Achieving Lightweight Multicast in Asynchronous Networks-on-Chip Using Local Speculation

Kshitij Bhardwaj
Dept. of Computer Science
Columbia University
New York, NY 10027
khardwaj@cs.columbia.edu

Steven M. Nowick
Dept. of Computer Science
Columbia University
New York, NY 10027
nowick@cs.columbia.edu

ABSTRACT
We propose a lightweight parallel multicast targeting an asynchronous NoC with a variant Mesh-of-Trees topology. A novel strategy, local speculation, is introduced, where a subset of switches are speculative and always broadcast. These switches are surrounded by non-speculative switches, which throttle any redundant packets, restricting these packets to small regions. Speculative switches have simplified designs, thereby improving network performance. A hybrid network architecture is proposed to mix the speculative and non-speculative switches. For multicast benchmarks, significant performance improvements with small power savings are obtained by the new approach over a tree-based non-speculative approach. Interestingly, similar improvements are also shown for unicast. Finally, another benefit is to reduce the address field size in multicast packets.

1. INTRODUCTION
In today’s many-core era, on-chip networks have major impact on system-level power and performance. Networks-on-chip (NoCs) have been an active area of research for the last decade [1], [2]. Most of the NoC research has been devoted to improving performance, power and fault-tolerance for unicast (i.e. one-to-one) traffic. However, in recent years, multicast (i.e. one-to-many) has also seen growing interest, with several optimization strategies [3].

In multicast, the same packet is sent from a source to an arbitrary subset of destinations. Multicast is widely-used in various parallel computing applications: for example, in cache coherency protocols to send write invalidates to multiple processors, in shared-operand networks for operand delivery, and in multi-threaded applications for barrier synchronization [4]. Each of these applications cause significant multicast traffic in NoCs. For example, for the Token cache coherency protocol, 52.4% of injected traffic is multicast [5].

There is also a growing interest in supporting multicast in NoCs using emerging technologies, such as wireless [6] and photonic [7]. Other emerging areas include large-scale neuromorphic chip multiprocessors [8] and the use of CDMA to handle multicast [9]. Both these approaches support multicast in application-specific asynchronous NoCs.

Asynchronous NoCs are at the core of designing modern globally-asynchronous locally-synchronous (GALS) systems [10]. Asynchronous NoCs eliminate the need for a global clock and are therefore free from associated overheads: clock skew, clock tree switching power and complex clock gating circuitry. Several recent examples have shown significant power and area reductions, compared to the synchronous NoCs, while achieving similar or better performance compared to mesh topologies [18]. A variant of the traditional MoT topology has been proposed that achieves higher saturation throughput with lower contention compared to the original MoT topology [19]. Variants MoTs have been recently used for core-to-cache connections in high performance shared memory parallel processors [20].

Contributions. Given the overheads in recent synchronous multicast approaches, there is a need for more cost-effective multicast solutions. The focus of this paper is on asynchronous NoCs, driven by their potential for lightweight design, and the growing interest in their use for large-scale system integration. However, despite much recent activity in this area, there are no general-purpose asynchronous NoCs with multicast capability. The goal of this work is to achieve lightweight and power-performance efficient multicast using asynchronous NoCs. We propose multiple solutions to support efficient multicast, in asynchronous NoCs, using a new routing strategy and network architectures.
The first contribution is a lightweight tree-based parallel multicast using an asynchronous NoC. The target topology is MoT. To the best of our knowledge, this is the first general-purpose asynchronous NoC to support multicast. This solution, although simple, is not very efficient in terms of performance. Further enhancements are therefore proposed for performance, while still maintaining design simplicity.

The second contribution is a novel strategy called local speculation that achieves high-performance parallel multicast. In local speculation, a packet (unicast or multicast) is always broadcast at a fixed subset of speculative switches in the network. To restrict the distance traveled by any redundant packets to small “local” regions, these packets are throttled by neighboring non-speculative switches. This localized approach limits the penalties of speculation, by terminating redundant packets early, resulting in minimal impact on congestion and power. Speculative switches are built for speed, as they do not do any route computation or output channel allocation. Non-speculative switches perform throttling with almost no hardware overhead. Interestingly, local speculation improves network performance not only for multicast, but also for unicast. In addition, the simple, 'sub-cycle' operation for local broadcast and low-overhead throttling, not discretized to clock cycles, make the new paradigm a good match for asynchronous design.

As a third contribution, local speculation leads to a new hybrid network architecture that mixes speculative and non-speculative switches. This is the first time a hybrid NoC has been used to support multicast (cf. [4], [5], [17]). Moreover, unlike previous unicast work that uses speculation within a router for early VC allocation [1], this work uses speculation at the network-level to support efficient multicast.

