Computer Science and Artificial Intelligence Laboratory
Technical Report

MIT-CSAIL-TR-2008-066 November 11, 2008

Energy Scalability of On-Chip Interconnection Networks in Multicore Architectures

Theodoros Konstantakopoulos, Jonathan Eastep, James Psota, and Anant Agarwal

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

CSAIL

Energy Scalability of On-Chip Interconnection Networks in Multicore Architectures

Theodoros Konstantakopoulos, Jonathan Eastep, James Psota, and Anant Agarwal
CSAIL, Massachusetts Institute of Technology

Abstract

On-chip interconnection networks (OCNs) such as point-to-point networks and buses form the communication backbone in systems-on-a-chip, multicore processors, and tiled processors. OCNs can consume significant portions of a chip's energy budget, so analyzing their energy consumption early in the design cycle becomes important for architectural design decisions. Although numerous studies have examined OCN implementation and performance, few have examined energy. This paper develops an analytical framework for energy estimation in OCNs and presents results based on both analytical models of communication patterns and real network traces from applications running on a tiled multicore processor.

Our analytical framework supports arbitrary OCN topologies under arbitrary communication patterns while accounting for wire length, switch energy, and network contention. It is the first to incorporate the effects of communication locality and network contention, and use real traces extensively.

This paper compares the energy of point-to-point networks against buses under varying degrees of communication locality. The results indicate that, for 16 or more processors, a one-dimensional and a two-dimensional point-to-point network provide 66% and 82% energy savings, respectively, over a bus assuming that processors communicate with equal likelihood. The energy savings increase for patterns which exhibit locality. For the two-dimensional point-to-point OCN of the Raw tiled microprocessor, contention contributes a maximum of just 23% of the OCN energy, using estimated values for channel, switch control logic, and switch queue buffer energy of $34.5pJ$, $17pJ$, and $12pJ$, respectively. Our results show that the energy-delay product per message decreases with increasing processor message injection rate.

I. Introduction

Microprocessor performance has increased dramatically over the last three decades as advancing semiconductor technology has vastly increased both the quantity and speed of on-chip transistors available to computer architects. Computer designers have taken advantage of these resources largely to build systems with centralized structures such as superscalar and pipelined processors with large caches, due to the simpler programming model compared to systems with distributed structures. In recent times, however, power consumption and wire delay have limited the continued scaling of centralized systems [1], while making multicore architectures increasingly popular.

The minimum feature size in production integrated circuits has continued on an exponential decline since the first integrated circuit appeared [2]. Riding this wave, as the number of processing elements scale up in multicore and tiled processor architectures, the study of on-chip interconnection networks – the medium for processor-to-processor or memory communication – becomes extremely important. Although network performance has been extensively studied (e.g., [3], [4], [5], [6], [7]), the power and energy of OCNs have not been explored as rigorously. As the energy consumption in OCNs increases [8], energy estimation tools that can provide a comparison of different network architectures under various network traffic patterns early in the design cycle become extremely useful to the computer architect.

This paper proposes an energy analysis framework for on-chip interconnection networks that can serve as a basis to model (a) multidimensional point-to-point networks or buses that transfer data between processors, (b) OCNs that connect distributed resources such as caches on chip, or for that matter (c) networks that communicate data between different components such as ALUs and register files.

The paper differs from previous work on energy analysis of OCNs [9], [10], [11], [12] by making the following four unique contributions: (1) The paper develops an analytical framework for the energy analysis of OCNs of any topology (buses, tori, 3-D meshes, etc.) and arbitrary communication patterns (uniform, truncated exponential, measured from a real workload, etc.). It compares the energy of one-, two-dimensional and multidimensional point-to-point OCNs (mapped to a 2-D physical on-chip substrate) to that of buses. The model includes the impact of switch energy, wire lengths and related capacitance, and communication locality. (2) The paper extensively uses real network traces from benchmarks running on a tiled microprocessor to compare the energy performance of OCNs, and to validate the analytical model. (3) The paper quantifies, using both the analytical model and network traces, the positive impact of communication locality on the interconnection energy dissipation. (4) The paper presents a contention energy analysis and derives a set of closed-form equations for the probability of contention in buses, one-, and two-dimensional networks. A more detailed comparison with related work is presented in Section VIII.

The paper shows that point-to-point interconnection networks have significant energy advantages over bus-based networks. Our framework demonstrates how energy savings depend on the number of nodes in the network and the degree of communication locality. We present our analysis for a one-dimensional point-to-point network and a bus-based network and show that the one-dimensional OCN results in approximately 66% energy savings over a bus for 16 or more processors, even

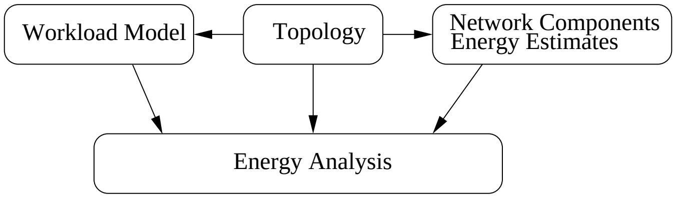


Fig. 1. Energy Analysis Framework. The framework considers the network topology, the workload model, and energy estimates of the network hardware components to calculate the energy dissipated in the interconnection network.

when the communication patterns are uniformly distributed. Increasing the network dimensionality to two results in additional energy savings. We show that for uniformly distributed communication patterns moving from one to two dimensions results in energy savings of $O(\sqrt{N})$, where N is the number of processors in the system. Applications that exhibit communication locality result in significantly greater energy savings. As an example, the results using the analytical model and confirmed with traces of real benchmarks show that the energy of a 2-D OCN can be 10 times better than that of a bus for 16 processors when the applications show communication locality (e.g., ADPCM), and about 5 times better when there is low locality (e.g., btrix).

With the advent of tiled architectures [13], [14], [15], [16], [17], compilers become increasingly responsible for balancing computational parallelism and communication locality. As such, the communication patterns of applications result in widely differing on-chip network energy consumption. Using network traces from a set of benchmarks compiled and run on a tiled processor, we quantify the degree to which applications with greater communication locality are more energy efficient.

The concept of communication locality is increasingly important in two-dimensional realizations of high-dimensional on-chip networks. This is because there is an important trade-off between the energy savings that result from the smaller logical distance between processors and the increased energy dissipation due to the greater wire lengths [6] and hence greater capacitance between processors, as the higher dimensions are mapped into the plane. We investigate this trade-off by comparing the total energy consumption of systems with the same number of processors but different dimensionality. For example, we show that for a chip with 256 processing cores, a 2-D mesh is more energy efficient than a 3-D OCN under the assumption that the energy of the switch logic is no greater than 0.5 times the energy of a network channel connecting a pair of physically adjacent cores.

Contention of network resources results in message delays and increased energy dissipation in the switch, when messages are written into queueing buffers waiting to be serviced by a specific output port. This work examines the effect of contention on the energy dissipated in interconnection networks. We derive a closed-form solution for the energy for various channel utilization values assuming processors communicate with each other with equal likelihood. Using energy estimates for the energy dissipated in the interconnection networks in the Raw microprocessor, we quantify the energy overhead due to contention and show that the maximum amount of additional overhead paid is 23.3%. Additionally, using network traces we estimate the energy dissipation in the communication network for the different applications.

The rest of this paper is organized as follows. Section II describes the development of the framework and presents an analysis of the energy advantages of moving from a bus-based system to a one-dimensional point-to-point interconnected system. Section III examines two-dimensional OCNs and describes their benefits over one-dimensional networks and buses. In Section IV we present an energy analysis based on communication patterns from actual network traces taken from a tiled processor with a two-dimensional point-to-point network. Section V describes energy trade-offs in implementing high-order networks when switch energy and wire lengths are taken into account. Section VI enhances the model to include the effect of contention on the OCN energy. Section VII presents our results and Section VIII discusses related work.

