ATAC: A 1000-Core Cache Coherent Processor with On-Chip Optical Network

by

George Kurian

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2010

Author .....
Department of Electrical Engineering and Computer Science
September 3, 2010

Certified by .....
Anant Agarwal
Professor
Thesis Supervisor

Accepted by .....
Terry P. Orlando
Chairman, Department Committee on Graduate Theses

ATAC: A 1000-Core Cache Coherent Processor with On-Chip Optical Network

by

George Kurian

Submitted to the Department of Electrical Engineering and Computer Science
on September 3, 2010, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

Based on current trends, multicore processors will have 1000 cores or more within the next decade. However, their promise of increased performance will only be realized if their inherent scaling and programming challenges are overcome. Fortunately, recent advances in nanophotonic device manufacturing are making CMOS-integrated optics a reality—interconnect technology which can provide significantly more bandwidth at lower power than conventional electrical signaling. Optical interconnect has the potential to enable massive scaling and preserve familiar programming models in future multicore chips.

This thesis presents ATAC [13], a new multicore architecture with integrated optics, and ACKwise, a novel cache coherence protocol designed to leverage ATAC's strengths. ATAC uses nanophotonic technology to implement a fast, efficient global broadcast network which helps address a number of the challenges that future multicores will face. ACKwise is a new directory-based cache coherence protocol that uses this broadcast mechanism to provide high performance and scalability. Based on 64-core and 1024-core simulations with Graphite [20] using Splash2, Parsec, and synthetic benchmarks, we show that ATAC with ACKwise out-performs a chip with conventional interconnect and cache coherence protocols. On 1024-core evaluations, ACKwise protocol on ATAC outperforms the best conventional cache coherence protocol on an electrical mesh network by 78% with Splash2 benchmarks and by 61% with synthetic benchmarks. Energy simulations show that the energy consumption of the ATAC network that assumes aggressive optical technology predictions is 2.24x lower than that of an electrical mesh network. However, with conservative optical technology predictions, the energy consumption of the ATAC network is 1.51x higher than that of an electrical mesh network.

Thesis Supervisor: Anant Agarwal
Title: Professor

Acknowledgments

I would like to thank Prof. Anant Agarwal for his enthusiatic support towards my research and all the technical/presentation skills I have learnt from him. I would like to thank Jason Miller for all the discussions I had with him and for his willingness to explain technical details clearly, precisely and patiently to me. I would also like to thank Nathan Beckmann, Harshad Kasture and Charles Gruenwald for all the interesting discussions I had during the Graphite project. I thank Hank Hoffmann, James Psota, David Wentzlaff, Lamia Youseff, Jason Ansel, Cree Bruins and Marco Santambrogio for all the many ways they have contributed to my learning process at MIT. I cannot stop thanking my parents and my brother for everything they have been to me thoughout my life and for making me the person I am today.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The trend in modern microprocessor architectures is clear: multicore is here. As silicon resources become increasingly abundant, processor designers are able to place more and more cores on a chip with massive multicore chips on the horizon. Many industry pundits have predicted manycores with 1000 or more cores by the middle of the next decade. But will current processor architectures (especially their interconnection mechanisms) scale to thousands of cores, and will such systems be tractable to program? This thesis argues that current multicore architectures will be challenged to scale to thousands of cores and introduces ATAC (pronounced ā-tack), a new processor architecture that addresses these issues. ATAC integrates an on-chip optical broadcast communication network within a mesh based tiled multicore architecture to significantly improve the performance, energy scalability, and ease of programmability of multicore processors [19, 18].

Although Moore's Law enables increasing numbers of cores on a single chip, the extent to which they can be used to improve performance is limited both by the cost of communication among the cores and off-chip memory bandwidth. Although our research is investigating the application of optical technology to both problems, this thesis focuses on the on-chip interconnect challenge. As computation is spread across multiple cores on a chip, distribution of instructions to the cores, and communication of intermediate values between cores account for an increasing fraction of execution time due to both latency and contention for communication resources. The outlook is particularly dismal for applications that require a lot of global communication operations (*e.g.,* broadcasts to maintain

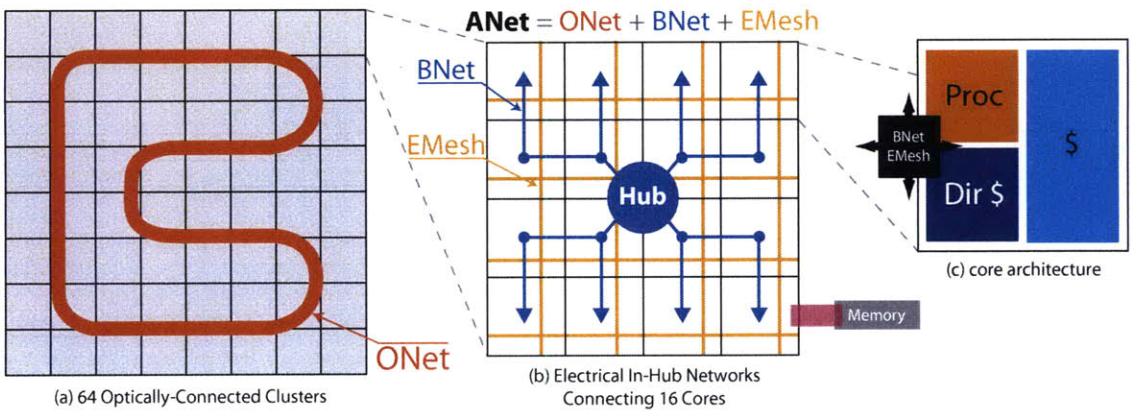

Figure 1-1: ATAC architecture overview

cache coherence) because each such operation ties up many resources and consumes a lot of energy.

State-of-the-art multicore chips employ one of two strategies to deal with interconnection costs. Small-scale multicores typically use a bus to interconnect cores. This simple design does not scale to large numbers of cores due to increasing bus wire length and contention. Another strategy is to use point-to-point networks such as the ring employed by the Cell processor [22]. This avoids long global wires but has the drawback that communication between distant cores requires multiple routing hops and overlapping messages experience significant contention and latency at large numbers of cores. Another such point-to-point network is the mesh employed by the Raw Microprocessor [25]. Although the mesh alleviates some of the bandwidth bottlenecks associated with the ring, it still suffers from contention with high traffic workloads.

Aside from interconnect scalability challenges, multicore architectures also face programming challenges. Programmers must spatially and temporally orchestrate computation and communication to extract high performance from the hardware. Even a simple function like broadcasting common data to all cores is difficult to perform efficiently. Broadcast and all-to-all communication operations in popular coherence and synchronization protocols present even greater challenges.

The ATAC processor architecture addresses these issues using on-chip optical communications technologies. Optical communications technologies have made tremendous

strides toward integrating optoelectronic components with standard CMOS fabrication processes. ATAC leverages these advances to eliminate communication contention using Wavelength Division Multiplexing (WDM). WDM allows a single optical waveguide to simultaneously carry multiple independent signals on different wavelengths. For example, a single waveguide with the same switching speed as its electrical counterpart and with 64 WDM channels would match the bandwidth of a 64-bit electrical bus. Optical waveguides, however, can also transmit data at higher speeds than electrical wires (a function of the index of refraction of the waveguide material for the optical signal; and a function of the RC delays, the dielectric material (SiO_2) surrounding the wires, and the delay of required repeaters for the electrical signal). This virtually eliminates the need for multiple hops between cores and the resulting contention at large scales. Optical signaling can also use less power (especially compared to long wires) because optical waveguides have relatively low loss and therefore do not require periodic repeaters or high-power drivers.

The ATAC processor is a tiled multicore processor augmented with an optical broadcast network. Each tile is interconnected electrically to its neighbors by a mesh network and optically through a global network that is low-latency and contention-free. The optical network consists of a set of optical waveguides that snake through the chip making a continuous loop as shown in Figure 1-1(a). Optical Hubs transmit data by modulating a laser light source and injecting it into the loop. The signal quickly propagates around the loop and can be received by all of the other Hubs in a single operation. Thus every message on the optical network has the potential to be an efficient global broadcast. Filtering at the receiving Hubs is used to limit the scope of the message for multicast or unicast messages.

ATAC's optical network is designed to provide the programming benefits of a bus interconnect while mitigating the scalability drawbacks. Like a bus, the optical network supports broadcast and provides uniform latency between network endpoints – two important properties for programming simplicity. Unlike a bus, it allows multiple senders to communicate simultaneously and without contention and is scalable to thousands of cores. Optical networks in Corona [11] and other works are tailored for point-to-point messages which do not confer these advantages.

The key contributions of this thesis are:

1. Proposes the novel ATAC opto-electronic network that leverages the recent advances in nanophotonic technology to solve the performance and energy problems that future multicore architectures will face.

2. Proposes ACKwise, a novel directory-based cache coherence protocol that leverages the broadcast capability of the ATAC network to provide efficient cache coherence across 1024 cores along with a programming environment where widespread sharing of data isn't frowned upon.

3. Evaluates the performance of the ATAC network and the ACKwise protocol against an electrical mesh network and conventional directory-based cache coherence protocols using Splash2, Parsec and synthetic benchmarks on 64-core and 1024-core processors.

4. Evaluates the energy consumption of the ATAC network using aggressive and conservative photonic technology predictions and compares that against an electrical mesh network.

The remainder of the thesis is organized as follows. Chapter 2 gives nanophotonics background, focusing on physical constraints imposed on the ATAC architecture. Chapter 3 provides an overview of the ATAC architecture, including its processing, communication, and memory mechanisms. Chapter 4 introduces the ACKwise cache coherence protocol. Chapter 5 describes in detail the power modeling methodology for the ATAC network and the electrical mesh network, stating clearly all the assumptions made regarding the underlying optical and electrical technology. Chapter 6 evaluates the ATAC architecture using the ACKwise protocol and provides a preliminary set of results using Splash2, Parsec and synthetic benchmarks focusing on how ATAC enables high performance cache coherence across 64 and 1024 cores. Chapter 6 also includes a energy analysis comparing the ATAC network against an electrical mesh network. Chapter 7 follows with a detailed discussion of related work, and Chapter 8 concludes the thesis.

Chapter 2

Optical Devices Background

Advances in electronic-photonic integration have enabled optical interconnect technologies with greater integration, smaller distances, and higher bandwidths [23], [24], [17], [30]. Further, recent research [21] has shown that optical devices can be built using standard CMOS processes, thereby allowing optics to replace global wires and on-chip buses [1]. The integration of photonic interconnects into chips has the potential to address some of the greatest challenges facing future large-scale multicore processors.

This section presents a brief overview of these CMOS compatible devices and their constraints. The key elements in a nanophotonic network such as the one employed by the ATAC chip include: the "optical power supply" light source; waveguides to carry optical signals; filters and modulators to place signals into the waveguides; and detectors to receive signals from the waveguides. This section discusses each of these components and describes the complete path for transmitting data optically.

In ATAC the light source, or "optical power supply", is generated by off-chip lasers and coupled into an on-chip waveguide. On-chip light sources exist, but consume large quantities of precious on-chip power and area. The power consumption of an off-chip laser is roughly 1.5 W, with 0.2 W of optical power ending up in the on-chip waveguide.

Waveguides are the on-chip channels for light transmission. They guide and confine light by a combination of a high-refractive-index material on the inside of the waveguide and a low-refractive-index material on the outside (the cladding). Waveguides can be made out of either silicon (Si) or polymer. Due to the fact that Si waveguides can be packed onto

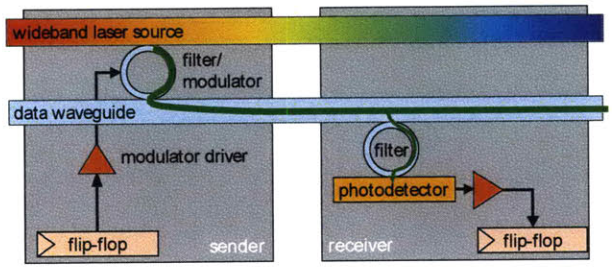

Figure 2-1: Optical transmission of a single bit

a chip at much higher densities and that modulators for Si can be made much more compactly, the ATAC design employs Si waveguides. These waveguides can be manufactured in a standard CMOS process, as both the waveguide and cladding materials are commonly used elsewhere. ATAC requires waveguides with losses of less than 0.3dB/cm and total power capacity of about 10 mW, both of which are achievable with Si.

The optical filter is a ring resonator that couples only a specific wavelength from the power supply waveguide to the data waveguide. The exact wavelength, as well as the spacing between wavelengths, is determined by the ring resonator dimensions and is fixed during manufacturing. Limited tuning can be achieved by changing the ring's temperature or by injecting charge into the ring. The modulator is an optical device that imprints a digital signal on the light extracted by the filter by varying the absorption in the device. Modulators are used to translate an electrical signal (amplified by the modulator driver) into an optical signal, and can therefore be thought of as an "optical switch", placing values onto optical waveguides. The modulators are Ge based electro-absorption modulators with integrated filters. The ring resonators are not used for modulation but just for wavelength filtering. It is assumed that athermal design [33] is implemented, so that the rings do not need to be tuned. The modulators used in the ATAC design have characteristics that are expected to be reached by designs available in 2012: insertion loss of 1dB; area less than 50 μm^2; modulation rate of 1 Gbps; energy required to switch less than 25 fJ; and power consumption of 25 μW at 1 GHz [16].

At the receiving end of a waveguide, additional components are used to receive the signal and convert it to an electrical signal. An additional optical filter is used to extract

light of a particular wavelength from the data waveguide and transfer it to a photodetector. The filter can be designed to extract any fraction of the total signal by adjusting the size of the gap between the waveguide and the filter. The photodetector is an extremely sensitive optical device which absorbs photons and outputs an electrical signal. The photodetector proposed for ATAC has a responsivity of greater than 1 Amp/Watt and 3dB bandwidth performance at 1 GHz. It has an area footprint of less than 20 μm^2. Furthermore, the expected capacitance of the photodetector is less than 1 fF [8]. In current technology nodes, the output of the photodetector would need to be amplified by a power-hungry TIA (transimpedance amplifier) before it could be used to drive a digital circuit. However, starting with the 22nm node, the smaller transistor input capacitances will allow the photodetector to directly drive a digital circuit, greatly reducing power consumption.