As a fourth contribution, two more architectures are introduced besides hybrid, with extreme degrees of speculation for an exhaustive design space exploration. The first does not use any speculation, which is the same as our first tree-based multicast solution. In contrast, the second is an almost fully speculative architecture.

Finally, as a fifth contribution, protocol optimizations are introduced, to further improve the power and performance of speculative and non-speculative nodes. These optimizations are performed only for multi-flit packets, and are triggered by the header flit. For speculative switches, once the header is processed, the switch can revert to non-speculative mode for body and tail flits, thereby saving power. For non-speculative switches, once the header is processed, the channel remains allocated for the current packet, with no repeated route computation, thereby improving throughput.

The above new techniques have been incorporated into new networks and extensive experiments are performed. For multicast benchmarks, the new simple tree-based parallel multicast network achieves 39.1-74.1% lower latency, with only small power overheads, than a unicast-based serial multicast approach. The hybrid network with local speculation provides additional 17.8-21.4% latency improvements, and small power reductions, over our simple tree-based solution. We also consider an extreme case of an almost fully speculative network, which has the best performance but incurs 10.8-13.4% power overheads over the hybrid solution. Interestingly, similar results are also seen for unicast traffic.

2. BACKGROUND

This section presents background on asynchronous communication, a variant MoT topology, and a baseline asynchronous NoC, which form the foundation of the new work.

Asynchronous Communication. A handshaking protocol defines the channel communication between an asynchronous sender and receiver, using request/acknowledge wires. A two-phase (NRZ) protocol has only 2 events per transaction (toggle on req followed by toggle on ack) [21], while a four-phase (RZ) protocol has 4 total events (req/ack initially zero, with rising then falling req/ack tran-
3. PROPOSED APPROACH: OVERVIEW

This section introduces multiple solutions to achieve efficient parallel multicast, using new routing strategy and network architectures. Protocol optimizations of the new fanout nodes and five new networks based on these architectures and optimizations are also presented.

Simple tree-based multicast. The first contribution is a simple tree-based parallel multicast, applied for the first time to a general-purpose asynchronous NoC. Routing of a unicast packet is the same as in the baseline network. The fanout network architecture, in Figure 3(a), has all non-speculative nodes. New fanout nodes are designed to handle parallel replication, as described in Section 4(b).

Source routing is used to encode the address for every fanout node on each path to the destination(s). The address at each fanout node must encode 3 symbols: top route, bottom route or both. Therefore, 2-bit encoding is used for the address field of each fanout node.

This basic tree-based multicast is simple but not efficient in terms of latency and throughput. The new fanout nodes are slow due to expensive route computation and channel allocation protocols, required to handle a more complex set of transmission modes. Another limitation is that the source routing, as described above, leads to low packet coding efficiency, which does not scale with larger network sizes.

Local speculation-based multicast. A new strategy, local speculation, is introduced for high-performance parallel multicast. In local speculation, a subset of fanout nodes are speculative and always broadcast a multicast (or unicast) packet. These nodes are surrounded by non-

Figure 3: New fanout network architectures: (a)-(c) full range for 8x8 MoT, (d) One possible hybrid network for 16x16 MoT. (c) Hybrid with almost full speculation, (d) One possible hybrid fanout network: 16x16 MoT

Extra power due to speculative nodes is minimized by re-verting to the non-speculative mode for body flits of pack-
Figure 5: Unoptimized fanout nodes: (a) speculative, (b) non-speculative.
transparent state after the tail arrives. Therefore, the tail also gets speculatively routed to both output channels.

(d) Optimized non-speculative fanout node. The basic idea of optimization is to use the header to pre-allocate the correct output channel(s) for body/tail. These fits are fast forwarded on their arrival, without route computation and output channel allocation.

Structure and operation. There is only one key difference in terms of more simplified Output Port Modules. Based on the routing of the header, these modules now pre-allocate the correct channel for trailing body/tail fits. The routing of the tail is used to release the channel.

5. EXPERIMENTAL RESULTS

This section presents the experimental framework for evaluation of the new parallel multicast solutions, along with node and network-level results on area, performance, power, and addressing overhead.

5.1 Experimental Framework

Experimental case studies. Two distinct case studies are used to evaluate the proposed parallel multicast solutions: (a) contribution trajectory, and (b) architectural design space exploration. The contribution trajectory incrementally evaluates the effectiveness of each contribution, against a serial baseline: parallel multicast, local speculation, and a hybrid network, and protocol optimizations. Architectural design space exploration, on the other hand, only evaluates the effects of varying degrees of speculation on the new parallel multicast networks. To isolate the focus, only optimized networks are targeted, thereby eliminating any interference from the optimization strategies.