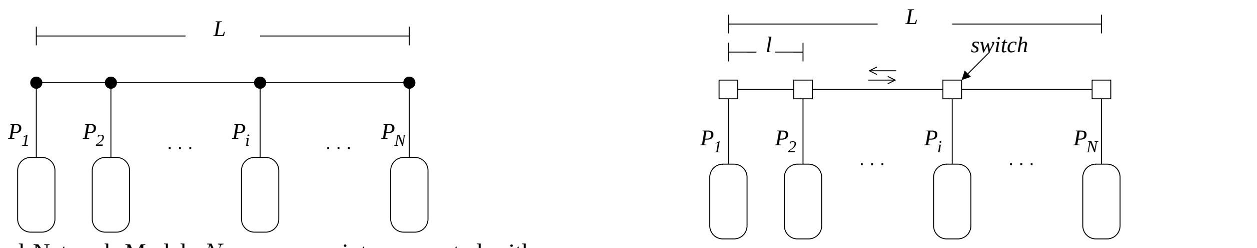
II. Interconnect Energy Analysis Framework

This section presents the energy analysis framework. We perform an analysis of two processor interconnection models, a bus-based model and a point-to-point network (P2P) model. Initially we develop our model for a one-dimensional mesh network (we examine two-dimensional and higher-dimensional networks in Sections III and V) and present the advantages of point-to-point interconnection network systems compared to bus-based systems in terms of energy consumption assuming different network traffic characteristics.

A. Energy Analysis Framework

Fig. 1 presents the components that feed the energy analysis. These components are: a) the network topology b) the workload model, and c) network hardware energy estimates. The topology also feeds the workload model. For example processors might not communicate frequently when they are not located close to each other. The energy of network components depends on the topology too. For example the number of processors in a bus-based system affects the cost for accessing the bus, because the bus wire length is proportional to the number of processors.

The following sections describe each of these components.



(a) Bus-Based Network Model: N processors interconnected with a bus.

(b) Point-to-Point Network Model: N Processors interconnected with a one-dimensional point-to-point network.

Fig. 2. Interconnection Network Models

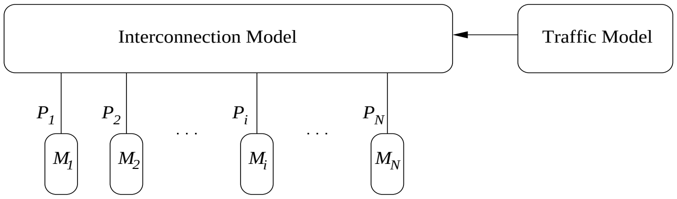


Fig. 3. Workload Model: N Processors, each transmitting M_i data words.

1) Network Topology: The topology describes the interconnection network. The framework can analyze any arbitrary network topology; we present an analysis for buses and one-, two-dimensional, and high-order point-to-point networks. We assume bi-directional channels with no end-around connections for the point-to-point networks.

Bus Interconnection Model: Fig. 2a depicts the simple bus-based model that we use in this analysis. In the bus model, when a processor transmits one data word, the data is available throughout the entire length L of the bus.

Point-to-Point Interconnection Model: Fig. 2b shows the one-dimensional point-to-point model, where N processors are interconnected with a one-dimensional mesh network. The total length of the bus is split into $N-1$ segments, each having length l. When processor P_i wants to send data to processor P_j, the data are available only on the segments that connect P_i with P_j, and each switch is responsible for re-transmitting the data or sending them to the attached processor.

2) Network Components Energy Estimates: In a bus-based system, when a processor puts data on the bus, this data is available to all processors, so the energy cost paid by all processors is the same. In a point-to-point interconnected system, the energy cost of one network hop can be decomposed into the energy cost of accessing the link between two neighboring processors (channel energy) and the energy dissipated in the switch logic (switch energy).

The channel and switch energy are always part of the model, however, at the first part of our analysis, we assume that the switch energy is small compared to the link energy, because we want to examine the effect of communication locality. Section V specifically talks about the switch energy when we discuss high-order networks and Section VI refines the switch energy when we examine contention.

3) Workload Model: Fig. 3 describes our workload model. Our system consists of N processors $(P_1, \cdots, P_N)$. The N processors are connected with an interconnection network and each processor wants to transmit M_i data words. These data words have as their destination other processors in the system. Applications that run on the system result in different data communication among processors. The different communication patterns are described by the traffic model.

We examine six different probability distributions (uniform, linear decay, exponential decay, step, truncated linear decay, and truncated exponential decay) to model the traffic between processors. We investigate the effect of these network traffic distributions on the energy consumption for the point-to-point network model. We want to observe non-localized communication patterns, as well as localized ones. In order to model traffic patterns with various locality characteristics, we apply distributions that allow communication among all processors, as well as distributions that allow communication among processors that are located within a specific radius.

B. Communication Energy Cost In a P2P Network

If E_l is the energy cost of accessing a bus segment for one network hop and $p_{i,j}$[1] is the probability that processor P_i communicates with processor P_j, the expected energy cost $E_{i,j}$ of communicating data from processor P_i to processor P_j, for a P2P network model, is given by

$$E_{i,j} = M_i \cdot p_{i,j} \cdot E_l \cdot H_{i,j}, \tag{1}$$

[1] We define the probability $p_{i,j}$ as the probability that processor P_i communicates with processor P_j, where $p_{i,j}$ satisfies $\sum_{j=1}^{N} p_{i,j|i=I} = 1$, $p_{i,j} = 0$ for $i = j$, and $I = 1, 2, ...N$.

where each element $H_{i,j}$ shows the number of hops the data make when transmitted from processor P_i to processor P_j. Each component of the framework contributes to the above equation: M_i comes from the workload model, the probability $p_{i,j}$ is specified by the traffic model, E_l is given by the hardware energy estimates, and $H_{i,j}$ is specified by the network topology. The product $M_i \cdot p_{i,j}$ gives the expected number of words sent from processor P_i to processor P_j and reveals the traffic patterns in the network. For the remaining analysis, we will assume that all processors in the system want to transmit the same number of words M.

1) Total Communication Energy: The total energy E_{P2P} when all processors have transmitted their data in a point-to-point network is given by Eq. 2

$$E_{P2P} = \sum_{i=1}^{N} \sum_{j=1}^{N} E_{i,j}, \tag{2}$$

where $E_{i,j}$ is the energy consumption due to the transmitted data from processor P_i to processor P_j. The total energy depends on the processor communication pattern. Eq. 2 applies to any one-, two- or high-dimensional point-to-point network.

In a bus-based model the total communication energy (E_BUS) is given by the product of the number of messages sent by each processor, the number of processors in the system , and the energy cost of accessing the bus.

$$E_{BUS} = M \cdot N \cdot E_L, \tag{3}$$

where $E_L = \frac{1}{4} C_L V_{DD}^2$ is the energy cost when a processor transmits one data word, C_L is the total capacitance of the bus, and V_{DD} is the supply voltage. [2]

2) Energy Comparison of the Two Systems for Uniform Distribution: If there is no sense of communication locality in our system, a processor communicates with any other processor with equal probability. Therefore the probability that each of the M data makes i $(i = 1, 2, ...N - 1)$ hops is uniform. So we can replace the communication probability $p_{i,j}$ of Eq. 1 with

$$p_{i,j} = \frac{1}{N - 1} \tag{4}$$

and get the following equation for the expected energy cost for the communication between processors P_i and P_j

$$E_{i,j} = \frac{M}{N - 1} \cdot E_l \cdot H_{i,j}. \tag{5}$$

The total expected energy cost of transmitting the data assuming uniform distribution (from Eq. 2) is

$$E_{P2P} = \sum_{i=1}^{N} \sum_{j=1}^{N} E_{i,j} = \frac{M}{N - 1} \cdot E_l \cdot \sum_{i=1}^{N} \sum_{j=1}^{N} H_{i,j}. \tag{6}$$

Calculating the above expression for a one-dimensional network results in

$$E_{1D} = M \cdot E_l \cdot \frac{N(N + 1)}{3}, \tag{7}$$

which is proportional to the average distance ($\approx N/3$) in such a network with no end-around connections for large N [6]. See [19] for a derivation of the above expression.