Figure 2-1 puts all of these elements together, showing how one bit is transmitted from a flip-flop of one core to a flip-flop of another core. In this figure, the core on the left shows the components relevant to sending and the core on the right shows the components relevant to receiving; however, in the actual chip all cores would contain both sets of components. From end to end, the process for sending a bit on the ATAC's optical network is as follows. The flip-flop signals the modulator driver to send either a 0 or a 1. The modulator driver, which consists of a series of inverter stages, drives the modulator's capacitive load. The modulator couples light at its pre-tuned wavelength λ_i from the optical power source and encodes either a 0 or 1 onto the data waveguide. The optically-encoded data signal traverses the waveguide at approximately one-third the speed of light and is detected by a filter that is also tuned to wavelength λ_i. Photons are detected by the photodetector and received by a flip-flop on the receiver side.

Chapter 3

Architecture Overview

As previously illustrated in Figure 1-1, the ATAC processor uses a tiled multicore architecture combining the best of current scalable electrical interconnects with cutting-edge on-chip optical communication networks. The ATAC architecture is targeted at 1000-core chips implemented in a 16nm process. However, it can also be scaled down to smaller chips. In this thesis we describe and evaluate 64- and 1024-core versions. We first review the baseline electrical architecture, and then describe how it is augmented with the optical interconnect.

The underlying electrical architecture consists of a 2-D array of processing cores connected by a point-to-point, packet-switched mesh network (called the *EMesh*) like those seen in other multicore processors [25, 14, 12]. Each core in ATAC contains a single- or dual-issue, in-order RISC pipeline with data and instruction caches (Figure 1-1(c)). ATAC uses a novel directory-based cache coherence scheme with a portion of the directory in each core (see Chapter 4).

To this electrical baseline, we add a global optical interconnect—the *ANet*—based on state-of-the-art optical technology. Whereas the EMesh is ideal for predictable, short-range point-to-point communication, the ANet provides low-latency, energy-efficient global and long-distance communication. The key component of the ANet is the all-optical ONet shown in Figure 1-1(a). In the 1024-core ATAC architecture (called ANet^{1024}), cores are grouped into 64 "clusters", each containing 16 cores. Each cluster contains a single ONet endpoint called a *Hub*. The Hub is responsible for interfacing between the optical compo-

nents of the ONet and the electrical components within a cluster. The ATAC architecture can be scaled down by reducing the number of cores with each cluster. A 64-core chip (called ANet^{64}) would connect each core directly to a Hub.

In ANet^{1024}, individual cores are connected to the Hub in two ways: data going from a core to the Hub uses the standard EMesh (described above); data going from the Hub to the cores uses the BNet, a small electrical broadcast network (Figure 1-1(b)). In the 22nm node, the clusters are small enough that a flit can travel from the Hub to all cores in a cluster within one clock cycle. Because the BNet is dedicated to broadcasts, it is essentially a fanout tree and requires no routers, crossbars, or internal buffering. It requires only a small amount of buffering and arbitration at the Hub and receiving buffers at the leaves. We estimate that a BNet requires less than one-eighth the area of a full EMesh of the same bitwidth.

The ANet^{1024} uses a 128-bit wide ONet (128 optical waveguides for data); one 128-bit wide electrical EMesh; and two parallel 128-bit wide BNets. The Hub arbitrates between the two BNets using a static policy: packets sent from clusters with even number IDs on the first BNet and odd number IDs on the second BNet. Together, the ONet, EMesh and BNet form the complete ANet^{1024}.

3.1 ONet Optical Network

The key to efficient global communication in a large ATAC chip is the optical ONet. The ONet provides a low-latency, contention-free connection between a set of optical endpoints called Hubs. Hubs are interconnected via waveguides that visit every Hub and loop around on themselves to form continuous rings (Figure 1-1(a)). Each Hub can place data onto the waveguides using an optical modulator and receive data from the other Hubs using optical filters and photodetectors. Because the data waveguides form a loop, a signal sent from any Hub will quickly reach all of the other Hubs. Each Hub's filters are tuned to extract approximately 1/64th of the signal, allowing the rest to pass on to the downstream Hubs. Thus every transmission on the ONet is actually a fast, efficient broadcast.

The ONet uses wavelength division multiplexing (WDM) to circumvent contention.

Each Hub has modulators tuned to a unique wavelength to use when sending and contains filters that allow it to receive signals on all of the other wavelengths. This eliminates the need for arbitration in the optical network. Taken together, these features mean that the ONet is functionally similar to a broadcast bus, but without any bus contention.

WDM is a key differentiator of the ATAC architecture from a performance scalability perspective. WDM allows a single waveguide to simultaneously carry bits of many overlapping communications. To contrast, an electrical wire typically carries a single bit. Whereas ATAC may share a single waveguide medium between a large number of simultaneous communication channels, implementing multiple simultaneous communication channels in the electrical domain requires additional physical wires. For network operations that are expensive to implement in the electrical domain (such as broadcast), the ATAC approach greatly improves efficiency.

The broadcast mechanism of the ATAC architecture is another key differentiator. Optical technology provides a way to build fast, efficient broadcast networks whereas electrical mechanisms do not. When using optical components instead of electrical components, signals may travel farther and be tapped into by more receivers before they need be regenerated. With electrical components, regeneration is accomplished via buffers or sizing-up of transistors for increased drive strength. When these electrical mechanisms are extensively employed, as they would be in a large electrical broadcast network, it leads to high energy consumption and poor scaling.

Besides broadcasts, optical technology also allows efficient long-distance point-to-point communication. Initiating an optical signal (*i.e.,* switching the modulator) requires more energy than switching a short electrical wire. However, once generated, the optical signal can quickly travel anywhere on the chip without the need for repeaters. To avoid wasting power and resources delivering these unicast messages to all cores, ATAC includes filtering at the receiving Hubs and cores. Packets labeled as intended for a single core are only rebroadcast on the BNet of the cluster containing that core. In addition, the other cores in that cluster will drop the packet immediately, rather than processing it.

The ATAC architecture was carefully designed taking into account the physical limitations and constraints of both the optical (see Chapter 2) and electronic devices. Based

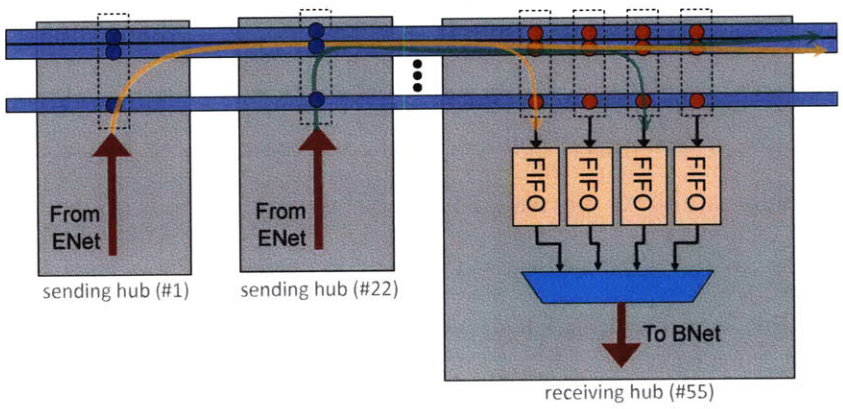

Figure 3-1: Hub-to-hub communication over the ONet

on these constraints, the ONet as described above should scale to at least 64 (and possibly as many as 100) Hubs. This limit is based on several factors: 1) the total range of wavelengths over which the optical devices can be tuned divided by the minimum spacing between wavelengths, 2) the total amount of optical power a waveguide can carry divided by the minimum amount that each photodetector needs to receive to reliably register a signal, and 3) the maximum length of a waveguide based on the acceptable propagation losses.

These limits can be overcome using multiple waveguides and dividing the communication channels between them. However, eventually the area needed for the optical components will become the limiting factor. The ONet's optical components and photonic interconnect can be placed on a separate layer in the CMOS stack, and can therefore overlap the electrical components to which they connect. However, for a 400 mm^2 chip, the entire area would be consumed by an ONet with approximately 384 Hubs. Since we believe that chips will eventually grow to thousands of cores, some sharing of Hubs will certainly be needed. Therefore, for the purposes of this thesis, we take the simple approach and assume that the ONet is limited to 64 Hubs.

Sending data using the ONet is shown in more detail in Figure 3-1. To provide adequate on-chip bandwidth, the ONet uses a bundle of waveguides, each waveguide containing 64 wavelengths. The ONet contains 128 waveguides for data, one for backwards flow control, and several for metadata. The metadata waveguides are used to indicate a message

type (*e.g.,* memory read, barrier, raw data) or a message tag (for disambiguating multiple messages from the same sender). ATAC, thus uses multiple waveguides to increase the bandwidth of its optical network (ONet) and keeps the network operating frequency at the frequency of the core (1 GHz). Hence, ATAC does not require any sophisticated clock synchronization schemes for the optical network. Each of the 64 Hubs transmits data on its unique wavelength without interference. Figure 3-1 shows two Hubs sending their data simultaneously without interference. The receiving Hub captures both of the values simultaneously into sender-Hub-specific FIFOs. These values are then propagated to the cores using the BNet.

3.2 Cache Subsystem

The data caches across all cores on the ATAC chip are kept coherent using a directory-based coherence protocol called *ACKwise* described in more detail in Chapter 4. The directory is distributed evenly across the cores. The directory in each core is stored in SRAM memory and is organized as a directory cache. Furthermore, each core is the "home" for a set of addresses (the allocation policy of addresses to homes is statically defined).

3.3 External Memory Subsystem

When cores need to communicate with external memory, they do so via several on-chip memory controllers. Each cluster has one core replaced by a memory controller. After receiving requests through the ANet, the memory controller communicates with external DRAM modules through I/O pins. Replies are then sent back to the processing cores through the ANet. Other ATAC chips with different memory bandwidths are possible by varying the number of cores replaced by memory controllers.

The primary task of the memory controller is to translate requests from the processing cores into transactions on a memory I/O bus. The choice of I/O bus technology is independent of the on-chip network architecture since the memory controller is performing a translation. However, to support the large number of memory controllers needed for a

1000-core chip, we assume that the connection to memory is optical as well.

A detailed design for an optical memory subsystem is left to future work. However, we can assume that an optical memory bus would consist of some number of on-chip waveguides that are coupled to external fibers, effectively creating optical "pins." Each optical pin could carry up to 64 wavelengths of light at speeds of up to 20 GHz. The actual transmission speed would likely be limited by design trade-offs in the electrical circuits driving the optical components. We estimate that optical I/O pins operating at 5 GHz (yielding 40 GB/s of bandwidth) should be practical. Thus each off-chip memory bus can be implemented using a single optical pin. This makes it practical to integrate the large number of memory controllers needed to meet the bandwidth needs of future 1000-core chips.

Chapter 4

Cache Coherence Protocol

This chapter presents *ACKwise* [13], a novel cache coherence protocol derived from a MOESI-directory based protocol [28]. Each directory entry in this protocol, as shown in Figure 4-1 is similar to one used in a limited directory scheme [2] but with a few modifications. The 3 fields in each directory entry are as follows: (1) **State**: This field specifies the state of the cached block(s) associated with this directory entry (one of the MOESI states); (2) **Global(G)**: This field states whether the number of sharers for the cache block exceeds the capacity of the sharer list. If so, a broadcast is needed to invalidate all the cached blocks corresponding to this address when a cache demands exclusive ownership; (3) $\textbf{Sharer}_1 - \textbf{Sharer}_k$: These fields represent the sharer list. The ACKwise protocol which holds the identities of a maximum of k sharers is denoted as ACKwise_k.

When the number of sharers exceeds k, the *global(G)* bit is set so that any number of sharers beyond this point can be accommodated. Once the global(G) bit is set, the $Sharer_k$ field holds the total number of sharers. The $Sharer_1 - Sharer_{k-1}$ fields still hold the identity of $k - 1$ distinct sharers.

Figure 4-1: Structure of an ACKwise_k directory entry

4.1 Operation of the ACKwise$_k$ Protocol

When a request for a shared copy of a cache block is issued, the directory controller first checks the state of the cache block in the directory cache. (a) If the state is *Invalid(I)*, it forwards the request to the memory controller. The memory controller fetches the cache block from memory and sends it directly to the requester. It also sends an acknowledgement to the directory. The directory changes the state of the cache block to *Exclusive(E)* and sets the $Sharer_1$ field to the ID of the requester. (b) If the state is one of the valid states (*i.e.,* one of *MOES*), it forwards the request to one of the sharers. The sharer forwards the cache block directly to the requester and sends an acknowledgement to the directory. Appropriate state changes happen in the directory according to the rules of the *MOESI* protocol [28]. The directory controller also tries to add the ID of the requester to the sharer list. This is straightforward if the *global(G)* bit is clear and the sharer list has vacant spots. If *global(G)* bit is clear but the sharer list is full, it sets the *global(G)* bit and stores the total number of sharers (in this case, $k + 1$) in the $Sharer_k$ field. If the *global(G)* bit is already set, then it increments the number of sharers by one.

When a request for an exclusive copy of a cache block is issued, the directory controller first checks the state of the cache block in the directory cache. (a) If the state is *Invalid(I)*, the sequence of actions followed is the same as that above except that the state of the cache block in the directory is set to *Modified(M)* instead of *Exclusive(E)*. (b) If the state is one of the valid states (*i.e.,* one of *MOES*), then the directory controller performs the following 2 actions: (i) It forwards the request to one of the sharers. (ii) If the global bit is clear, it sends unicast invalidation messages to each core in the sharer list. Else, if the global bit is set, it broadcasts an invalidation message (to all the cores in the system). Now, the sharer which receives the forwarded request sends the cache block directly to the requester, invalidates the block and acknowledges the directory. The other sharers invalidate their cache blocks and acknowledge the directory. The directory controller expects as many acknowledgements as the number of sharers (encoded in the $Sharer_k$ field if the *global(G)* bit is set and calculated directly if the *global(G)* bit is clear). After all the acknowledgements are received, the directory controller sets the state of the cache block to *Modified(M)*,

the *global(G)* bit to 0 and the $Sharer_1$ field to the ID of the requester. Due to the broadcast capabilities of ATAC as described in Chapter 3, the sending of broadcast messages can be achieved easily. In addition, the ACKwise$_k$ protocol requires only as many unicast acknowledgements as a full-map directory-based protocol. Hence the name ACKwise since the protocol intelligently tracks the number of sharers of a cache block and requires acknowledgements from only the actual sharers on an invalidation broadcast message.