The contribution trajectory compares 4 networks: (i) Baseline [21], only supporting serial multicast; (ii) BasicNonSpeculative, using simple tree-based parallel multicast; (iii) BasicHybridSpeculative, using local speculation in a hybrid network; and (iv) OptHybridSpeculative, similar to the previous one, but including protocol optimizations.

The architecture design space exploration compares 3 optimized new networks with varying degrees of speculation: (i) OptNonSpeculative, with no speculation; (ii) OptHybridSpeculative, with local speculation; and (iii) OptAllSpeculative, with almost full speculation.

Experimental setup. Six different 8x8 MoT networks are implemented using FreeFDK NanGate 45 nm technology. Designs are technology-mapped and pre-layout. Six types of nodes are implemented, as building blocks: five fanout and one fanin. Nodes are mapped to the NanGate standard cell library in the Cadence Virtuoso tool. Accurate gate-level models are extracted using the Spectre simulator (typical process corner), to determine rise/fall times for every I/O path of each gate. Channel lengths and delays are borrowed from a synchronous MoT chip [21] and scaled to 45 nm technology. These extracted models of nodes and channels are used to implement the networks in structural Verilog.

An asynchronous NoC simulator is used for both unicast and multicast traffic. It includes a Programming Language Interface (PLI) to connect a C-based traffic generator and test environment to the technology-mapped network. A fixed packet size of 5 flits is used. Injection of headers of different packets follows an exponential distribution. A procedure similar to [2] is followed to ensure long warmup and measurement phases. Two steps are used to measure power: (i) record and annotate precise switching activity of every wire in the network over a benchmark run, and (ii) compute total power using the Synopsys PrimeTime tool.

Benchmarks. Experiments are conducted on six synthetic benchmarks. There are 3 unicast benchmarks [2]: 1) Uniform random, 2) Bit permutation:shuffle, and 3) Hotspot. There are 3 multicast benchmarks: 4) Multicast5 and 5) Multicast10, where all sources inject multicast traffic at rates of 5% and 10%, respectively, to random subsets of destinations, and otherwise do uniform random unicast, and 6) Multicast\_static, where 3 sources perform only random multicast, and the others do only uniform random unicast.

5.2 Node- and Network-Level Results

(a) Node-level results. Area and latency of the four new fanout nodes (Section 4) and Baseline fanout were evaluated. The unoptimized speculative nodes, due to their simplicity, have significantly lower area and latency (247 \(\mu\)m\(^2\), 52 ps) than Baseline (342 \(\mu\)m\(^2\), 263 ps). The more complex unoptimized non-speculative nodes have only small overhead (406 \(\mu\)m\(^2\), 299 ps) over Baseline. The optimized speculative nodes have moderate cost increases (373 \(\mu\)m\(^2\), 120 ps) over unoptimized, but will provide substantial network-level power savings (see below). Interestingly, the optimized non-speculative nodes have slightly lower costs (366 \(\mu\)m\(^2\), 279 ps) than the unoptimized ones.

(b) Contribution trajectory. This first case study explores the incremental impact of each key contribution: parallel multicast, local speculation, and optimizations.

Network latency. Figure 6(a) shows the average network latency results. We measure latency of each network at 25% of the saturation throughput of that network, up to the arrival of all headers at destinations. This load is high enough to show the impact of different benchmarks, while keeping the network largely uncongested. Moreover, long warmup and measurement times are used, for example, for Uniform Random/Multicast\_static benchmarks, warmup is 320 ns/640 ns, and measurement is 3200 ns/6400 ns with injection of 2100/4000 fits at each active source.

For multicast benchmarks, the simple tree-based parallel multicast network, BasicNonSpeculative, obtained significant benefits over the serial Baseline, from 39.1% (Multicast5) to 74.1% (Multicast\_static), highlighting the severe overheads of the serial multicast approach. The BasicHybridSpeculative and OptHybridSpeculative show further improvements of 10.5-14.9% and 17.5-21.4%, respectively, over the BasicNonSpeculative, illustrating the individual benefits of hybrid design and optimizations.

For unicast benchmarks, BasicNonSpeculative inculcates a small latency overhead over Baseline: since unicast is serial, the added node complexity to support parallel multicast becomes an overhead. However, the two hybrid networks provide noticeable benefits over BasicNonSpeculative, following similar trends as observed with multicast benchmarks. Interestingly, these latter results show that local speculation can significantly accelerate unicast traffic due to very fast speculative nodes.