The ratio of the energy dissipated on the point-to-point network over the energy dissipated on the bus (Eq. 7 and Eq. 3) is

$$\frac{E_{1D}}{E_{BUS}} = \frac{M \cdot E_l \cdot \frac{N \cdot (N+1)}{3}}{M \cdot N \cdot E_L} = \frac{N + 1}{3} \cdot \frac{E_l}{E_L} \tag{8}$$

The energy cost E_l of accessing the bus segment for one network hop can be broken into two components: E_c the energy for charging one segment of capacitance C_l, and E_s the energy dissipated on the control logic of the switch. Thus,

$$E_l = E_c + E_s. \tag{9}$$

E_s does not appear in the bus equation (Eq. 3) because there is no intermediate switching logic; therefore the energy required to charge the whole bus, E_L, is related to the energy for charging one segment, E_l, in the following manner

$$E_L = (N - 1) \cdot E_c. \tag{10}$$

[2] The total energy supplied by an inverter for a low to high transition is $C_L V_{DD}^2$ [18]. Half of it is stored on the bus line as electrical capacitive energy and the remainder is dissipated as heat in the inverter output resistance. If the bus is high, then no additional energy is required from the driver to pull it high. If the bus line is high and the inverter pulls it low, then there is no additional energy, but the stored energy of the bus line ($C_L V_{DD}^2/2$) is dissipated as heat in the inverter pull-down. If the bus is low and the inverter pulls it low then there is no exchange of energy. The average of these cases is $C_L V_{DD}^2/4$. Even with repeaters the total energy remains the same and only the bus delay changes.

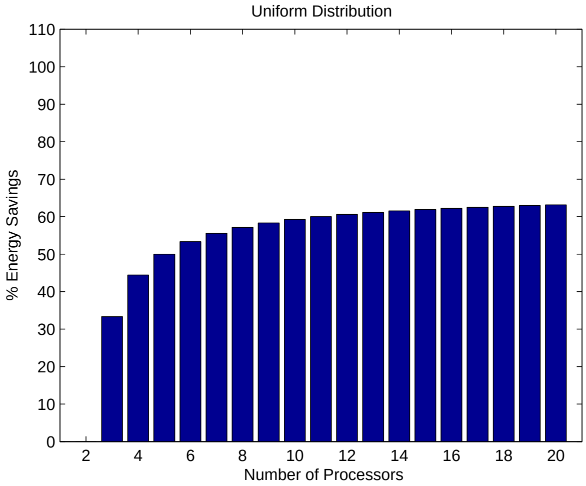


Fig. 4. Energy savings of a one-dimensional P2P network over a bus based system for a uniform distribution traffic model

Eq. 8 becomes

$$\frac{E_{1D}}{E_{BUS}} = (\frac{N+1}{3}) \cdot \frac{E_c + E_s}{E_L} = (\frac{N+1}{3}) \cdot \frac{E_L/(N-1) + E_s}{E_L} = \frac{N+1}{3} \cdot (\frac{1}{N-1} + \frac{E_s}{E_L}). \tag{11}$$

We assume at this point that the energy overhead of the switch is very small, relative to the required energy for accessing the full length of the bus (we investigate the effect of the switch power on the total energy of the network in Section V).

The previous equation becomes

$$\frac{E_{1D}}{E_{BUS}} = \frac{1}{3} \cdot \frac{N+1}{N-1} \approx \frac{1}{3},\ for\ large\ N. \tag{12}$$

Fig. 4 shows the percentage savings for a system based on a one-dimensional point-to-point network with varying number of processors for the uniform distribution. As the number of processors in the system increases, the savings over the bus approach the theoretical limit of 66%.

3) Localized Traffic Distributions: We apply five different probability distributions (linear decay, exponential decay, step, truncated linear decay and truncated exponential decay) to model different communication patterns and locality characteristics. Recall that probability $p_{i,j}$ (Eq. 1) is the probability that processor P_i communicates with processor P_j and that the probability distributions are different for the communication patterns that we examine.

4) Savings Comparison for Different Distributions: Having examined three of the six traffic models, this section compares them all to show the effect of varying degrees of communication locality on network energy, Fig. 5 groups the percentage energy savings of the point-to-point network over the energy consumed in the bus model, for all the six different distributions that we examined.

For the case of the truncated distributions, we select a radius equal to five as the maximum distance for the transmitted data. The additional savings for the truncated distributions (step, truncated linear and truncated exponential) become evident in the systems with seven processors or more. As we move to systems with many processors, the energy savings of a point-to-point interconnection system increase significantly for greatly localized traffic patterns. Different communication locality patterns can have a significant effect on the energy performance of the point-to-point interconnection model. In systems with fifteen processors or more, the energy for the exponential distribution and the truncated distributions is half the energy for the linear distribution.

The bus interconnection model does not take bus contention into account. Even with no contention on the bus it is evident that the energy savings of point-to-point networks are significant. We address contention on point-to-point networks in Section VI.

The probability distributions that we choose to examine present traffic patterns with zero, low, and high locality. We have shown how important communication locality is, so applications that can be parallelized have to exploit it to achieve reduced energy dissipation on the interconnection network. Although applications running on tiled architectures will not have traffic patterns that exactly match the patterns that we examine in this section, we will see in Section IV, when we examine the energy dissipation for actual network traces, that in many cases the communication characteristics can be modeled quite accuratelly with the probability distributions that we examined.

III. Two-Dimensional Interconnection Networks

The analysis thus far assumes that the topology of the point-to-point network is one-dimensional. It is obvious however, that if we increase the network's dimensionality, the total energy savings will increase since the average and maximum communication distance decreases. For example, moving from 1-D to 2-D, the average number of hops decreases from $O(N)$ to $O(\sqrt{N})$.

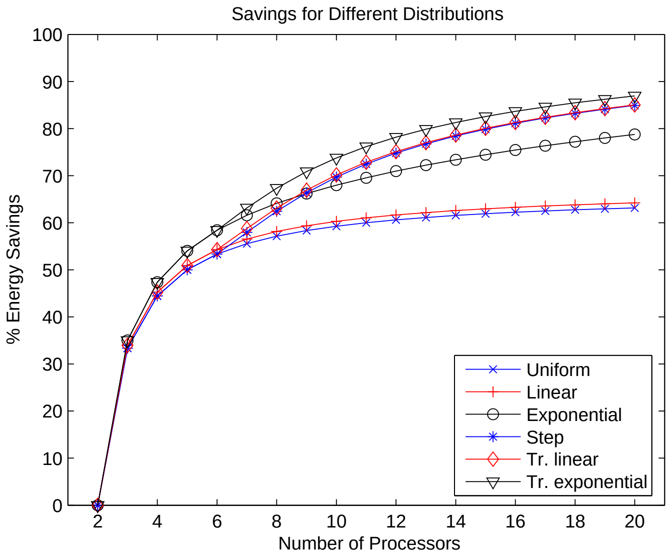


Fig. 5. Energy Savings for All Distributions (1-D vs Bus)

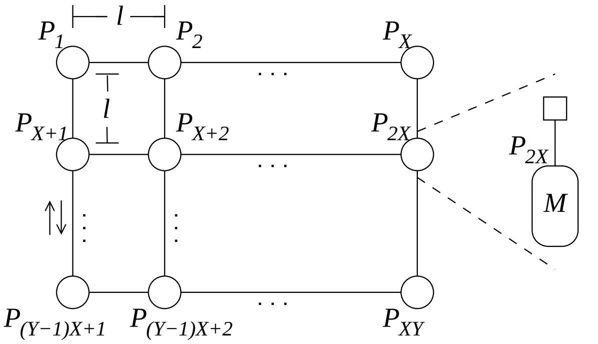


Fig. 6. 2-D Point-to-Point Network Model: N Processors, each transmitting M data on a 2-D mesh network. There are X processor on each row and Y processors on each column.