4.2 Silent Evictions

Silent evictions refer to the evictions of clean shared data from the cache of a core without notifying the directory that the core is no longer a sharer of the data. Silent evictions are supported in certain cache coherence protocols since they are found to increase system throughput by decreasing network utilization. Silent evictions are not supported in the ACKwise protocol since the directory should always have an accurate count of the number of sharers of a cache line for correct operation. However, our measurements show that disallowing silent evictions is not detrimental to the performance of the ACKwise protocol because: (1) the additional coherence messages do not lie on the critical path of load or store misses and hence do not directly affect the average memory latency and thereby processor performance; (2) these messages do not include data and hence contribute only a small percentage to the overall network traffic and thereby do not really affect the network latency.

Chapter 5

Power Modeling

This chapter discusses the power estimation techniques used to compare the ATAC network (ANet) against an electrical mesh network (EMesh). Section 5.1 talks about the power estimation techniques used for the electrical mesh network while section 5.2 details the power estimation techniques for the ATAC network, showing how aggressive and conservative optical technology predictions affect the energy consumption of its constituent optical network.

5.1 Electrical Mesh Network

The electrical mesh is composed of routers and links arranged in a mesh topology as shown in Figure 5-1. Communication between tile A and tile B is realized using multiple hops as shown in Figure 5-2. (Here, communication between tile A and tile B takes 4 hops). During each hop, the packet is processed by a router and traverses a link.

The packet processing stages of a router are as follows:

1. Buffer Write (if the downstream link is contended)
2. Switch Allocation
3. Buffer Read (if the downstream link is contended)
4. Crossbar Traversal

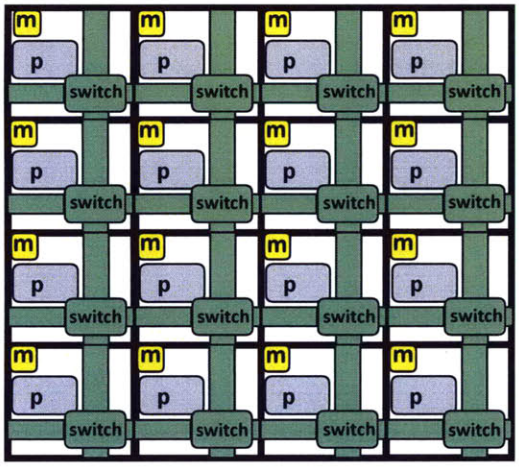

Figure 5-1: Electrical Mesh Network connecting 16 tiles

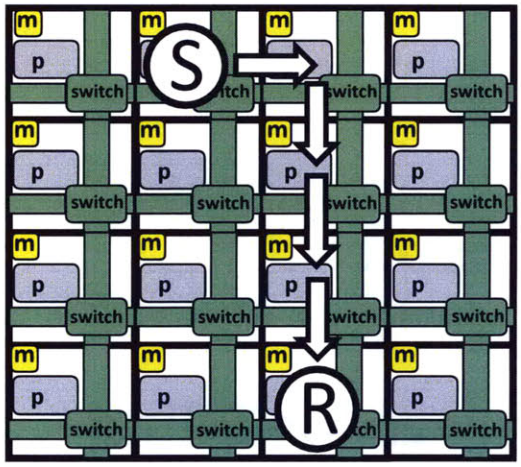

Figure 5-2: Routing of a Packet between tile A and tile B

The buffer write and read stages only occur if the downstream link is contended. The different components of a router, including the buffer, the switch allocator and the crossbar are shown in Figure 5-3. Static power and dynamic energy for the four processing stages of the router as well as for the link are computed using Orion [5]. Event counters are maintained during simulation that increment whenever an activity occurs (such as link traversal, buffer write, etc.) and at the end of simulation, the events counters are multiplied by the dynamic energy associated with each event (obtained from Orion citeorion) to get the total dynamic energy. Total static energy is computed by summing the static power associated with each component of the network and multiplying that number by the total simulation time. Static power and dynamic energy estimates for the router and link components of a 256-bit wide electrical mesh are shown in Table 5.1. Dynamic energy estimates are for a 1-flit (256-bit) packet assuming half the bits flip.

Electrical Mesh Component		Dynamic Energy	Static Power
256-bit 5×5 Router	Buffer (Read + Write)	22.05 pJ	21.98 mW
	Crossbar	59.34 pJ	32.32 mW
	Switch Allocator	0.44 pJ	0.41 mW
256-bit Link	Link	18 pJ	9.4 mW

Table 5.1: 256-bit wide Electrical Mesh Energy Parameters

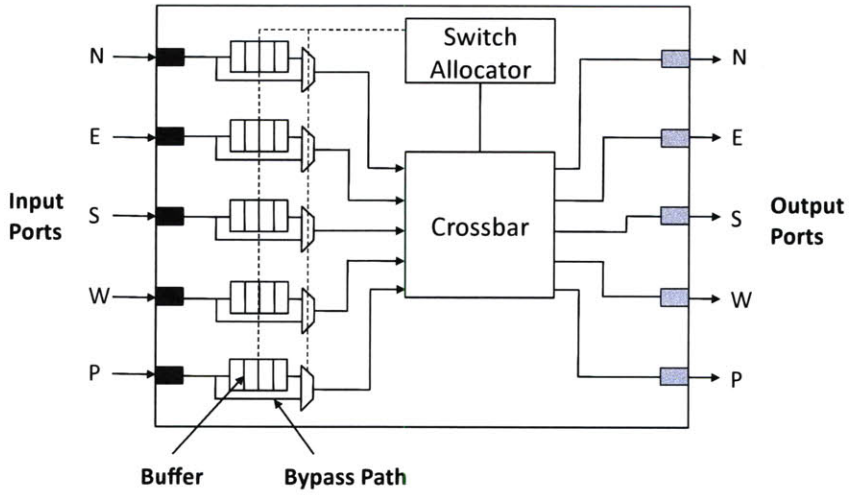

Figure 5-3: Electrical Mesh Router Microarchitecture

Energy/bit Calculation The average dynamic energy consumption per bit in an electrical mesh is given by the following equation:

$$
\begin{aligned}
&Average\ Dynamic\ Energy\ per\ bit \\
&= \frac{Dynamic\ Energy\ per\ Packet\ per\ Hop \times Average\ Number\ of\ Hops}{Packet\ Size \times Flit\ Size} \\
&= \frac{2\sqrt{N} \times (B \times (Dynamic_{Buffer} + Dynamic_{Crossbar} + Dynamic_{Link}) + Dynamic_{SA})}{3 \times B \times W}
\end{aligned}
$$

The average number of hops of a packet in an N-tile ($\sqrt{N} \times \sqrt{N}$) electrical mesh network is $\frac{2\sqrt{N}}{3}$. In the above equation, $B(=2)$ stands for the average number of flits in a packet; $W(=256)$ stands for the number of bits in a flit; $N(=1024)$ is the number of cores; and $Dynamic_{Component}$ is the dynamic energy associated with each network router or link component. Dynamic energy values are substituted from Table 5.1. Average dynamic energy per bit is computed to be 8.3 pJ (6.8 pJ from router and 1.5 pJ from link). If the packet traverses only k hops, the average dynamic energy per bit is $390 \times k$ fJ. The results are summarized in Table 5.2.

Modeling Inaccuracies All electrical network power estimates are obtained at 32 nm from Orion even though 1024 cores cannot be fabricated on a chip until the 22 nm technology node. Moreover, Orion's models are calibrated at 65 nm and then scaling factors are

EMesh Component	Dynamic Energy (per bit)	Dynamic Energy (per bit per hop)
Router	6.8 pJ	319 fJ
Link	1.5 pJ	71 fJ
Total	**8.3 pJ**	**390 fJ**

Table 5.2: Dynamic Energy per bit on a 256-bit wide Electrical Mesh Network.

applied to get power estimates at 32 nm.

5.2 ATAC Network

The building blocks of the ATAC network (ANet) include a modified electrical mesh network, an all-optical network (ONet) and a broadcast network (BNet). Denote the modified electrical mesh network as ENet for this discussion. There is one ENet, one ONet and $C \times 2$ BNets' in the ATAC network (C being the number of clusters). The topology of the ATAC network is shown in Figure 5-4.

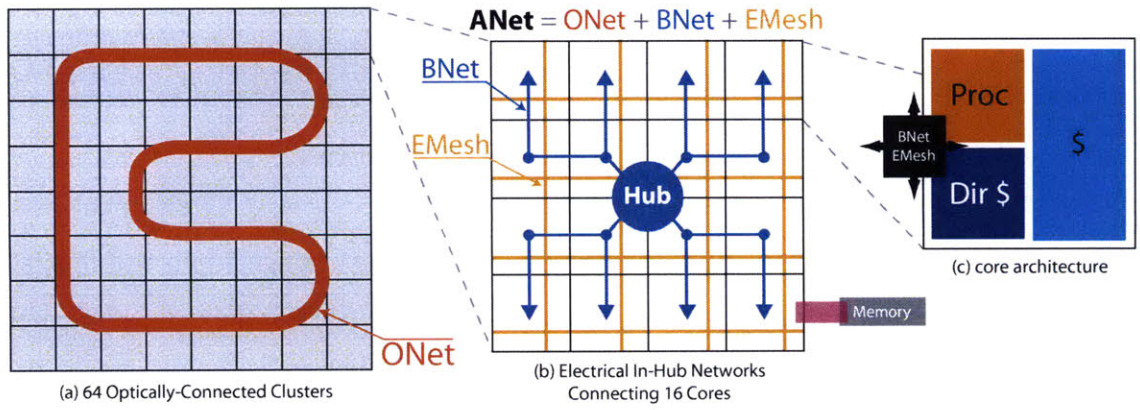

Figure 5-4: ATAC Network Architecture

5.2.1 ENet

The ENet can be modeled as an electrical mesh where two routers is each cluster have an additional output port to connect to the Hub. Since the bandwidth bottleneck is at the Hub and not in the ENet, this will suffice. (The ENet can sustain a throughput of 256 bits/cycle

to the Hub while the Hub can only process 128 bits/cycle). Dimension-order (X-Y) routing is followed to send packets from a tile to the Hub.

ENet Component		Dynamic Energy	Static Power
128-bit 5×5 Router	Buffer (Read + Write)	11.07 pJ	11.18 mW
	Crossbar	21.21 pJ	16.22 mW
	Switch Allocator	0.44 pJ	0.41 mW
128-bit Link	Link	9 pJ	4.7 mW

Table 5.3: 128-bit wide ENet (Modified Electrical Mesh) Energy Parameters

5.2.2 ONet

Optical Losses	Aggressive	Conservative
Waveguide Propagation Loss	2 dB (0.2 dB/cm)	10 dB (1 dB/cm)
Modulator Insertion Loss	1 dB	3 dB
Filter Set Through Loss	1e-2 dB	1 dB
Filter Drop Loss	1 dB	2 dB
Backplane Coupler Loss	1 dB	2 dB
Splitter Loss	1 dB	1 dB
Photodetector Loss	0.1 dB	1 dB
Total	**7 dB**	**20 dB**

Table 5.4: Optical Waveguide Losses

The optical network is composed of an off-chip laser source that produces photons, waveguides for the transmission of photons, modulators for imprinting a digital signal on the wavelengths and a photodetector for receiving the data. Refer to Chapter 2 for more details on the optical background. The power consumption of the all-optical ONet can be calculated by adding the static power of the laser and the rings and the dynamic power

Laser Power (per wavelength)	Aggressive	Conservative
Charge required to flip gate (Q_g)	1.88 fC	4.28 fC
Optical Losses	7 dB	20 dB
Frequency (f_c)	1 GHz	1 GHz
Responsivity of Photodetector (R_d)	1.1 A/W	1.1 A/W
Number of Detectors (N_R)	64	64
Laser Power (per wavelength)	**0.55 mW**	**29.4 mW**

Table 5.5: On-Chip Laser Power Calculation

Power Component	Aggressive	Conservative
Sender Dynamic Energy (Modulator + Modulator Driver)	10 fJ	45 fJ
Receiver Dynamic Energy (Photodetector + Receiver)	2.3 fJ	15 fJ
Laser Power (per wavelength)	0.55 mW	29.4 mW
Ring Tuning Power (per ring resonator)	0 mW	0.02 mW

Table 5.6: ONet Power Components

of the modulator, modulator driver, photodetector and photodetector receiver. The ONet power components are listed in Table 5.6.

Laser Power

The total laser power is dependent on the following five factors:

1. Optical Losses (L)
2. Operating Frequency (f_c)
3. Number of Receivers (N_R)
4. Responsivity of Photodetector (R_d)
5. Laser efficiency (E_l)

To calculate the optical laser power, first all the optical losses are enumerated. This is done in Table 5.4. Next, the capacitance of the receiver is calculated by summing the capacitance of the photodetector, the wire connecting the photodetector and the transistor and the gate capacitance of the transistor. At 11 nm, this capacitance is aggressively and conservatively estimated to be 2.35 fF and 5.35 fF respectively. With an operating voltage of 0.8 V, the aggressive and conservative estimates for the charge needed to flip a gate, Q_g are 1.88 fC and 4.28 fC respectively. The operating frequency, f_c is 1 GHz and the number of receivers per wavelength, N_R is 64. The responsivity of the photodetector, R_d is 1.1 A/W. The amount of optical laser power per wavelength per bit is given by the following equation.

$$P_{optical} = \frac{10^{\frac{L}{10}} \times Q_g \times f_c \times N_R}{R_d} \tag{5.1}$$

The electrical laser power per wavelength per bit is given by the following equation.

$$P_{electrical} = \frac{P_{optical}}{E_l} \tag{5.2}$$

The above calculations for computing laser power are shown in Table 5.5.