Saturation throughput. Table 1 shows saturation throughput results. For multicast benchmarks, the new simple parallel network, BasicNonSpeculative, shows considerable benefits over the serial Baseline, ranging from 14.8% (Multicast5) to 39.5% (Multicast\_static). The two hybrid networks exhibit additional improvements up to 9.5% and 19.7%, respectively, over BasicNonSpeculative, demonstrating that local speculation, with accelerated packet transmission, provides a higher threshold for saturation.

For unicast benchmarks, results are more complex. Hotspots is highly-adversarial, with identical throughput for every network. For Uniform random, the OptHybridSpeculative network showed substantial improvements (28.0%) over BasicNonSpeculative. For Shuffle, two new networks show moderate throughput degradation (BasicNonSpeculative, BasicHybridSpeculative) over the Baseline, while OptHybridSpeculative obtains 32.8% higher throughput than BasicNonSpeculative and 9.5% higher than Baseline.

Total network power. Table 1 shows power results for 4 benchmarks. An injection rate that is 25% saturation load measured in Baseline, for a normalized comparison of energy per packet. Overall, as expected, Baseline has the lowest power due to its low complexity and serial multicast approach. BasicNonSpeculative has moderate overhead over Baseline (5.8-11.9%), due to more complex nodes. The overhead increases significantly for BasicHybridSpeculative.
(13.4-23.8% over Baseline), due to creation of redundant speculative copies. However, using OptHybridSpeculative, most of this overhead is removed (only 2.9-10.3% over Baseline): due to elimination of all redundant body flits (speculative nodes), and reduced switching activity because of channel pre-allocation (non-speculative nodes).

(c) Architectural design space exploration. This second case study only includes region, rather than optimize individual designs, while varying the degree of speculation.

Network latency. As shown in Figure 6(b), the hybrid network with local speculation (OptHybridSpeculative) achieves 9.7-11.9% latency improvements (unicast and multicast) over OptNonSpeculative, showing the effectiveness of the proposed techniques. The extreme case, OptAllSpeculative, exhibits 8.7-12.0% additional latency improvements over OptHybridSpeculative (18.5-21.7% over OptNonSpeculative), due to its almost fully speculative architecture (but will have significant power overheads).

Saturation throughput. For all benchmarks, in Table 1, the hybrid approach (OptHybridSpeculative) and extreme speculation (OptAllSpeculative) have nearly identical throughput to the non-speculative (OptNonSpeculative).

Total network power. Interestingly, even with its significant performance benefits, the optimized hybrid approach inures only minor power overheads of 3.5-6.1% over the non-speculative approach, since redundant copies are restricted to small local regions, and a power-oriented optimization is applied to disable speculation for body flits. In contrast, the fully-speculative approach (OptAllSpeculative) inures considerable power overheads (10.8-15.8% over OptHybridSpeculative, 14.7-22.9% over OptNonSpeculative) due to larger regions of speculation in OptAllSpeculative. It will have with larger MoT networks, this overhead will only increase, due to wider speculative regions.

(d) Addressing scheme comparisons. As highlighted earlier, an additional benefit of local speculation is to reduce the address field size. The serial Baseline has the shortest address field, using source routing, with a 1-bit address per fanout node on a unicast path: 3 bits for 8x8 MoT, and 4 bits for 16x16 MoT (not evaluated in this paper). However, the large performance overheads of these designs make them impractical for multicast.

Of the proposed parallel architectures, each using source routing, address field sizes for an 8x8 MoT are: 14 bits in non-speculative, 12 bits in hybrid, and 8 bits in almost fully-speculative. For a 16x16 MoT, the benefits of speculation are even greater: 30, 20 and 16 bits, respectively. Effectively, the speculative architectures reduce the total number of address fields, by only addressing non-speculative nodes.

6. CONCLUSIONS AND FUTURE WORK

The paper presents a lightweight multicast using Mesh-of-Trees based asynchronous NoCs. A new strategy, local speculation, is introduced, where fixed speculative switches always broadcast, but redundant packets are restricted to small regions. A hybrid network architecture is proposed, mixing speculative and non speculative switches. For multicast, the network achieves 17.8-21.4% improvements in network latency with small power reductions over a optimized non speculation approach. The approach is the first general-purpose multicast for asynchronous NoCs. For future work, we plan to extend the approach to larger MoT networks, alternative topologies (e.g. 2D-mesh), as well as synchronous NoCs.

7. REFERENCES