This section expands the one-dimensional model to include any planar processor arrangement. We present a two-dimensional point-to-point interconnection model. Then we analyze the communication costs for a uniform communication pattern and derive the communication energy savings over the bus-based model.

A. Interconnection Network Model for 2-D Systems

Fig. 6 shows a two-dimensional point-to-point model. The N processors are interconnected with a two-dimensional mesh network (where X is the number of columns and is Y the number of rows in the network). When processor P_i wants to send data to processor P_j, the data are available only at the segments that connect P_i with P_j, and each switch is responsible for re-transmitting the data or sending it to the attached processor. We assume dimension-order routing as a deterministic routing strategy.

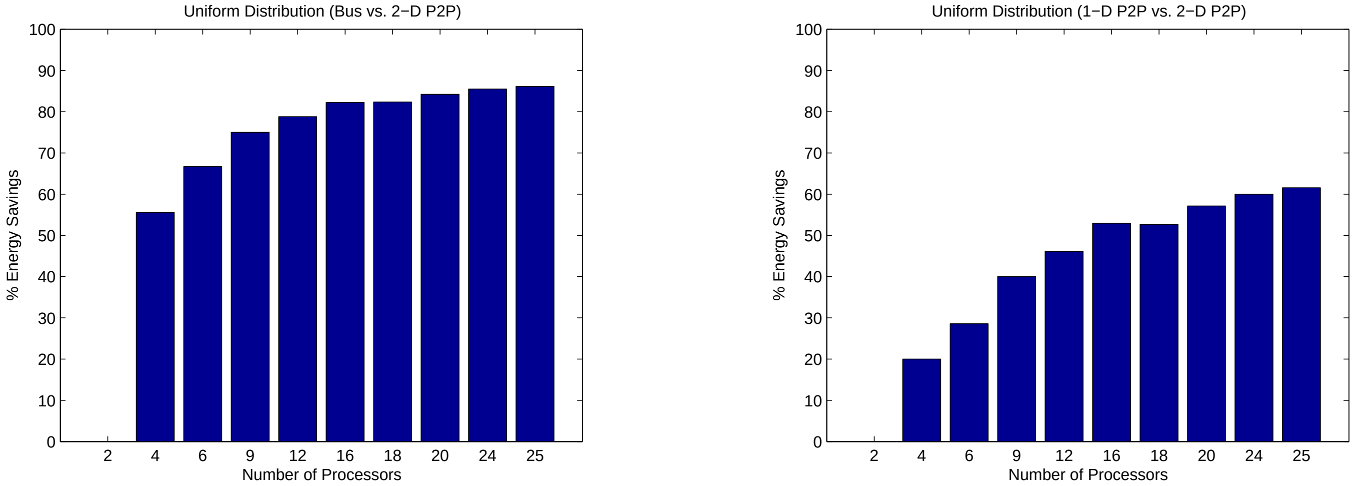
As described in Section II-B.1 the total energy E_{P2P} after all processors have transmitted their data will be given by

$$E_{P2P} = \sum_{i=1}^{N} \sum_{j=1}^{N} E_{i,j}, \tag{13}$$

in which $N = X \cdot Y$ and $E_{i,j}$ is the energy consumption due to the transmitted data from processor P_i to processor P_j. As was the case with the one-dimensional system, the total energy depends on the processor communication pattern and the network topology.

B. Traffic Distribution

In the one-dimensional system we model the traffic patterns of the networks with six probability distributions which describe different localized or non-localized traffic patterns. In this section we generalize the probability distributions to their two-



(a) Energy Savings Assuming Uniform Distribution (Bus vs. 2-D Mesh)

(b) Energy Savings Assuming Uniform Distribution (1-D Mesh vs. 2-D Mesh)

Fig. 7. Energy Savings

dimensional analogs. In the proposed interconnection model we assume equal energy cost moving in each dimension.[3]

1) Uniform Distribution: The uniform communication probability of Eq. 1 is equal to $\frac{1}{XY-1}$, and the expected energy cost for the communication between processors P_i and P_j is given by

$$E_{i,j} = \frac{M}{XY-1} \cdot E_l \cdot H_{i,j}. \tag{14}$$

The total expected energy cost of transmitting the data is

$$E_{2D} = M \cdot E_l \cdot \frac{XY(X+Y)}{3}. \tag{15}$$

The energy ratio of the mesh network over the bus-based model is

$$\frac{E_{2D}}{E_{BUS}} = \frac{1}{3} \cdot \frac{X+Y}{XY-1} \approx \frac{2}{3\sqrt{N}},\ in\ a\ square\ mesh\ for\ large\ N, \tag{16}$$

and shows that for a square mesh network the energy savings are $O(\sqrt{N})$ compared to the energy dissipated on a bus. The graph in Fig. 7a shows the energy savings for the two systems. It is clear that, in the case of a two-dimensional mesh, the limit of the energy ratio in Eq. 16 is zero, while the limit in the one-dimensional network was $1/3$.

The energy savings for a square mesh network compared to a one-dimensional network are also expected to be $O(\sqrt(N)$ for large N, since there is a constant factor of $1/3$ that relates the energy dissipated on a bus based system and a one-dimensional network.

We validate this by comparing the energy of the two point-to-point networks for the same number of processors and plot in Fig. 7b the energy savings of the two-dimensional mesh compared to the one-dimensional mesh

$$\frac{E_{2D}}{E_{1D}} = \frac{M \cdot E_l \cdot \frac{XY(X+Y)}{3}}{M \cdot E_l \cdot \frac{N(N+1)}{3}} = \frac{XY(X+Y)}{N(N+1)}. \tag{17}$$

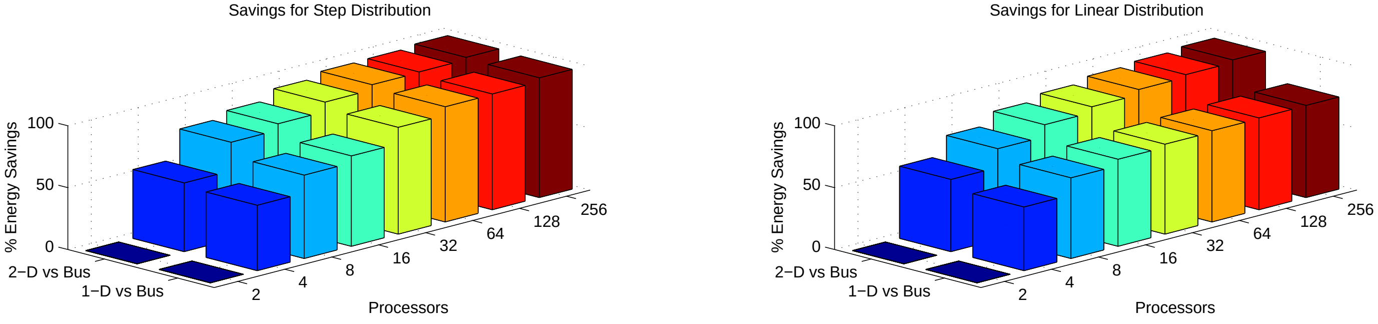
For a square mesh the previous ratio becomes

$$\frac{E_{2D}}{E_{1D}} = \frac{2\sqrt{N}}{N+1} \approx \frac{2}{\sqrt{N}}, for\ large\ N. \tag{18}$$

2) Localized Traffic Patterns: As we did in the case of a one-dimensional point-to-point network, we examine the effect of communication locality on the total energy dissipated in the interconnection network in the case of a two-dimensional network. We modify the probability distributions (communication probability matrix p) that model the communication patterns between two processors in the network to represent different locality characteristics, as we did in the previous section.