Ring Tuning Power

An athermal design of ring resonators has been proposed in [33]. However it is not clear whether such designs are easy to be built and are compatible with a monolithic silicon fabrication process. Hence, a ring tuning power of 0 mW is used as an aggressive estimate. The conservative estimate is calculated by assuming a ring tuning power of 1 μW for a 1 K temperature range. Assuming a normal operating temperature range of 20 K, this comes out to be 0.02 mW per ring resonator.

Modulator + Modulator Driver Power

Based on aggressive and conservative predictions of the capacitance of the devices, the dynamic energy is estimated to be 10 fJ and 45 fJ respectively.

Photodetector + Receiver Power

Based on aggressive and conservative projections of the capacitance of the devices, the dynamic energy is estimated to be 2.3 fJ and 15 fJ respectively.

5.2.3 BNet

The BNet connects the Hubs to the cores as discussed in Chapter 3. Since there are only 2 BNets per cluster, the data that arrives on the ONet compete for access to the BNet. The data that arrives from the different clusters is statically partitioned among the BNets according to the sender cluster's ID. A BNet is composed of the following two elements.

1. A $\frac{C}{2} \times 1$ router ($\frac{C}{2}$ inputs and 1 output) that arbitrates between the data received from $\frac{C}{2}$ clusters and forwards that data to the tiles.

2. A pipelined broadcast bus that transports data from the output port of the router to all the tiles in the cluster.

The $\frac{C}{2} \times 1$ router is modeled using Orion [5]. The dynamic energies due to the buffers, switch allocators and crossbar are faithfully modeled. The pipelined broadcast bus is modeled as a tree of links that connect the Hub to all the tiles in the cluster. There are $n + 1$ nodes in this tree (n tiles + 1 Hub). Hence, the tree is composed of n electrical links and the length of each electrical link is equal to the width of a tile. Each electrical link is again modeled using Orion [5].

BNet Component		Dynamic Energy	Static Power
128-bit $\frac{C}{2} \times 1$ Router	Buffer (Read + Write)	11.07 pJ	71.55 mW
	Crossbar	15.14 pJ	30.45 mW
	Switch Allocator	26.04 pJ	5.32 mW
128-bit Pipelined Broadcast Tree Link	Link	9 pJ	4.7 mW

Table 5.7: 128-bit wide BNet (Pipelined Broadcast Tree) Energy Parameters. C stands for the number of clusters ($C = 64$). n stands for the number of tiles in each cluster ($n = 16$).

Energy/bit Calculation To calculate energy/bit, the contributions of the three constituent networks (ENet, ONet and BNet) are considered.

For the ONet network, the average dynamic energy per bit is given by the following equation:

$$\begin{aligned} &Average\ Dynamic\ Energy\ per\ bit(ONet) \\ &= Dynamic\ Energy\ of\ Modulator + C \times Dynamic\ Energy\ of\ Receiver \end{aligned}$$

For the ONet network, the average static energy per bit is given by the following equation:

$$\begin{aligned} &Average\ Static\ Energy\ per\ bit(ONet) \\ &= \frac{Laser\ Power\ (per\ wavelength\ per\ bit) + C \times Ring\ Tuning\ Power}{Network\ Frequency} \end{aligned}$$

ANet Component	Dynamic Energy (per bit)	Static Energy (per bit)
ONet	157 fJ	550 fJ
ENet	480 fJ	-
BNet	1533 fJ	-
Total	**2.72 pJ**	

Table 5.8: Energy/bit computation for the ANet network assuming aggressive optical technology predictions.

For the ENet and BNet networks, the average dynamic energy per bit is calculated as illustrated in Section 5.1. The results are summarized in Table 5.8.

Here, 2.72 pJ is the average energy consumed to transmit a bit on the 128-bit wide ANet. 390 fJ is the energy required to transmit a bit across one hop in a 256-bit wide EMesh (see Table 5.2). Hence, transmitting a bit between two cores that are at least 7 hops apart is more energy-efficient on the ANet network.

Chapter 6

Evaluation

The purpose of this section is to: (1) Demonstrate the capabilities of the ATAC network (*ANet*) over a pure electrical mesh network (*EMesh*), (2) Demonstrate the performance advantages of using the $ACKwise_k$ protocol over the Dir_kB and Dir_kNB limited directory-based cache coherence protocols [2], and (3) Perform an energy comparison of the ATAC network assuming both aggressive and conservative optical technology predictions with an electrical mesh network.

Dir_kB is a limited directory based protocol which broadcasts once the capacity of the sharer list is exceeded and collects acknowledgements from all the cores in the system. $ACKwise_k$ on the other hand, intelligently tracks the number of sharers once the capacity of the sharer list is exceeded and needs acknowledgements from only the actual sharers of the data on a broadcasted invalidation. Dir_kNB always ensures that the number of sharers of a cache line is less than the capacity of the sharer list. k denotes the number of hardware sharers in each of the above protocols. This section evaluates the performance of Splash2 and Parsec benchmarks as well as synthetic applications on 64 and 1024 cores using six combinations of on-chip networks and cache coherence protocols: (a) *ANet-*$ACKwise_k$, (b) *ANet-*Dir_kB, (c) *ANet-*Dir_kNB, (d) *EMesh-*$ACKwise_k$, (e) *EMesh-*Dir_kB and (f) *EMesh-*Dir_kNB. Results demonstrate the advantages of using *ANet* over *EMesh* due to its higher bandwidth, lower latency and broadcast capabilities as well as the performance benefits of the $ACKwise_k$ protocol over the Dir_kB and Dir_kNB protocols.

Core Model	In-Order
Frequency	1 GHz
EMesh Hop Latency	2 cycles(router delay - 1, link traversal - 1)
ONet Hop Latency	3 cycles(E/O + O/E conversion - 2, link traversal - 1)
1024 cores	
$ANet^{1024}$ = ONet + ENet + 64x2 BNets	
ONet	128-bit wide
ENet	128-bit wide
BNet	128-bit wide
EMesh for comparison	256-bit wide
Memory Bandwidth	64 memory controllers 5 GBps per controller
L2 Cache Size	32 KB
64 cores	
$ANet^{64}$ = ONet + ENet	
ONet	64-bit wide
ENet	32-bit wide
EMesh for comparison	64-bit wide
Memory Bandwidth	4 memory controllers 5 GBps per controller
L2 Cache Size	256 KB

Table 6.1: Target System Architecture Configuration Parameters

6.1 Methodology

The Graphite [20] distributed multicore simulator is used for all evaluations in this section. For the 64 core simulations, the $ANet^{64}$ network is compared to a 64-bit wide electrical mesh network. For the 1024 core simulations, the $ANet^{1024}$ network is compared to a 256-bit wide electrical mesh network. The above comparisons are justified because the optical components of the ONet can be placed on a separate layer, thereby making the ONet have only few area requirements for receiver-side electrical buffering and arbitration. In addition, the area of a 128-bit wide BNet is roughly one-eighth the area of a 128-bit wide electrical mesh (see Chapter 3).

Table 6.1 summarizes the detailed target architectures. In $ANet^{64}$, short unicast messages less than four hops away are sent on the EMesh while broadcasts and long unicast messages are sent on the ONet. In $ANet^{1024}$ intra-cluster communication occurs through

the EMesh network while inter-cluster communication is carried out using the ENet, ONet and BNet networks. Small private L2 cache sizes were assumed for the 64-core study due to the small working set sizes of Splash2 benchmarks. All the references to EMesh in the remaining part of the evaluation section refer to the respective 64-core 64-bit wide and 1024-core 256-bit wide pure electrical mesh networks against which the $ANet^{64}$ and $ANet^{1024}$ networks are compared.

6.2 Performance Evaluation

6.2.1 Parsec and Splash2 Benchmarks

Nine applications from the Splash2 benchmark suite and three applications from the Parsec benchmark suite are simulated on 64 and 1024 cores using the 6 combinations of cache coherence protocols and networks mentioned previously.

64 cores

The configurations *ANet-Dir$_{64}$NB* and *ANet-Dir$_{64}$B* are expected to show the same performance as *ANet-ACKwise$_{64}$* since the directory type of the cache coherence protocol does not play a role when the number of hardware sharers is equal to the number of cores simulated. Similarly, the performance of *EMesh-ACKwise$_{64}$*, *EMesh-Dir$_{64}$NB* and *EMesh-Dir$_{64}$B* are expected to be the same. In the following discussion, ANet refers to $ANet^{64}$ described in Table 6.1.

Figure 6-1 plots the performance of the twelve benchmarks observed when running with the Dir$_k$NB cache coherence protocol on the ANet and EMesh networks. The performance is plotted as a function of the number of hardware sharers (k). Results are normalized to the performance observed when running with EMesh-Dir$_2$NB. With the Dir$_k$NB protocol, ANet is observed to outperform EMesh at all values of k and the performance difference is observed to decrease with increasing values of k. ANet-Dir$_2$NB outperforms EMesh-Dir$_2$NB by 30.9% while ANet-Dir$_{64}$NB outperforms EMesh-Dir$_{64}$NB by 12.8%. The performance of the Dir$_k$NB protocol is also observed to highly sensitive to the number

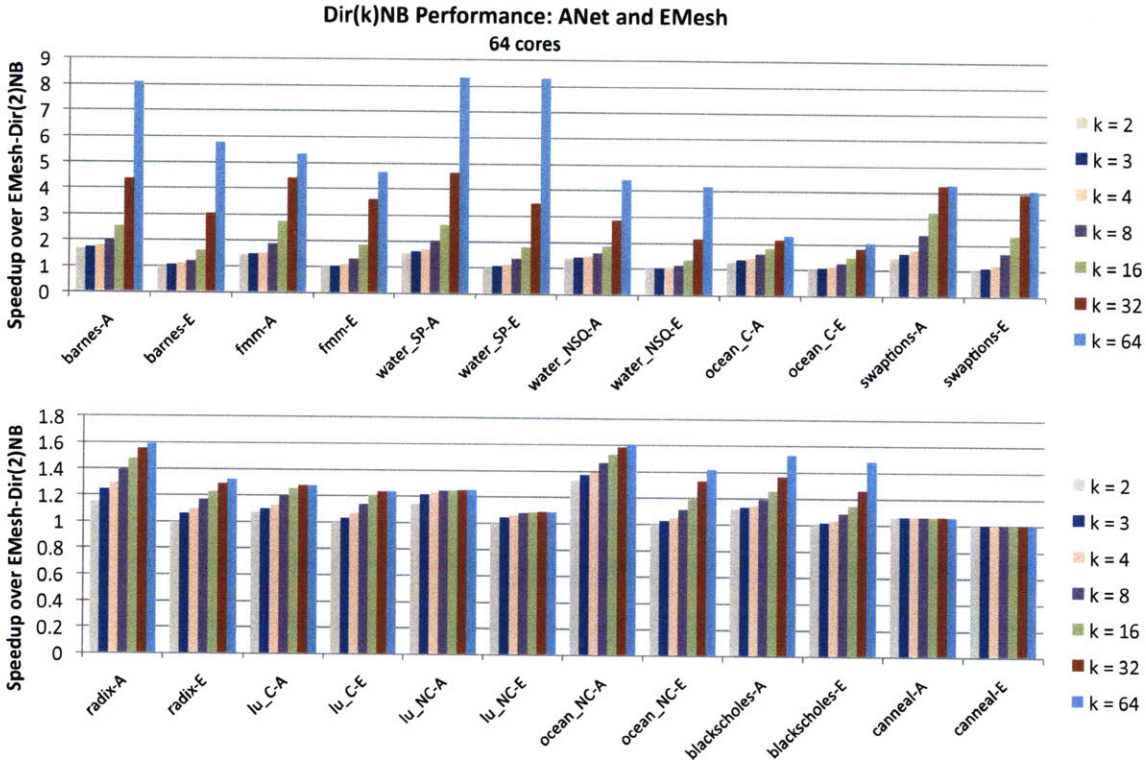

Figure 6-1: Performance of Splash2 and Parsec benchmarks when using the Dir_kNB protocol on the ANet and EMesh networks. Results are normalized to the performance of EMesh-Dir_2NB. The number of hardware sharers are varied as 2, 3, 4, 8, 16, 32 and 64. The x-axis values take the form *benchmark - network*. *A* and *E* stand for *ANet* and *EMesh* networks respectively.

of hardware sharers. The performance is extremely poor at low numbers of sharers and gradually improves as the number of sharers is increased. On the ANet network, Dir_{64}NB outperforms Dir_2NB by an average of 2.63x and a maximum of 5.51x (in *water-spatial*). On the EMesh network, Dir_{64}NB outperforms Dir_2NB by an average of 3.04x and a maximum of 8.29x (also in *water-spatial*).

The above results can be understood by observing Figure 6-4 which plots the cache miss rates of the benchmarks when run with the Dir_kNB protocol. The cache miss rates are observed to decrease as the number of hardware sharers (k) is increased. Hence, the performance increases with an increase in the value of k. High cache miss rates occur at low values of k due to the presence of a large number of true shared reads in these benchmarks. (A core is said to perform a true shared read when it reads from an address that is cached by at least another core in the system). The true shared reads lead to the occurrence of frequent

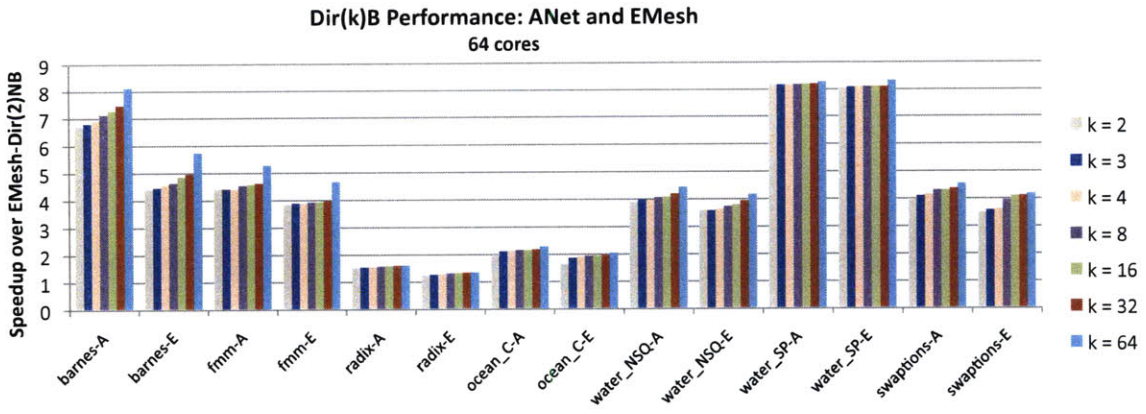

Figure 6-2: Performance of Splash2 and Parsec benchmarks when using the Dir_kB protocol on the ANet and EMesh networks. Results are normalized to the performance of EMesh-Dir_2NB. The number of hardware sharers are varied as 2, 3, 4, 8, 16, 32 and 64. The x-axis values take the form *benchmark - network*. *A* and *E* stand for *ANet* and *EMesh* networks respectively.

