. of the Second Annual ACM Symp. on Parallel Algorithms and Architectures*, 1990, pp. 21–30.

[6] P. Baran, "On distributed communications networks," *IEEE Trans. on Communications Systems*, vol. 12, no. 1, pp. 1–9, March 1964.

[7] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits*, 2nd ed. Prentice Hall Inc., 2002.

[8] W. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., 2003.

[9] L.-S. Peh and W. J. Dally, "A delay model and speculative architecture for pipelined routers," in *HPCA '01: Proc. of the 7th Int'l Symp. on High-Performance Computer Architecture*, 2001, p. 255.

[10] G. Michelogiannakis, D. Sanchez, W. J. Dally, and C. Kozyrakis, "Evaluating bufferless flow control for on-chip networks," in *Proc. of the Fourth ACM/IEEE Int'l Symp. on Networks-on-Chip*, 2010, pp. 9–16.

[11] P. S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, G. Hållberg *et al.*, "Simics: A full system simulation platform," *Computer*, vol. 35, no. 2, pp. 50–58, 2002.

[12] M. M. K. Martin, D. J. Sorin, B. M. Beckmann, M. R. Marty, M. Xu *et al.*, "Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset," *SIGARCH Comput. Archit. News*, vol. 33, no. 4, pp. 92–99, 2005.

[13] L.-S. Peh, N. Agarwal, N. Jha, and T. Krishna, "Garnet: A detailed on-chip network model inside a full-system simulator," in *Int'l Symp. on Performance Analysis of Systems and Software (ISPASS)*, 2009.

[14] N. Kurd, J. Douglas, P. Mosalikanti, and R. Kumar, "Next generation Intel micro-architecture (Nehalem) clocking architecture," in *IEEE Symp. on VLSI Circuits*, 2008, pp. 62–63.

[15] P. Kongetira, K. Aingaran, and K. Olukotun, "Niagara: A 32-way multithreaded Sparc processor," *IEEE Micro*, vol. 25, no. 2, pp. 21–29, 2005.

[16] C. N. Keltcher, K. J. McGrath, A. Ahmed, and P. Conway, "The AMD Opteron processor for multiprocessor servers," *IEEE Micro*, vol. 23, no. 2, pp. 66–76, 2003.

[17] S. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson *et al.*, "An 80-tile 1.28TFLOPS network-on-chip in 65nm CMOS," in *IEEE ISSCC Dig. Tech. Papers*, 2007, pp. 98 –99.

[18] H.-S. Wang, X. Zhu, L.-S. Peh, and S. Malik, "Orion: a power-performance simulator for interconnection networks," in *MICRO 35: Proc. of the 35th Annual ACM/IEEE Int'l Symp. on Microarchitecture*, 2002, pp. 294–305.

[19] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The SPLASH-2 programs: characterization and methodological considerations," in *ISCA '95: Proc. of the 22nd Annual Int'l Symp. on Computer architecture*, 1995, pp. 24–36.

[20] B. J. Smith, "A pipelined, shared resource MIMD computer," *Advanced computer architecture*, pp. 39–41, 1986.

[21] B. J. Smith, "Architecture and applications of the HEP mulitprocessor computer system," *Readings in computer architecture*, pp. 342–349, 2000.

[22] R. Alverson, D. Callahan, D. Cummings, B. Koblenz, A. Porterfield, and B. Smith, "The Tera computer system," in *ICS '90: Proc. of the 4th Int'l Conf. on Supercomputing*, 1990, pp. 1–6.

[23] J. Bannister, F. Borgonovo, L. Fratta, and M. Gerla, "A performance model of deflection routing in multibuffer networks with nonuniform traffic," *IEEE/ACM Trans. Netw.*, vol. 3, no. 5, pp. 509–520, 1995.

[24] N. Maxemchuk, "Comparison of deflection and store-and-forward techniques in the Manhattan street and shuffle-exchange networks," in *INFOCOM '89. Proc. of the Eighth Annual Joint conf. of the IEEE Computer and Communications Societies. Technology: Emerging or Converging*, 1989, pp. 800–809 vol.3.

[25] M. Millberg, E. Nilsson, R. Thid, and A. Jantsch, "Guaranteed bandwidth using looped containers in temporally disjoint networks within the nostrum network on chip," in *DATE '04: Proc. of the conf. on Design, automation and test in Europe*, 2004, p. 20890.

[26] Z. Lu, M. Zhong, and A. Jantsch, "Evaluation of on-chip networks using deflection routing," in *GLSVLSI '06: Proc. of the 16th ACM Great Lakes Symp. on VLSI*, 2006, pp. 296–301.

[27] C. Gómez, M. E. Gómez, P. López, and J. Duato, "Reducing packet dropping in a bufferless NoC," in *Euro-Par '08: Proc. of the 14th Int'l Euro-Par conf. on Parallel Processing*, 2008, pp. 899–909.

[28] X. Chen and L.-S. Peh, "Leakage power modeling and optimization in interconnection networks," in *ISLPED '03: Proc. of the 2003 Int'l Symp. on Low power electronics and design*, 2003, pp. 90–95.

[29] R. M. Andrew, A. West, and S. Moore, "Low-latency virtual-channel routers for on-chip networks," in *In Proc. of the 31st Annual Int'l Symp. on Computer Architecture*, 2004, pp. 188–197.