Fig. 8 shows the energy savings comparing a two-dimensional point-to-point network with a bus, and a one-dimensional point-to-point network with a bus for two probability distributions and different numbers of processors. In the case of the step distribution (Fig. 8a) there is high locality since communication is not allowed among distant processors, while in the case of the linear decay distribution there is low communication locality and every processor can send data to any other processor in the network.

[3] The framework can easily be modified to account for unequal energy costs moving in different dimensions.



(a) Energy Savings Assuming Step Distribution

(b) Energy Savings Assuming Linear Decay Distribution

Fig. 8. Energy Savings Comparing a 2D and 1D P2P Network with a Bus

In the first case, the energy savings of the two-dimensional network are comparable to the savings of the one-dimensional network, while in the case where there is low locality, the savings for the two-dimensional network are significant compared to the one-dimensional network. This suggests that two-dimensional point-to-point networks are more energy efficient for the same number of processors over a wider variety of traffic patterns compared to one-dimensional networks.

IV. Energy Savings Using Network Traces

So far in our analysis, we used probability distributions to model different traffic patterns and get the expected energy in the various networks that we examined. This section describes how we incorporate actual network traces in our framework to get an estimation of the energy dissipated on the on-chip interconnection network and validate our model.

The network traces were obtained from the MIT Raw group. The traces for the fourteen benchmarks[4] were produced using a cycle-accurate tiled processor simulator for four different processor configurations: two tiles, four tiles, eight tiles, and sixteen tiles. The compiler [20], [21] schedules for ILP and arranges sequential C or Fortran programs across the tiles in two steps. First, the compiler distributes the data and code across the tiles balancing locality and parallelism. Then, it schedules the computation and communication to maximize parallelism and minimize communication latency.

On each cycle we recorded the communication among the tiles and obtained the network traces for traffic across the tiles. Knowing the communication information between the tiles we write Eq. 1 as

$$E_{i,j} = E_l \cdot T_{i,j} \cdot H_{i,j}, \tag{19}$$

where $T_{i,j}$ replaces $m \cdot p_{i,j}$ and describes the actual number of data words that processor P_i sent to processor P_j, and captures the locality patterns specific to each benchmark.

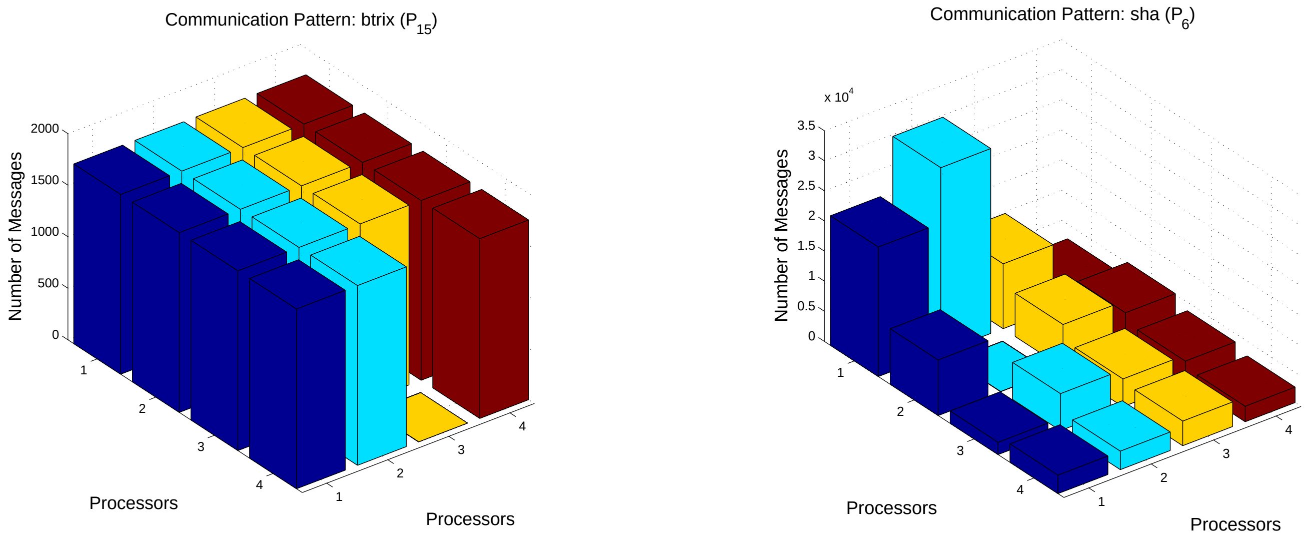
For example, Fig. 9a shows the number of messages originating from processor $P_1 5$ (the one on the fourth row and the third column) for the *btrix* benchmark. In this case, the communication pattern resembles a uniform distribution. On the other hand in Fig. 9b we plot the data transfers originating from processor P_6 (the one on the second row and the second column) for the *sha* benchmark. In this case, the compiler exploits communication locality, making the largest number of data transfers have as their destination a neighboring tile.

We investigate the locality characteristics for each benchmark. Fig. 10a presents the average distance in network hops of communication between processors for all the benchmarks on a four-by-four tile configuration. The benchmark with the most localized communication pattern is *adpcm* and the communication patterns resembled in most cases a truncated exponential decay and a step distribution. The benchmark with the least localized pattern is *mxm*. In this case we observed permutation traffic patterns [3] directed to non-neighboring tiles, which explains why the average distance was greater than the average distance for the uniform distribution, which is 2.67 hops for a four-by-four mesh.[5]

Fig. 10b shows the percentage energy savings for each benchmark on the two-dimensional Raw mesh compared to the energy dissipated if the processors were interconnected through a bus. It is evident how the average distance relates to the energy savings. The energy savings per application are inversely proportional to the average distance traveled for the application. *adpcm* with the lower distance shows the biggest energy savings, while *mxm* with the largest average distance shows the least energy savings.

[4] The benchmarks and the corresponding sources are: adpcm (Mediabench), swim (Spec95), tomcatv, btrix, cholesky, mxm, vpenta, fpppp (Nasa7:Spec92), jacobi, jacobi_big, life (Raw bench. suite), sha (Perl Oasis), aes, aes_fix (FIPS-197).

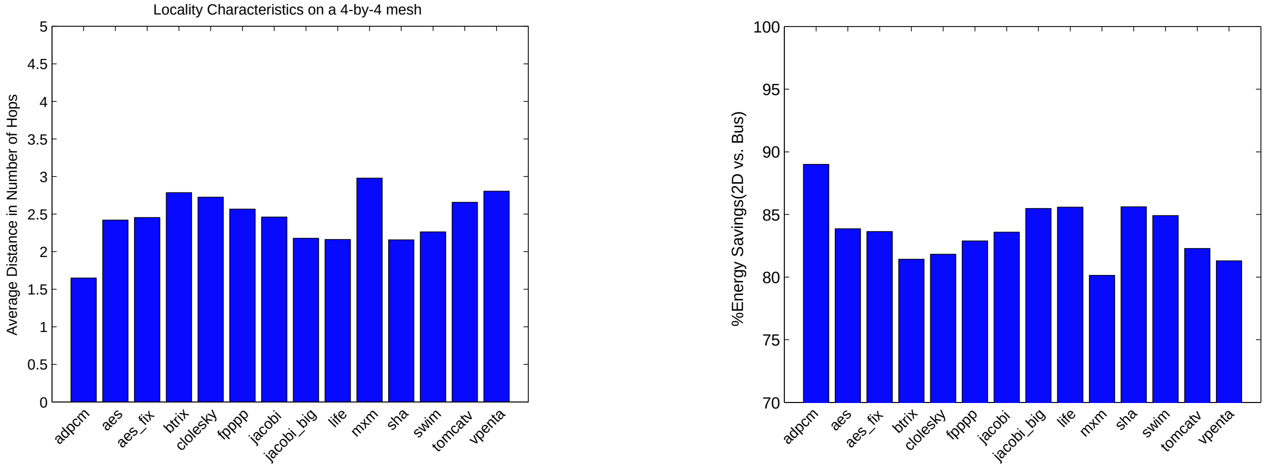
[5] The average distance in an n-dimensional network with no end-around connections is given by the product of the dimensions n and the radix k divided by 3 ($\frac{n \cdot k}{3}$) [6].