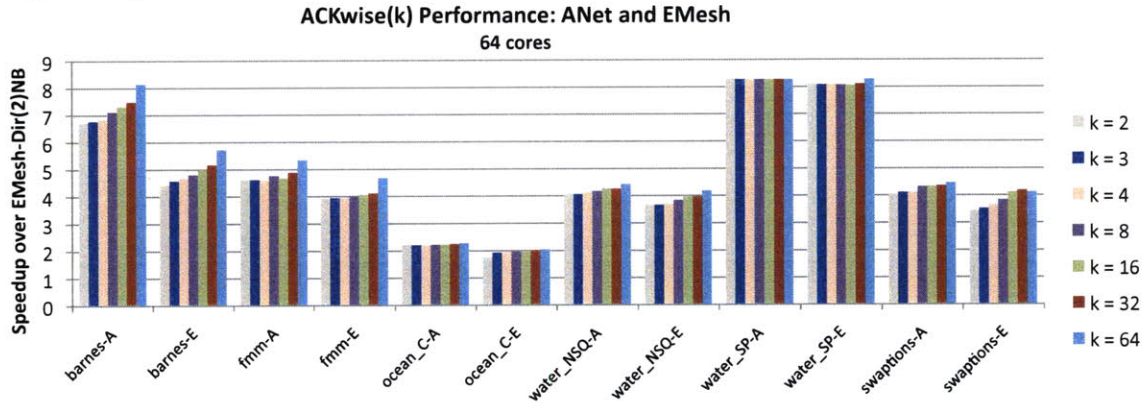

Figure 6-3: Performance of Splash2 and Parsec benchmarks when using the ACKwise_k protocol on the ANet and EMesh networks. Results are normalized to the performance of EMesh-Dir_2NB. The number of hardware sharers are varied as 2, 3, 4, 8, 16, 32 and 64. The x-axis values take the form *benchmark - network*. *A* and *E* stand for *ANet* and *EMesh* networks respectively.

invalidations because a large number of cores try to simultaneously read globally shared data and evict each others' cache lines in the process due to the restriction on the number of hardware sharers. The rate of increase of performance with k is directly correlated to the rate of decrease of cache miss rates with k as can be observed from Figures 6-1 and 6-4. This explains why benchmarks like *water_spatial* show a speedup of 8.29x on EMesh while others like *lu_non_contiguous* show very little speedup (9% on EMesh) when the number of hardware sharers is increased from 2 to 64. The cache miss rates of all benchmarks except

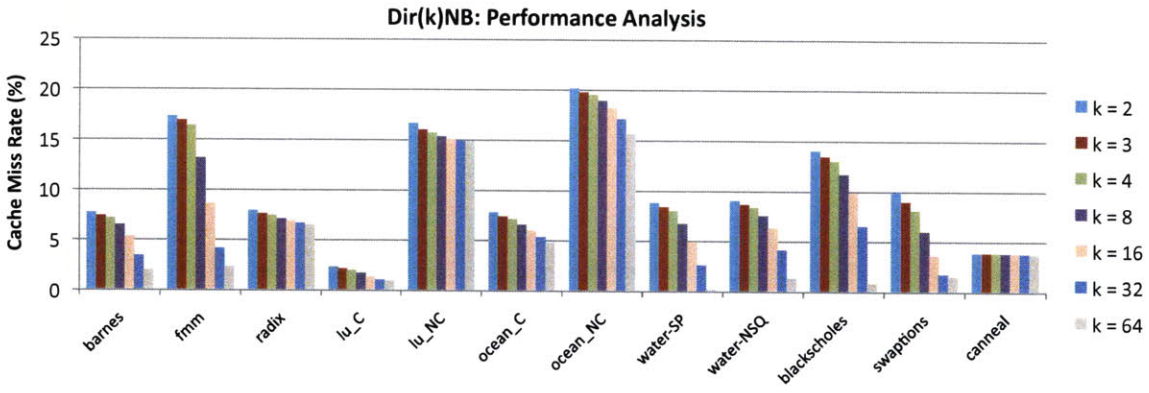

Figure 6-4: Cache miss rates observed when Splash2 and Parsec benchmarks are run using the Dir$_k$NB protocol. The number of hardware sharers are varied as 2, 3, 4, 8, 16, 32 and 64.

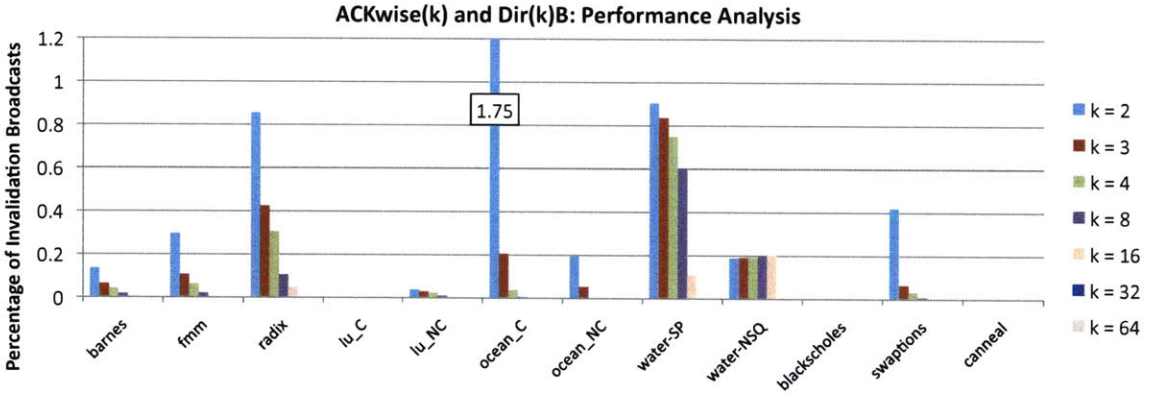

Figure 6-5: Percentage of invalidation broadcasts generated due to memory requests at the directory of a broadcast enabled cache coherence protocol (ACKwise$_k$ or Dir$_k$B).

canneal decrease with increasing k. *Canneal* has a very large working set with almost zero temporal locality. Due to this, any cache coherence protocol used with *canneal* is expected to show a constant miss rate given a particular cache size and cache line size.

At low values of k, since the cache miss rates are high, the network traffic intensity is also high. The bisection bandwidth of ANet^{64} is proportional to N while that of EMesh is proportional to $\sqrt{N}$ (N being the number of cores). Hence, ANet is more capable of handling higher network loads than the EMesh network. This explains why the performance difference between ANet and EMesh decreases with an increase in k and proves that ANet outperforms EMesh even with a purely unicast traffic pattern.

Figure 6-2 shows the performance of seven benchmarks when using the Dir$_k$B protocol on the ANet and EMesh networks. The results here are also normalized to the performance

of EMesh-Dir$_2$NB. The Dir$_k$B protocol shows less performance sensitivity to the number of hardware sharers than the Dir$_k$NB protocol. For the twelve benchmarks evaluated, Dir$_{64}$B outperforms Dir$_2$B by an average of 10.7% and a maximum of 21.3% (in *barnes*) on the ANet network. On the EMesh network, Dir$_{64}$B outperforms Dir$_2$B by an average of 13.2% and a maximum of 30.7% (also in *barnes*). The ANet network is observed to outperform the EMesh network at all values of k. On an average, ANet has a speedup of 14.1% over EMesh. The performance difference between ANet and EMesh is observed to slightly drop with increasing values of k. For the five benchmarks not shown in Figure 6-2, the performance speedup when the number of hardware sharers is increased from 2 to 64 is $<$ 6% for both the ANet and EMesh networks.

A Dir$_k$B protocol adversely affects the performance of the system when cache lines are widely shared and writes occur frequently to the widely shared cache lines. When a write occurs to a cache line that is shared by more than k cores, the following two types of messages are generated: (a) an invalidation broadcast message (from the sender core to all the cores in the system), and (b) N unicast messages (generated as acknowledgements to the invalidate message) from all cores in the system to the sender core (N is the number of cores). Since ANet^{64} possesses a specialized optical broadcast network (ONet), the broadcast message is handled efficiently. It does not affect the network load since the ONet network is contention-free. However, it does slightly increase the contention delay at the receiving core since there needs to be arbitration among the different messages destined for the same core at the receiving network interface. Since EMesh does not possess a specialized broadcast network, a broadcast is realized using N unicast messages directed from the sender core to all the other cores on the chip. These unicast messages raise the network load of the EMesh network significantly. On the other hand, the N unicast messages generated as acknowledgements raise the network load of both the ANet and EMesh networks. These N unicast messages have to be generated even if much fewer cores have cached the data. The invalidation broadcast message along with the acknowledgement messages account for the increase in performance when the number of hardware sharers is increased from 2 to 64.

Figure 6-5 shows the percentage of cache misses that lead to invalidation broadcast

messages in the benchmarks evaluated. Although it is difficult to quantify the exact dependence of performance on the amount of broadcast traffic due to other factors such as the burstiness of traffic, working set size, etc, it is nevertheless clear from the explanation above and from Figure 6-2 that performance increases steadily with decreasing broadcast traffic (or increasing number of hardware sharers). However, the Dir_kB protocol shows less performance sensitivity to the number of hardware sharers than the Dir_kNB protocol. This is because the benchmarks evaluated exhibit only a small number of true shared writes. (A core is said to perform a true shared write when it writes to an address that is cached by at least another core in the system.) Since the number of true shared writes are small, the invalidation broadcasts and the corresponding acknowledgements do not raise the network contention by a significant amount to adversely affect the overall system throughput. This fact is obvious from Figure 6-5 which shows that the percentage of cache misses that turn into invalidation broadcasts is almost always less than 1%. True shared reads, on the other hand, do not affect the performance of the Dir_kB protocol since the protocol does not place any restriction on the number of cores that can simultaneously cache an address in the read-only state.

Figure 6-3 shows the performance of the $ACKwise_k$ protocol on the ANet and EMesh networks. The results are again normalized to the performance of EMesh-Dir_2NB. The $ACKwise_k$ protocol shows the least performance sensitivity to the number of hardware sharers among the three protocols discussed. On average, $ACKwise_{64}$ outperforms $ACKwise_2$ by 7.9% on the ANet network and by 11.7% on the EMesh network. Like the previous two protocols, the ANet network is observed to outperform the EMesh network at all values of k. On average, ANet has a speedup of 14.5% over EMesh. For the six benchmarks that are absent in Figure 6-3 as well as for *ocean_contiguous*, the speedup of $ACKwise_k$ when the number of hardware sharers is increased from 2 to 64 is $< 3\%$ on ANet and $< 4\%$ on EMesh.

Like the Dir_kB protocol, $ACKwise_k$ is not affected by true shared reads since it allows any number of cores to simultaneously cache an address in the read-only state. The cache miss rates with the $ACKwise_k$ protocol are observed to be almost independent of the number of hardware sharers (k). For a true shared write to an address that has a sharing degree

$> k$, ACKwise$_k$ generates the invalidation broadcast message like the Dir$_k$B protocol. The impact of the invalidation broadcast on the performance of the ANet and EMesh networks is as described with the Dir$_k$B protocol. However, since ACKwise$_k$ intelligently tracks the number of sharers of a cache line once the capacity of the sharer list is exceeded, it needs acknowledgements from only the actual sharers of the cache line and not from all the cores in the system as in the Dir$_k$B protocol. In fact, the ACKwise$_k$ protocol only requires as many invalidation acknowledgements as a full-map directory-based protocol.

For both the ACKwise$_k$ and the Dir$_k$B protocols, the EMesh network shows a greater performance speedup than the ANet network when the number of hardware sharers is increased from 2 to 64 since it is not optimized for broadcast traffic. The ANet network, on the other hand, handles both unicast and broadcast traffic more efficiently due to its higher bisection bandwidth and specialized optical broadcast network, even at low numbers of hardware sharers.

The above results indicate the presence of a large amount of frequently read and sparsely written data in the twelve benchmarks evaluated which is corroborated in [31, 7]. Almost all the benchmarks studied exhibit significant read sharing and little write sharing. Due to this, the Dir$_k$NB protocol performs extremely poorly on both types of networks when compared to the ACKwise$_k$ and Dir$_k$B protocols. ACKwise$_k$ outperforms Dir$_k$NB by an average of 69.3% (across all values of k) and a maximum of 2.45x (when $k = 2$) on ANet. On EMesh, ACKwise$_k$ outperforms Dir$_k$NB by an average of 83.1% and a maximum of 2.73x (when $k = 2$). ACKwise$_k$ is only found to marginally outperform Dir$_k$B, the reason being the low percentage of true shared writes that the evaluated benchmarks generate. On average, ACKwise$_k$ outperforms Dir$_k$B by 2.1% on ANet by 1.6% on EMesh. In Section 6.2.2, the amount of write sharing is varied using a synthetic benchmark and the performance of the cache coherence protocols and networks are evaluated.