(a) Communication Pattern - btrix

(b) Communication Pattern - sha

Fig. 9. Network Traces



(a) Average Distance on a 4-by-4 mesh

(b) Energy Savings for a 4-by-4 mesh over bus

Fig. 10. Benchmark Locality Characteristics

Using Eq. 19 we calculate the total communication energy for each benchmark and compare it to the energy for the same communication pattern assuming a bus-based model. Fig. 11 shows the energy savings of the mesh networks over the bus-based network for the fourteen benchmarks that we run on the four different processor configurations (2-tiles, 4-tiles, 8-tiles, and 16-tiles).

On the 4-tile configuration *jacobiLbig* shows the least energy savings of 52%. On the 8- and 16-tile configurations *mxm* shows the least energy savings of 68% and 80%, respectively. On all configurations adpcm shows the most energy savings. An interesting observation is that for different tile configurations the benchmarks do not always exhibit proportional energy savings. For example, when we observe the energy savings for *swim* and *tomcatv*, we see that on a 4-tile system the energy savings of *tomcatv* are higher compared to *swim*. However, on the 8- and 16-tile system *swim* exhibits higher energy savings. Another similar case appears when we compare the energy savings for an 8-tile and 16- tile system for *tomcatv* and *vpenta*. On the 8-tile system the energy savings for vpenta are higher. However, this case changes when the two benchmarks run on 8 and 16 tiles.

The reason for this phenomenon has to do with the way the compiler distributes the data across the tiles, schedules the computation within each tile, and schedules the communication among tiles. Some applications exhibit more parallelism as the number of tiles increases and less communication among the tiles.

Fig 12 shows the energy savings for the various benchmarks using trace data along with the energy savings estimated by the analytical framework assuming uniform, linear and exponential decay distribution probabilities. The average distance for the linear decay probability (with parameters $b = 14$ and $a = 2$) is 2.32 hops for the sixteen processor system and best matches *life* and *jacobi*. The exponential decay probability distribution (with parameters $b = 5.5$ and $d = 0.5$) yields greater energy savings with average distance 1.71 hops for the sixteen processor case and matches *adpcm*.

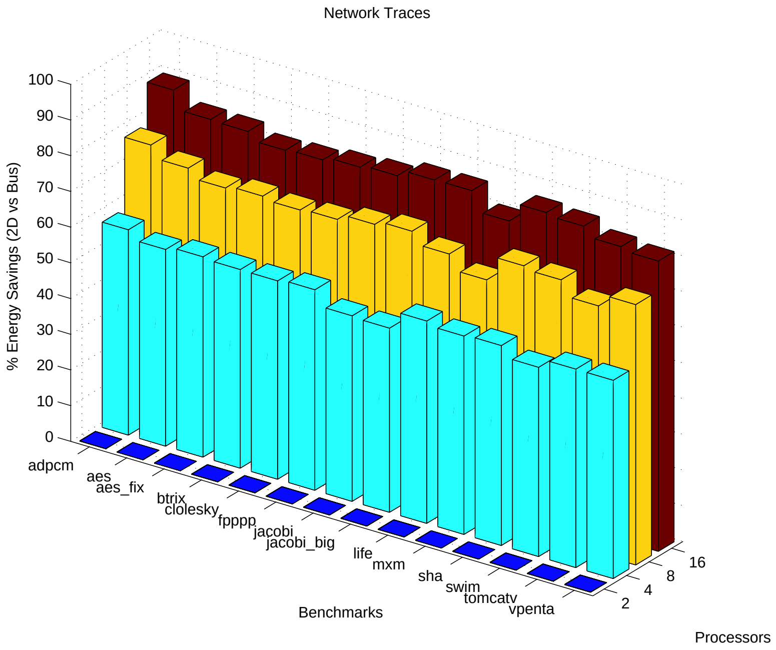


Fig. 11. Energy savings for two-dimensional P2P network over bus for various number of tiles

This analysis shows that while applications running on tiled architectures exhibit different traffic patterns, if we can estimate what the communication among tiles would look like, we can model these communication patterns using probability distributions and have interconnection network energy reports using our framework without running simulations to record the network traces.

V. Switch Energy and Multi-Dimensional Networks

This section introduces the switch energy and multi-dimensional networks into the framework.

A. Switch Energy

When we consider the energy dissipated on the switch control logic the expected energy cost $E_{i,j}$ (Eq. 1) of transferring the data from processor P_i to processor P_j, when every link of the network has the same length, is given by

$$E_{i,j} = m \cdot p_{i,j} \cdot E_l \cdot H_{i,j} = m \cdot p_{i,j} \cdot (E_c + E_s) \cdot H_{i,j}, \tag{20}$$

where E_c is the energy cost of accessing one segment that connects two neighboring processors and E_s is the energy dissipated on the switch.

B. Switch Energy and Varying Wire Lengths

When high-dimensional networks are mapped to two dimensions the wire lengths are different for each dimension. In that case, the expected energy cost $E_{i,j}$ (Eq. 20) of transferring the data from processor P_i [6] to processor P_j is given by

$$E_{i,j} = m \cdot p_{i,j} \cdot (E_c \cdot D_{i,j} + E_s \cdot H_{i,j}), \tag{21}$$

where $D_{i,j}$ is the physical length of wire (in multiples of l) which is traversed between processors P_i and P_j, after the network is mapped into two dimensions, and $H_{i,j}$ corresponds to the logical distance on multiple dimensions between processors P_i and P_j. High-dimensional networks are expected to suffer higher switch energy E_s compared to two-dimensional networks. At this point however, we assume that the switch energy E_s of the networks that we examine is the same. [NOTE TO REVIEWER: In the final version of the paper we will obtain real values for both E_s and E_c from real processors like the Raw [13] and Trips [16] microprocessors and use them in our analysis.]

The total expected energy cost of transmitting the data when all higher dimensions are mapped to two dimensions is

$$E_{P2P} = \sum_{i=1}^{N} \sum_{j=1}^{N} E_{i,j} = m \sum_{i=1}^{N} \sum_{j=1}^{N} [p_{i,j}(E_c \cdot D_{i,j} + E_s \cdot H_{i,j})]. \tag{22}$$

An algorithm that calculates matrices D and H is given in [19].

[6] We make the assumption that the processor IDs increase along the first, then second, then third etc. dimensions.

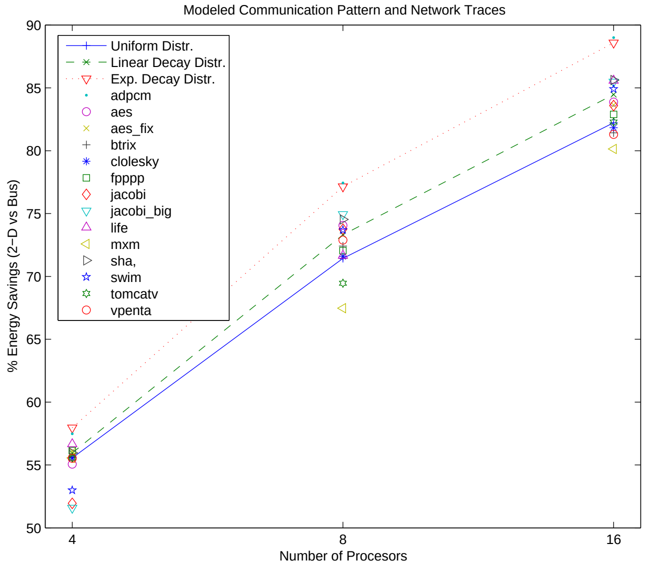
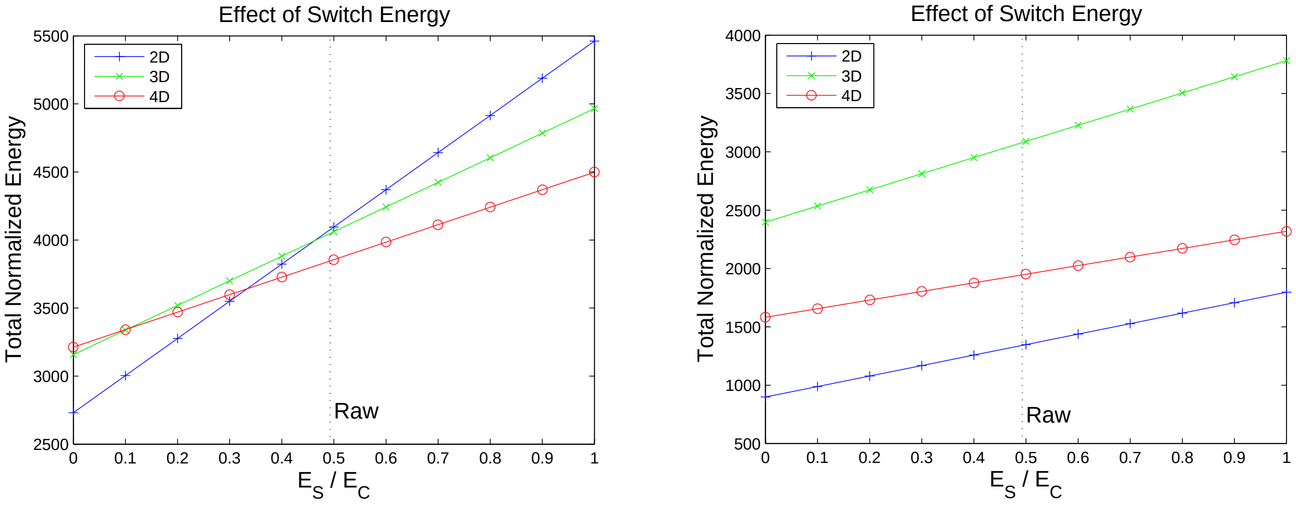


Fig. 12. Energy Savings (2-D vs Bus) for Benchmarks and Modeled Communication Patterns



(a) Uniform Distribution

(b) Exponential Decay Distribution

Fig. 13. Energy Comparison for 2-D, 3-D and 4-D Networks

C. Energy Comparison for High-Dimensional Networks

We compare the total energy of three point-to-point networks: a two-dimensional network with processors laid out in an array of sixteen rows and sixteen columns, a three-dimensional network with twelve processors on the first dimension, seven processors on the second dimension and three processors on the third dimension[7], and a four-dimensional network with 4 processors on each dimension. Using Eq. 22 we calculate the total normalized energy (for $m = 1$ and $E_c = 1$) for the three networks assuming uniform and exponential decay distributions to investigate the energy consumption of those networks when there is no or high locality in the communication among processors.

The average logical and physical distance for the uniform and exponential decay distributions is shown in Table I. With varying switch energy the trade-offs between the physical and logical distance are evident in Fig. 13. Fig. 13a plots the total energy assuming uniform distribution. When the switch energy, E_s, is small relative to E_c, the most efficient network is the two-dimensional because of the smallest physical distance compared to the other networks. The high-order networks, on the other hand, show smaller logical distance so when E_s increases, the portion of the total energy dissipated on the switch becomes significant. Because of the smaller logical distance of the four-dimensional network the total energy of the two-dimensional network exceeds the total energy of the four-dimensional network at a smaller value of E_s compared to the three-dimensional

[7] We choose this arrangement although it results in less processors than the other two systems (252 vs 256) because it provides an efficient trade-off between the physical and logical distance when a three-dimensional network is laid out in two dimensions. See the Appendix for details on the mapping algorithm.

TABLE I

AVERAGE PHYSICAL AND LOGICAL DISTANCE FOR UNIFORM AND EXPONENTIAL DECAY DISTRIBUTIONS

Dimensions	Phys. Dist.	Log. Dist.	Phys. Dist.	Log. Dist.
2-D	10.67	10.67	3.51	3.51
3-D	12.53	7.18	9.51	5.49
4-D	12.55	5.02	6.18	2.88

network.

However, two-dimensional networks are more energy efficient when communication locality exists in the traffic patterns. Fig. 13b plots the total normalized energy for the three networks assuming an exponential decay probability distribution. The average physical distance for the two-dimensional network is much smaller compared to the average distance of the four-dimensional network and the two networks have comparable logical distance values because of the high communication locality of the exponential decay distribution, as shown in Table I. From this analysis, there is an evident connection between the effect of the switch to the total energy, and the effect of switch delays to latency of interconnection network discussed in [6], where two-dimensional networks have the lowest latency when switch delays are ignored, but higher-dimensional networks are favored otherwise. As is the case for the energy, when communication locality exists two-dimensional networks regain their advantage.

The graph also reveals that high-order networks with dimensionality that is not power of two, are not energy efficient when they are mapped into two dimensions. The reason is that there can never be an arrangement that minimizes both the physical and the logical distances. Minimizing the logical distance requires as similar as possible number of processors on each dimension and minimizing the physical distance requires a planar realization of the network with the number of processors on the two dimensions as similar as possible.

VI. CONTENTION ENERGY ANALYSIS FOR MULTI-DIMENSIONAL NETWORKS

This section presents an analysis of the effects of network contention on the total energy dissipation in on-chip interconnection networks. We present a high-level schematic of a typical network router and show the major components for energy dissipation. Then we expand our framework to include the effect of network contention and the resulting energy dissipated on the network queuing buffers. We examine contention in the interconnection network when processors communicate with equal likelihood and present energy estimates for different channel utilization values. We derive a closed-form solution for the probability of contention in one-, two-dimensional networks and buses. Finally, we validate our model using network traces from benchmarks running on the Raw multicore processor.

A. Switch Model

The model assumes that the switch has a buffer associated with every input port at each network dimension as well as a buffer for messages generated from the processor of the same node. When multiple packets request the same output port in a cycle, the control logic arbitrates among them allowing only one message to be transmitted on the output port and causing the other messages to be queued in their respective input queuing buffers. Fig.14 shows the high-level microarchitecture of a typical on-chip network switch similar to the one used in the Raw multicore processor. The switch consists of input and output ports, input buffers, a crossbar, and control logic circuitry. The input and output ports correspond to the processor port and north, east, south, and west directions of the node.

In the schematic we depict the energy costs that are essential to our analysis. E_C is the energy consumed at the channels that connect two neighboring nodes as before. E_C consists of the energy dissipated on the long wires that connect an output port of the switch to the corresponding input port on the neighboring switch and the energy dissipated on the repeaters that are used to drive the signals (for example in Raw there are two repeater stages).

E_T is the energy dissipated on the crossbar and the switch control logic circuitry that determines whether a message is consumed at the node, changes or continues in the same direction, or gets queued at the input buffer. The energy expended in the crossbar is dominated by wiring capacitances that connect the inputs with the outputs through various multiplexors, while the energy of the control logic is expended in the gates that perform the logic functions.

E_Q is the energy expended when writing and reading the message in the input buffer queue (energy lumped together for a FIFO write and subsequent read). Input buffers have a bypass path that a message takes in the case when the queue is empty and the output port is not servicing any other requests. The input buffer in a Raw switch is a 4-entry, 32-bit wide FIFO.

Using extracted capacitance values from the Raw layout and the Raw netlist we estimated the energy costs of these components in Raw for one network hop. For the networks in the Raw microprocessor the values for the energy costs of the different components in the switch are $E_C = 34.5pJ$, $E_S = 17pJ$, $E_Q = 12pJ$. These numbers assume independent and identical data sequences. (In the Appendix we describe the methodology for the estimation of these energy values).