1024 Cores

In this section, five applications from the Splash2 benchmark suite are simulated on 1024 cores using the ANet^{1024} and EMesh networks and the 3 cache coherence protocols described previously. Figures 6-6, 6-7 and 6-8 show the performance results. The ANet-

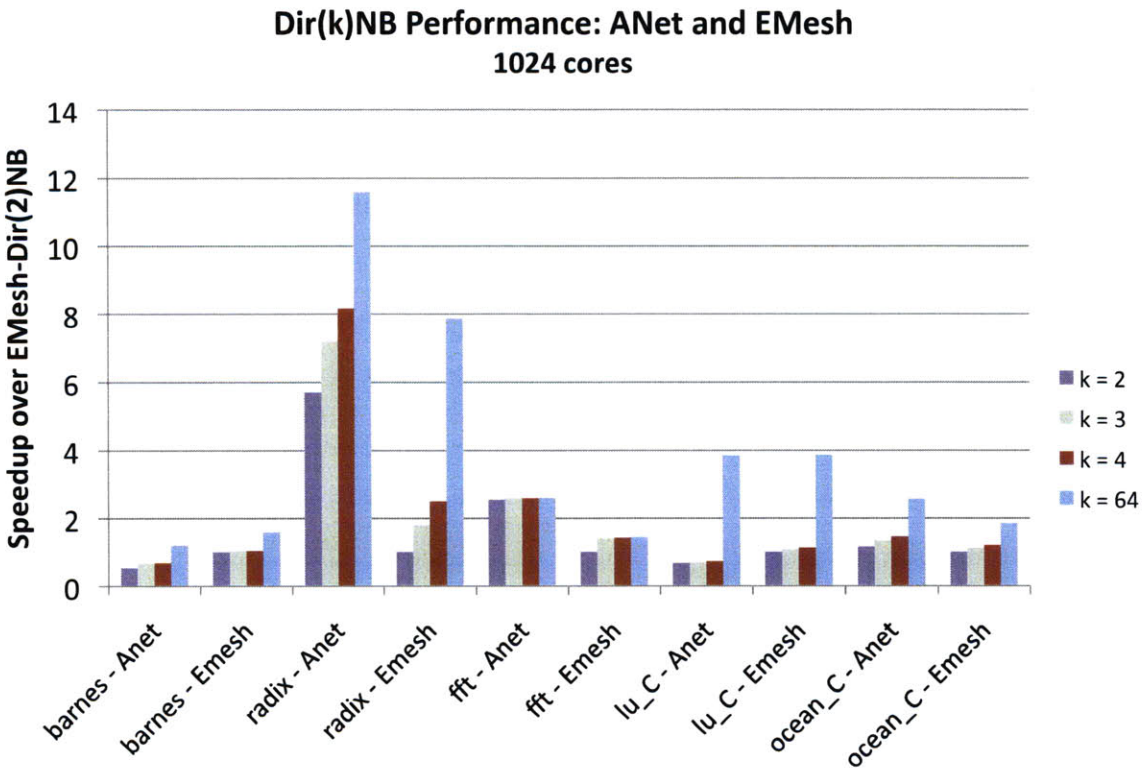

Figure 6-6: Performance of Splash2 when using the Dir$_k$NB protocol on the ANet and EMesh networks. Results are normalized to the performance of EMesh-Dir$_2$NB. The number of hardware sharers are varied as 2, 3, 4 and 64. The x-axis values take the form *benchmark - network*.

Ackwise$_k$ combination outperforms all other combinations of networks and cache coherence protocols across all evaluated values of k. The Dir$_k$NB protocol performs poorly on all benchmarks except *fft* on both the ANet and EMesh networks. The Dir$_k$B protocol also performs poorly on all benchmarks except *lu_contiguous*.

The Dir$_k$NB protocol performs poorly due to the presence of widely shared data that is frequently read and sparsely written in most Splash2 benchmarks. The negative effect of the protocol is even more evident on 1024-core processors since there are a larger number of threads trying to gain read-only access to the same cache line than on 64-core processors.

The Dir$_k$B protocol performs worser on 1024-core processors than on 64-core processors. On an electrical mesh network, with an injection rate of N packets/cycle, the utilization of each link is directly proportional to $\sqrt{N}$ (assuming an average hop count of $\frac{2\sqrt{N}}{3}$ per packet and a total of $4N$ links). Hence, the N unicast acknowledgements generated in

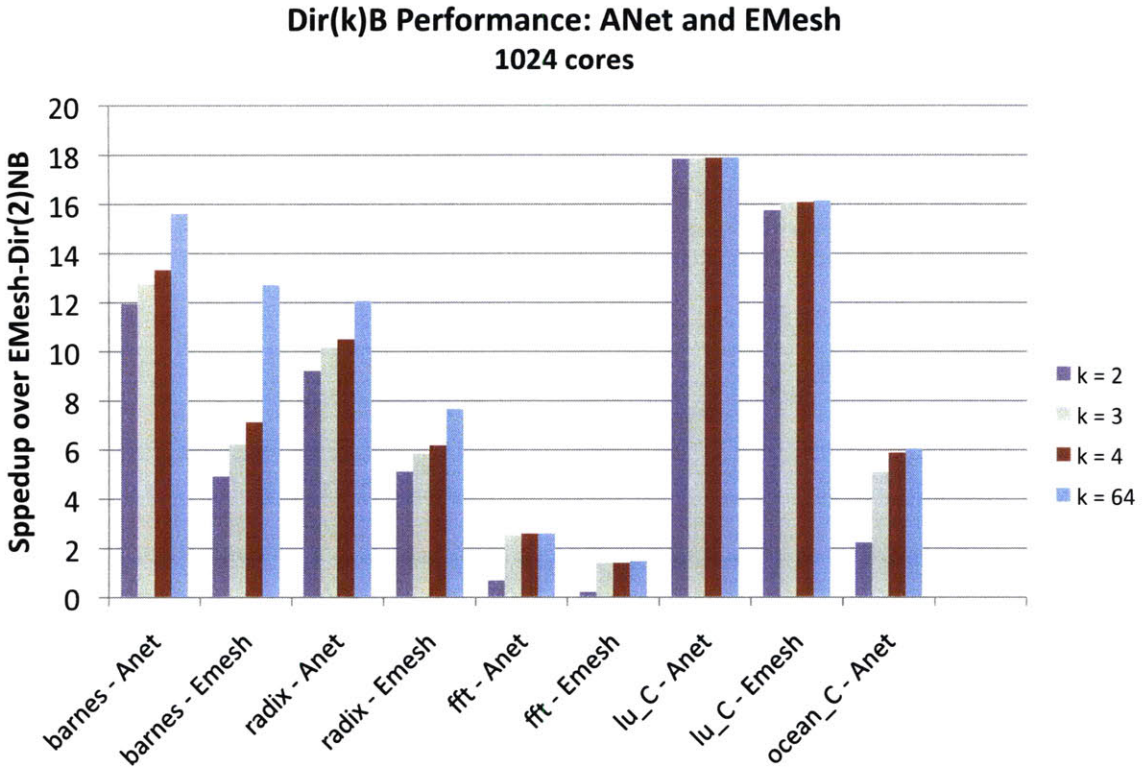

Figure 6-7: Performance of Splash2 when using the Dir_kB protocol on the ANet and EMesh networks. Results are normalized to the performance of EMesh-Dir_2NB. The number of hardware sharers are varied as 2, 3, 4 and 64. The x-axis values take the form *benchmark - network*.

response to an invalidation broadcast create higher network congestion in 1024-core processors than in 64-core processors with an electrical mesh network. This explains why the Dir_kB protocol performs poorly on the EMesh network. On the ATAC network, the bisection bandwidth is only doubled when the number of cores is increased from 64 to 1024 while the network traffic caused by N acknowledgements is proportional to N, and hence is increased 16-fold. Hence, the Dir_kB protocol performs poorly on the ATAC network also.

The performance of the ACKwise_k protocol on the ATAC network remains almost unchanged when the number of sharers is increased from 2 to 64. (The greatest fluctuation is 5% with *barnes*). The performance of the ACKwise_k protocol on the ATAC network is unchanged since the additional broadcasts that are generated are handled efficiently by the ATAC network. However, on the EMesh network, only two applications, *fft* and

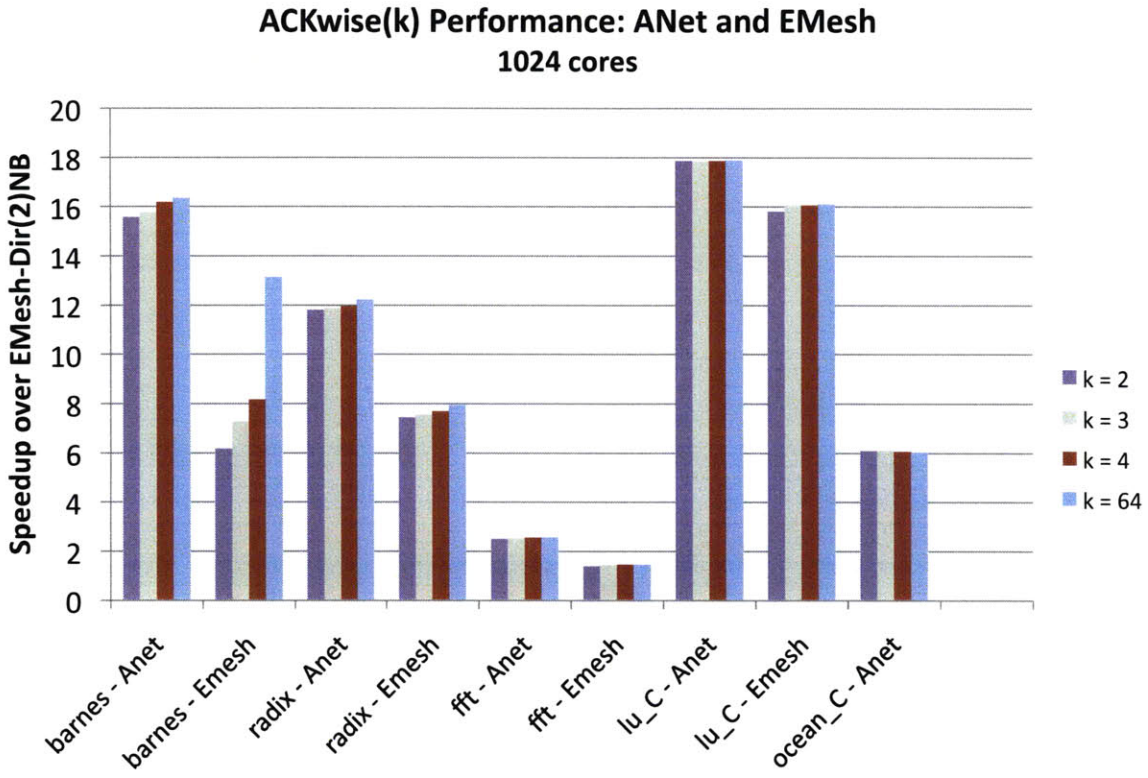

Figure 6-8: Performance of Splash2 when using the ACKwise$_k$ protocol on the ANet and EMesh networks. Results are normalized to the performance of EMesh-Dir$_2$NB. The number of hardware sharers are varied as 2, 3, 4 and 64. The x-axis values take the form *benchmark - network*.

lu_contiguous show unchanged performance. *barnes* shows a 2x increase in performance when the number of sharers is increased from 2 to 64. This is because the EMesh network is not optimized for broadcast as discussed previously.

However, for unicast traffic, most of the performance gains of ANet are due its lower hop count (lower uncontended latency). There are no bandwidth gains for ANet over the electrical mesh network since the bisection bandwidth of the 128-bit wide ANet^{1024} network is $128 \times 64 = 8192$ Gbps which is less than that of the 256-bit wide electrical mesh network ($256 \times 32 \times 2 = 16384$ Gbps). This fact can be observed from Figure 6-6 where the performance of the ATAC network is slightly less than that of the EMesh network for two applications, *barnes* and *lu_contiguous*. Also notice that the performance of the Dir$_{64}B$ and Dir$_{64}$NB protocols is less than the performance of the ACKwise$_{64}$ protocol. This is because the number of hardware sharers, 64 is still less than the number of cores, 1024.

On an average, the ACKwise$_k$ protocol and the ATAC network outperform conventional directory-based coherence protocols on an electrical mesh by 78% with 1024-core simulations with Splash2 benchmarks.

6.2.2 Synthetic Benchmarks

The Splash2 and Parsec benchmarks are highly structured applications that exhibit extremely good cache behavior as observed in the previous section. They exhibit very high read sharing and little write sharing which is corroborated in [31, 7]. They are not representative of future multicore workloads that widely share data and exhibit highly unstructured access to them. In this section, we evaluate the performance of a synthetic benchmark that emulates different types of workloads (which exhibit different fractions of read and write sharing) when run with the 6 combinations of cache coherence protocols and networks mentioned previously. Experiments are done both on 64 and 1024 cores.

The characteristics of the synthetic benchmark used are shown in Table 6.2. The benchmark is constructed by assigning probabilities to instructions and memory access types. Data accessed by the synthetic benchmark is divided into three types: (a) private data, (b) shared data that is only read (read-only shared data) and (c) shared data that is read and written (read-write shared data). Among the instructions that access private data and read-write shared data, the fraction of reads to the fraction of writes is assumed to be 2:1 (because most operations read data from two memory locations, do some computation, and store the result in a third location). The only variables in the synthetic benchmark are the fraction of instructions that access read-only shared data and the degree of sharing of the shared data. For read-only shared data, a sharing-degree d denotes that this data can be read by a total of d sharers and for read-write shared data, degree d denotes that this data can be read/written by a total of d sharers. The amount of private data each core can access is 16 KB and the total amount of shared data is 64 KB and 1 MB for 64-core and 1024-core simulations respectively.

Instruction Mix: Non-Memory Instructions Shared Data Access Private Data Access	 70% 10% 20%
Read-only Fraction of Shared Data	{25%, 75%}
Private Data per Thread	16 KB
Total Shared Data	64 KB (64-core) 1 MB (1024-core)
Degree of Application Sharing	{1, 2, 4, 8, 16, 32, 64}
Instructions Simulated per Thread	1 million (64-core) 100,000 (1024-core)

Table 6.2: Synthetic Benchmark Characteristics

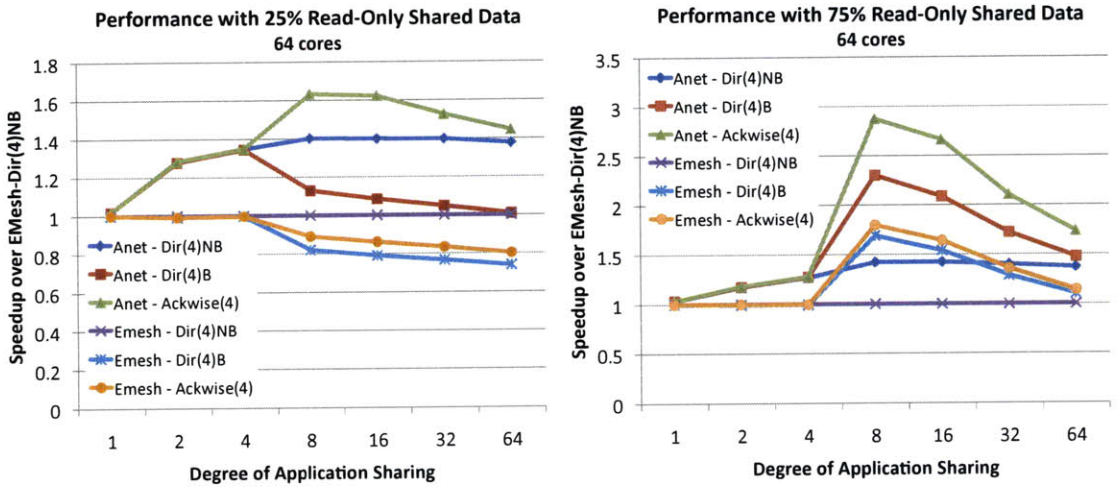

Figure 6-9: Performance of the synthetic benchmark running on 64 cores with six different combinations of networks and cache coherence protocols. The performance is normalized to that of EMesh-Dir$_4$NB.

64 Cores

In the following experiments, the network architectures (ANet^{64} and *EMesh*) and cache coherence protocols (*ACKwise*$_k$, *Dir*$_k$*B* and *Dir*$_k$*NB*) studied are as discussed in Table 6.1. k, the number of hardware sharers, is fixed at 4. The percentage of instructions that access read-only shared data among those that access shared data is set to either 25% or 75%. The number of application sharers is varied from 1 to 64 in powers of 2.

25% Read-Only, 75% Read-Write From Figure 6-9(a), it can be observed that the ACKwise$_4$ protocol performs best on ANet and the Dir$_4$NB protocol performs best on EMesh. The ACKwise$_4$ and Dir$_4$B protocols perform poorly on EMesh. The performance worsens as the degree of application sharing increases. This is because an increase in the degree of sharing increases the number of broadcast invalidations and a pure electrical mesh performs poorly with a lot of broadcast traffic. The Dir$_4$NB protocol, on the other hand does not produce any broadcast traffic. Moreover, the performance penalty of evicting a sharer in order to accommodate another sharer is small for 75% of the data because exclusive requests arrive frequently for cache lines in that address space.

The ANet network on the other hand supports broadcast traffic efficiently and hence ACKwise$_4$ has the best performance. The Dir$_4$B protocol still suffers due to the many unicast acknowledgements that have to be sent as a result of a broadcasted invalidation. The Dir$_4$NB protocol on ANet is found to perform slightly worse than ACKwise$_4$.

75% Read-Only, 25% Read-Write From Figure 6-9(b), it can be observed that the ACKwise$_4$ protocol performs best on both ANet and EMesh. With 75% read-only shared data, the Dir$_4$NB protocol performs poorly on both networks because all sharers of a read-only shared cache line cannot have the data in their private caches at the same time. Hence, the cores accessing read-only shared data keep invalidating each other frequently. The performance of the Dir$_4$B protocol lies between that of ACKwise$_4$ and Dir$_4$NB protocol. Even though the Dir$_4$B protocol achieves the same performance as ACKwise$_4$ on read-only shared data, it still suffers when there are a sufficient number of broadcast invalidation requests because it has to collect acknowledgements from all the cores for each broadcasted invalidation. This configuration produces results extremely similar to those produced by the Splash2 and Parsec benchmarks.

1024 Cores

The network architectures ANet^{1024} and EMesh studied are as discussed in Table 6.1. In this section we only show results for the synthetic benchmark that has 25% read-only data. The results for the 75% read-only synthetic benchmark are very similar to those shown in

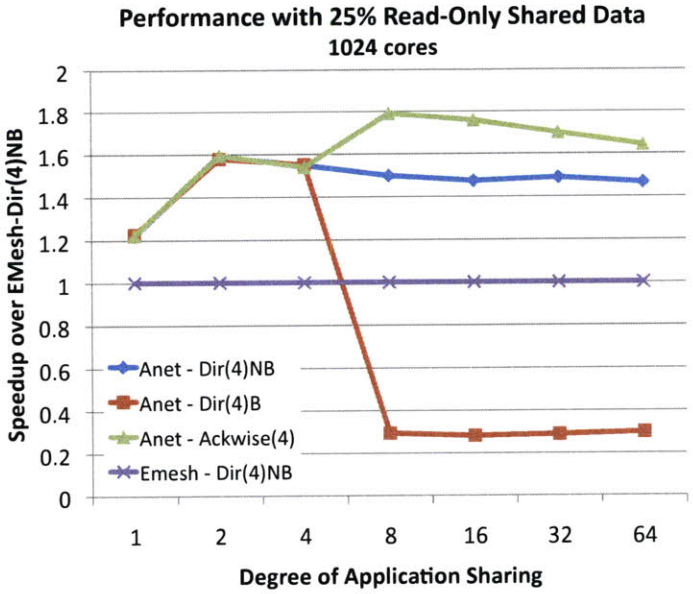

Figure 6-10: Performance of the synthetic benchmark running on 1024 cores with 4 different combinations of networks and cache coherence protocols. The performance is normalized to that of EMesh-Dir$_4$NB. ACKwise$_4$ and Dir$_4$B protocols perform poorly on a pure electrical mesh with this synthetic benchmark as discussed in Section 6.2.2.

Section 6.2.2.

Figure 6-10 shows that the ACKwise$_4$ protocol coupled with the ANet network provides the best results. The Dir$_4$B protocol performs extremely poorly on ANet due to its lack of network bandwidth for the large number of unicast acknowledgements generated by the protocol. This fact is corroborated by the extremely large queueing delays observed at the sending Hub with the Dir$_4$B protocol. Overall, ANet-ACKwise$_4$ outperforms the best cache coherence protocol on EMesh (Dir$_4$NB in this case) by an average of 61%. ACKwise$_4$ and Dir$_4$B protocols perform poorly on a pure electrical mesh with this synthetic benchmark due to the reasons outlined in Section 6.2.2.

From the experiments conducted, it can be concluded that the Dir$_k$B protocol performs well on benchmarks that have widely shared data which is frequently read and sparsely written. The Dir$_k$NB protocol performs well when the widely shared data is frequently written. ACKwise$_k$ performs well on both the above types of benchmarks given the presence of a network with specialized broadcast support. This paper has built and evaluated such a network using nanophotonic technology.

6.3 Energy Evaluation

In this section, the energy consumption of 12 Splash2 benchmarks when running on 1024 cores with the ATAC and EMesh networks is evaluated. The ACKwise protocol with 4 hardware sharers is used for all experiments conducted. The architectural parameters used for comparison are shown in Table 6.1.

Both aggressive and conservative predictions of the devices in the optical network (ONet) are used to obtain the chip power consumption of the ANet network. Aggressive and conservative optical losses and device characteristics are evaluated to obtain lower and upper bounds for the energy consumption as well as to provide a goal for optical and electronic device researchers to target. The total energy consumption of the EMesh network is also computed for understanding the energy feasibility of optical technology in building efficient on-chip networks. The results are plotted in Figures 6-11 and 6-12 and are normalized to the energy consumed when running with the aggressive ANet network. For convenience, the ANet network with the aggressive power estimate is denoted as *ANet-Agg* and that with the conservative estimate is denoted as *ANet-Cons*.

From the graphs, it is clear that the energy consumption of the EMesh network lies between that of the ANet-Agg and ANet-Cons networks. On an average, the energy consumption of the EMesh network is $2.24x$ higher that of the ANet-Agg network and $1.51x$ lower than that of the ANet-Cons network. The energy consumption of the ANet-Agg network is $3.38x$ lower than that of the ANet-Cons network. The differences in energy consumption between the ANet-Agg and ANet-Cons network arise due to the differences in the (1) laser power; (2) ring tuning power; (3) dynamic power of modulator + modulator driver; and (4) dynamic power of the photodetector + photodetector receiver in the constituent ONet network. These differences arise due to optical losses and device capacitances assumed for the aggressive and conservative cases as discussed in Chapter 5.2. The laser power is clearly the dominant factor among the above four sources of differences as is shown by Figures 6-11 and 6-12.

For the ANet-Agg network, the constituent electrical networks (ENet and BNet) consume more energy than the optical network (ONet). The energy consumption of the ONet

is only $\approx 5\%$ of the energy consumption of the ANet-Agg network. Almost all the energy consumed by the ONet ($95\% - 99\%$) is due to the static power consumed by the laser. This is due to the fact that (1) Laser Energy (per wavelength per bit) is an order of magnitude higher than the dynamic energy of the modulator, modulator driver, photodetector and receiver; and (2) The utilization of the ONet varies from $5\% - 35\%$ in the benchmarks evaluated. A low utilization implies that the laser energy will be more dominant in the net energy consumption of the ONet.

For the ANet-Cons network, the major source of power is the laser power which forms $\approx 65\%$ of the total chip power consumption. With conservative optical technology predictions, carrying out a unicast with the ONet optical network is extremely expensive since the laser power corresponding to 64 receivers has to be paid even though a unicast requires only one receiver. For the ANet-Cons network to be power efficient, there has to be sufficient broadcast traffic to offset the impact of the static laser power. A unicast (even a long-distance one) on the ANet-Cons network is highly inefficient in terms of energy. To transmit a bit from one end of the chip to the other using a 256-bit wide electrical mesh, it takes $390 \times 62 = 24.57$ pJ which is less than the laser energy for transmitting a bit (24.9 pJ) (see Table 5.2 and Table 5.6).

In addition, the ATAC architecture leverages off-chip lasers to generate photons to transmit data. State-of-the-art off-chip lasers operate at an efficiency of 30%. The wall-plug (electrical) power consumption of the laser is related to its chip (optical) power consumption by the equation:

$$wall-plug\ (electrical)\ laser\ power = \frac{chip\ (optical)\ laser\ power}{laser\ efficiency} \tag{6.1}$$

as discussed in Chapter 5.2.2. Hence, to get the total system power of the ANet network, the electrical laser power must be substituted in-place of the optical laser power. However, in this evaluation, the graphs plotted only consider the optical laser power. This is due to the fact that chip power dissipation is a more important metric for evaluation than total system power since chip power determines the cost of the cooling method that must be used (whether conventional cooling solutions can be applied or a specialized cooling solution is

required). For the ANet-Agg network, since the laser power forms only $\approx 5\%$ of the total network power, the total system power is expected to be almost the same as the chip power. However, for the ANet-Cons network, the total system power will be at least 2x the chip power since the laser dominates the power consumption.

On an average, the ANet-Agg network has 2.24x lower energy consumption than the EMesh network while the ANet-Cons network has 1.51x higher energy consumption than the EMesh network. The ATAC network architecture with conservative optical predictions should be highly utilized with broadcast traffic for it to be efficient and if such a traffic pattern does not exist, then the ATAC network architecture has to be redesigned to optimize the energy consumption for unicast traffic in order to make it more efficient.

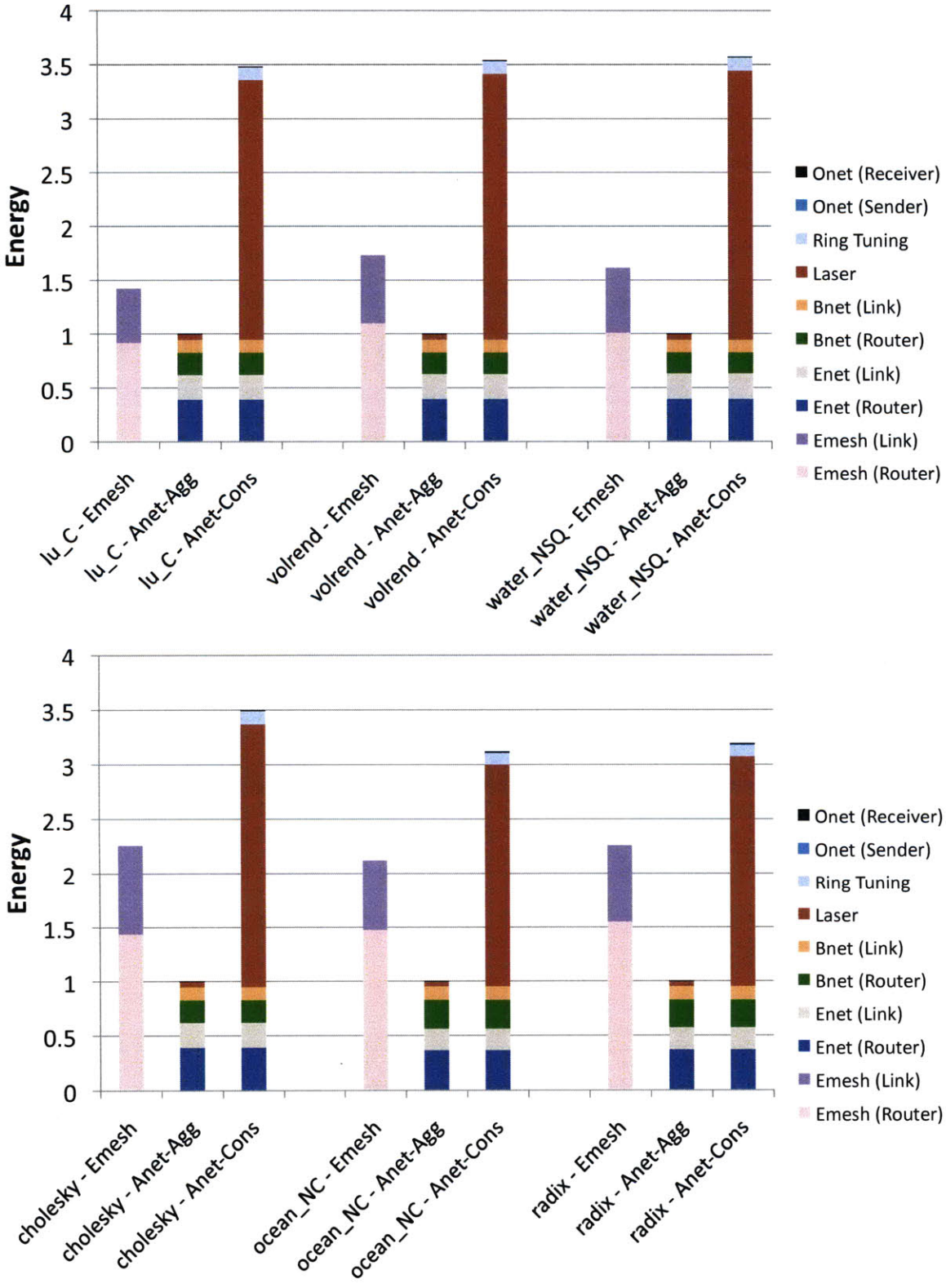

Figure 6-11: Energy Comparison of ANet and EMesh networks. The x-axis values take the form *benchmark - network*. *EMesh*, *ANet-Agg* and *ANet-Cons* stand for the EMesh network and the aggressive and conservative ANet networks respectively.

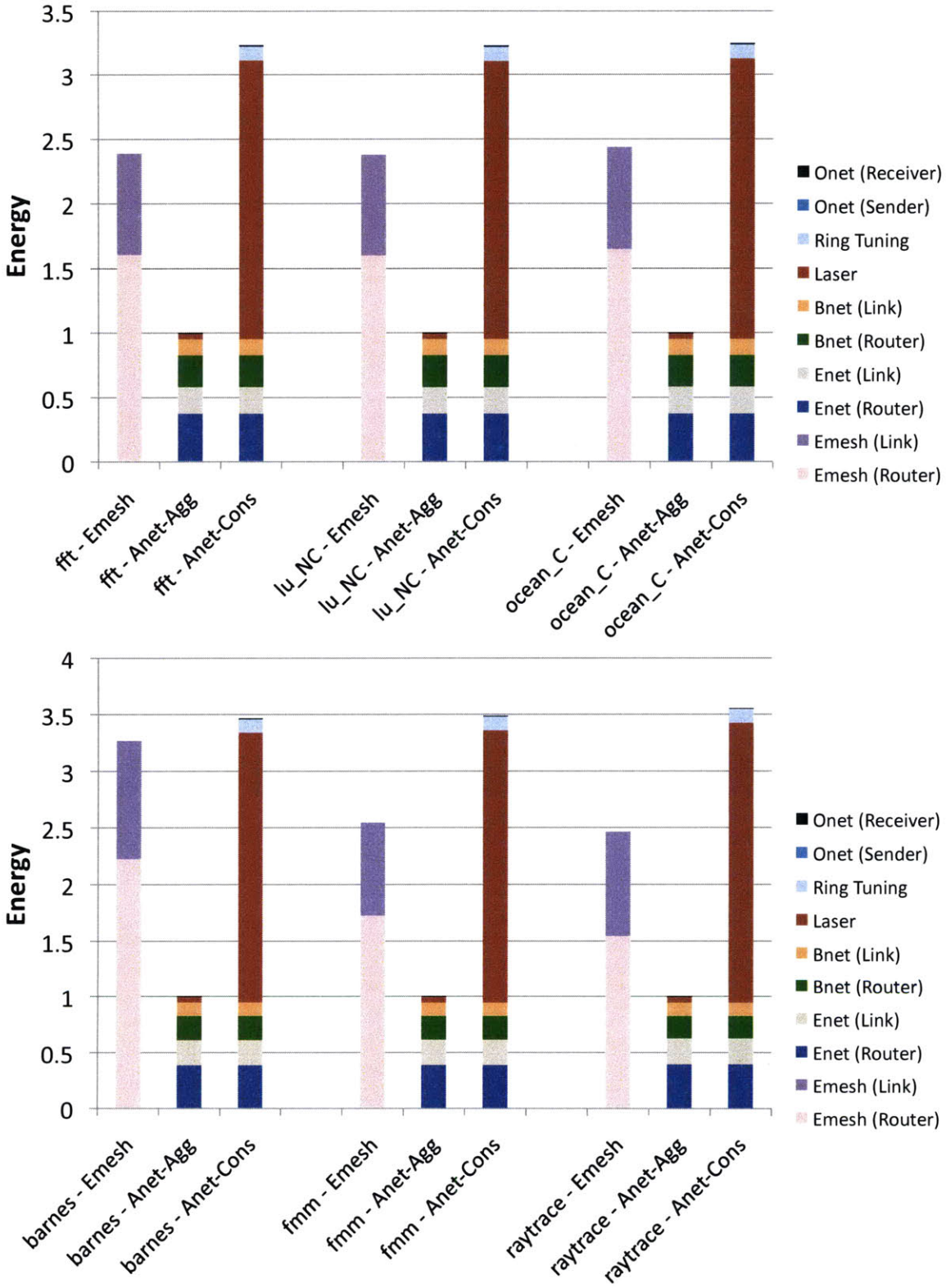

Figure 6-12: Energy Comparison of ANet and EMesh networks. The x-axis values take the form *benchmark - network*. *EMesh*, *ANet-Agg* and *ANet-Cons* stand for the EMesh network and the aggressive and conservative ANet networks respectively.

Chapter 7

Related Work

CMOS-compatible nanophotonic devices are an emerging technology. Therefore there have only been a few architectures proposed that use them for on-chip communication: Corona [11], the optical cache-coherence bus of Kirman et al [27], and the switched optical NoC of Shacham et al [4].

The Corona architecture primarily differs from ATAC in the way that it assigns communication channels. While Corona assigns a physical channel to each receiver and uses WDM to send multiple bits of a dataword simultaneously, ATAC assigns a physical channel to each sender and uses WDM to carry multiple channels in each waveguide, eliminating contention and the need for arbitration.

Kirman et al [27] design a cache-coherent hierarchical optoelectronic bus, consisting of a top-level optical broadcast bus which feeds small electrical networks connecting groups of cores. The design of their network is similar to ATAC but is limited to snooping cache coherence traffic whereas ATAC is composed of a network supporting a general communication mechanism and a coherence protocol (i.e, *ACKwise*) designed to scale to hundreds of cores.

Shacham et al [4] propose a novel hybrid architecture in which they combine a photonic mesh network with electronic control packets. Their scheme is somewhat limited by the propagation of electrical signals since they use an electronic control network to setup photonic switches in advance of the optical signal transmission. It only becomes efficient when a very large optical payload follows the electrical packet. ATAC, on the other hand,

leverages the efficiencies of optical transmission for even a single word packet.

Pan et al. [29] proposed Firefly, a hybrid electrical-optical network architecture. Similar to ATAC, Firefly breaks the chip into clusters of cores interconnected by electrical links. Clusters communicate via a single-writer multiple-reader optical network. Unlike ATAC, Firefly's photonic links use an optical crossbar which must be configured by a handshake between the sender and receiver. Firefly partitions its crossbar into multiple smaller logical crossbars to eliminate the need for global arbitration.

Batten et al. [6] take a different approach and use integrated photonics to build a high-performance network that connects cores directly to external DRAM. However, their design does not allow for optical core-to-core communication. An ATAC processor could leverage their design to connect its memory controllers to DRAM.

Previous techniques for reducing cache coherence directory storage space include using hierarchical directories [32], coarse vectors [3], sparse directories [3], chained directories [9], and maintaining limited directories with broadcasting capabilities [2] or software support [10]. The *ACKwise* protocol, on the other hand, augments a limited directory based protocol by tracking the number of sharers once the capacity of the sharer list is exceeded. It also borrows the strategy of maintaining a clean owner for reducing the offchip miss rate from *cooperative caching* [15]. Recent proposals for a cache organization combining the low hit latency of a private L2 cache and the low miss rate of a shared L2 cache [15, 26] are orthogonal to *ACKwise* and could be used along it.

Chapter 8

Conclusion

The recent advances of optical technology have certainly inspired confidence in computer architects that optics may very well continue to make its way into smaller and smaller packages; just as optical interconnect has moved from connecting cities to connecting data centers, it seems likely that it will soon connect chips and on-chip components.

Overall, this paper presented a novel manycore architecture that scales to 1000 cores by embracing new technology offered by recent advances in nanophotonics. This paper also introduced *ACKwise*, a novel directory-based cache coherence protocol that takes advantage of the special properties of the ATAC network to achieve high performance. From 64-core and 1024-core evaluations with Splash2, Parsec and synthetic benchmarks, it is observed that the *ACKwise* protocol on ANet outperforms all other combinations of networks and cache coherence protocols. On 1024-core evaluations, ACKwise protocol on ATAC outperforms the best conventional cache coherence protocol on an electrical mesh network by 78% with Splash2 benchmarks and by 61% with synthetic benchmarks. Energy simulations show that the energy consumption of the ATAC network that assumes aggressive optical technology predictions is 2.24x lower than that of an electrical mesh network. However, with conservative optical technology predictions, the energy consumption of the ATAC network is 1.51x higher than that of an electrical mesh network.

Bibliography

[1] The International Technology Roadmap for Semiconductors (ITRS) Technology Working Groups, 2008.

[2] A. Agarwal et al. An evaluation of directory schemes for cache coherence. In *ISCA*, 1988.

[3] A. Gupta et al. Reducing Memory and Traffic Requirements for Scalable Directory-Based Cache Coherence Schemes. In *ICPP*, 1990.

[4] A. Shacham et al. Photonic NoC for DMA Communications in Chip Multiprocessors. In *Hot Interconnects*, Aug 2007.

[5] Andew B. Kahng, Bin Li, Li-Shiuan Peh and Kambiz Samadi. Orion 2.0: A fast and accurate noc power and area model for early-stage design space exploration. In *Design, Automation and Test in Europe, DATE*, pages 423–428, April 2009.

[6] C. Batten et al. Building manycore processor-to-dram networks with monolithic silicon photonics. In *Hot Interconnects*, pages 21–30, Aug 2008.

[7] C. Bienia et al. The PARSEC Benchmark Suite: Characterization and Architectural Implications. In *PACT*, 2008.

[8] D. Ahn et al. High performance, waveguide integrated Ge photodetectors. In *Optics Express 15, 3916*, 2007.

[9] D. Chaiken et al. Directory-Based Cache Coherence in Large-Scale Multiprocessors. In *IEEE Computer, Vol 23, p 49-58*, June 1990.

[10] D. Chaiken et al. Limitless Directories: A scalable cache coherence scheme. In *ASPLOS*, 1991.

[11] D. Vantrease et al. Corona: System Implications of Emerging Nanophotonic Technology. In *ISCA*, 2008.

[12] D. Wentzlaff et al. On-chip interconnection architecture of the Tile Processor. *IEEE Micro*, 27(5):15–31, 2007.

[13] George Kurian, Jason E. Miller, James Psota, Jonathan Eastep, Jifeng Liu, Jurgen Michel, Lionel Kimerling and Anant Agarwal. Atac: A 1000-core cache-coherent

processor with on-chip optical network. In *Parallel Architectures and Compilation Techniques, PACT*, September 2010.

[14] Intel Corporation. Intel's Teraflops Research Chip. http://techresearch.intel.com/articles/Tera-Scale/1449.htm.

[15] J. Chang et al. Cooperative Caching for Chip Multiprocessors. In *ISCA*, 2006.

[16] J. F. Liu et al. Waveguide-integrated, ultra-low energy GeSi electro-absorption modulators. In *Nature Photonics 2, 433*, 2008.

[17] J. Michel et al. Advances in Fully CMOS Integrated Photonic Circuits. In *Proc. of the International Society for Optical Engineering (SPIE) 6477, p64770P-1-11*, 2007.

[18] J. Psota et al. ATAC: All-to-All Computing Using On-Chip Optical Interconnects. In *BARC*, 1/2007.

[19] J. Psota et al. Improving performance and programmability with on-chip optical networks. In *ISCAS*, 2010.

[20] J.Miller et al. Graphite: A Distributed Parallel Simulator for Multicores. 2009.

[21] R. Kirchain and L.C. Kimerling. A roadmap for nanophotonics. In *Nature Photonics, 1 (6): 303-305*, 2007.

[22] Michael Kistler, Michael Perrone, and Fabrizio Petrini. Cell multiprocessor communication network: Built for speed. *IEEE Micro*, 26(3):10–23, May-June 2006.

[23] J. F. Liu and J. Michel. High Performance Ge Devices for Electronic-Photonic Integrated Circuits. In *ECS Transactions, Vol 16, p 575-582*, 2008.

[24] M. Beals et al. Process flow innovations for photonic device integration in CMOS. In *Proc. of the International Society for Optical Engineering (SPIE) 6898, 689804*, 2008.

[25] M. Taylor et al. Evaluation of the Raw Microprocessor: An Exposed-Wire-Delay Architecture for ILP and Streams. In *ISCA*, 2004.

[26] M. Zhang et al. Victim Replication: Maximizing Capacity while Hiding Wire Delay in Tiled Chip Multiprocessors. In *ISCA*, 2005.

[27] N. Kirman et al. Leveraging Optical Technology in Future Bus-based Chip Multiprocessors. In *MICRO*, 2006.

[28] P. Sweazey et al. A Class of Compatible Cache Consistency Protocols and their Support by the IEEE Futurebus. In *ISCA*, 1986.

[29] Yan Pan, Prabhat Kumar, John Kim, Gokhan Memik, Yu Zhang, and Alok N. Choudhary. Firefly: illuminating future network-on-chip with nanophotonics. In *ISCA*, pages 429–440, 2009.

[30] Clint Schow. Optical Interconnects in Next-Generation High-Performance Computers. OIDA 2008 Integration Forum, 2008.

[31] S.Woo et al. The SPLASH-2 Programs: Characterization and Methodological Considerations. 1995.

[32] Y. Maa et al. Two economical directory schemes for large-scale cache coherent multiprocessors. In *ACM SIGARCH Computer Arch News, Vol 19*, Sept 1991.

[33] Winnie N. Ye, Jurgen Michel, and Lionel C. Kimerling. Athermal high-index-contrast waveguide design. *IEEE Photonics Technology Letters*, 20(11):885–887, 2008.