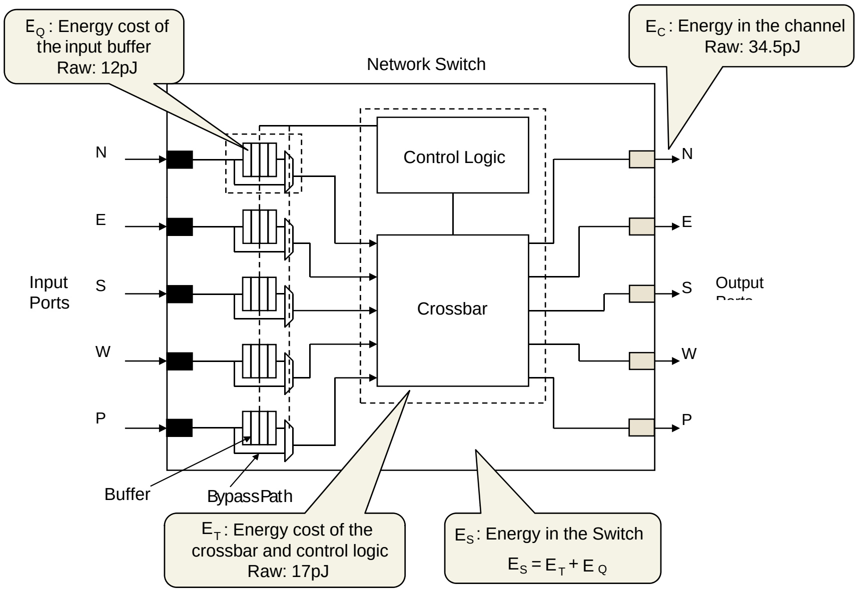

E_Q : Energy cost of the input buffer Raw: 12pJ
Network Switch
E_C : Energy in the channel Raw: 34.5pJ
Control Logic
Crossbar
N
E
S
W
P
Input Ports
Output Ports
Buffer
Bypass Path
E_T : Energy cost of the crossbar and control logic Raw: 17pJ
E_S : Energy in the Switch
$E_S = E_T + E_Q$

Fig. 14. Network Switch

B. The Life of a Message in the Network

In the absence of contention on the output port, the message goes through the bypass path[8] within the input queueing buffer to the crossbar. In this case, the energy cost for moving a message between two neighboring processors is

$$E_l = E_C + E_T. \tag{23}$$

On the other hand, when the output port is not free, the message is queued into the input buffer and is read when the output port has finished servicing other messages. Therefore the above equation becomes

$$E_l = E_C + E_T + E_Q. \tag{24}$$

At any given switch, the energy overhead of queueing the message into the input buffer depends on the probability the message is delayed due to contention. Thus, we can model the expected energy for a network hop as

$$E_l = E_C + E_T + q \cdot E_Q. \tag{25}$$

where the probability q describes the likelihood that a message arriving at a switch will suffer contention and will be stored in the queue buffer. For a message that is injected into the network and has to travel some number of hops to its destination, there is a lower and upper bound on the energy expended due to contention.

For a message from P_i to P_j the upper energy bound is

$$E_{max} = (E_C + E_T + E_Q) \cdot H_{i,j}, \tag{26}$$

when the message suffers contention at every switch in its path ($q = 1$).

The lower bound is given by

$$E_{min} = (E_C + E_T) \cdot H_{i,j}, \tag{27}$$

in the absence of contention in the network ($q = 0$).

The analysis so far suggests that preventing the energy overhead of storing messages into the input queue buffers requires careful scheduling by the compiler. Specifically, compilers need to schedule message injections into the network after assuring a minimum number of conflicts for resources for the entire message path. From the energy standpoint, a message should wait as many cycles as possible in a buffer until contention is minimized at every stage of the total route.

[8] We assume that the bypass energy is negligible

C. Calculation of Contention Probability

We can calculate the probability q of contention at the switch in a given cycle in the case of uniform communication patterns among the processors in the system for a message entering the switch requesting an output port. The probability q is a function of the number of messages that request the given output port on any cycle and the state of the queue corresponding to the input port on which the message arrived, whether it is empty or not. The analysis that describes the calculation of the contention probability for the uniform distribution is performed under the condition that a message is entering the switch through one of the input ports.

If the queue is not empty, any new message entering the switch will contend and will be stored in the queue. This probability is described by the probability there is at least one message in the queue.

If the queue is empty, the probability the message will contend depends on the total number of messages requesting the same output port. We define υ as the random variable that describes the number of messages that request a single output port. If the message is the only one requesting the output and the input queue is empty, it will not contend for it. With an empty queue, there is probability of contention when $\upsilon \geq 2$ and that probability is $(\upsilon - 1)/\upsilon$, since every message has equal probability of $1/\upsilon$ to be serviced.

Therefore, we can write the probability q of contention on any given cycle as

$$q = \left[\begin{array}{c}\text{probability queue}\\ \text{is not empty}\end{array}\right] + \left[\begin{array}{c}\text{probability queue}\\ \text{is empty}\end{array}\right] \text{ x } \left[\begin{array}{c}\text{probability the message}\\ \text{is not routed out}\end{array}\right]. \tag{28}$$

Equivalently,

$$q = p_{|Q| \neq 0} + p_{|Q|=0} \cdot \sum_{\upsilon=2}^{\infty} p(\upsilon) \cdot \frac{\upsilon - 1}{\upsilon}, \tag{29}$$

where

- $p_{|Q|=0}$ is the probability the queue is empty.
- $p_{|Q| \neq 0} = 1 - p_{|Q|=0}$ is the probability the queue has *at least* one message.
- $p(\upsilon)$ is the probability υ messages request that specific port.

For M/M/1, M/G/1, and M/D/1 systems the probability of an empty queue is $p_{|Q|=0} = 1 - \rho$ ([22], [23]), where ρ is the channel utilization and describes the probability of a message arriving at an incoming channel.

Thus, Eq. 29 becomes

$$q = \rho + (1 - \rho) \cdot \sum_{\upsilon=2}^{\infty} p(\upsilon) \cdot \frac{\upsilon - 1}{\upsilon}. \tag{30}$$

Now, using the derivation in [19] for $\rho(\upsilon)$ we can substitute for it and re-write Eq. 30 for the 2-d point-to-point, 1-d point-to-point, and bus network topologies.

1) Contention Probability in Two-Dimensional Networks:

$$q_{2D} = \rho + (1 - \rho) \cdot \rho^2 / 2nk_d. \tag{31}$$

2) Contention Probability in One-Dimensional Networks:

$$q_{1D} = \rho + (1 - \rho)\rho^2(k_d - 1)/(2k_d^2). \tag{32}$$

3) Contention Probability in Bus-Based Networks:

$$q_{BUS} = \rho + (1 - \rho) \sum_{\upsilon=2}^{N} \binom{N}{\upsilon} m^{\upsilon}(1 - m)^{N-\upsilon} \cdot \frac{\upsilon - 1}{\upsilon} \tag{33}$$

In the above equations, k_d is the average number of hops each message travels in each dimension, for a total of nk_d hops. m is the message injection rate to the network by the processor. N is the number of processors in the system.

So far our analysis provided closed-form equations for the probability of contention in two- dimensional networks, one-dimensional networks, and bus-based systems (Eq. 31, Eq. 32, and Eq. 33, respectively). Next we compare these probability values with respect to the message injection rate of the processors in the system. Fig. 15 plots the probability of contention for the three systems with 16 processors in each, assuming the same message injection rate. For the same injection rate, the probability of contention in a bus-based system is 25 times greater compared to probability of contention in the two-dimensional network, and almost 6 times greater compared to the probability of contention in the one-dimensional network. Comparing the two point-to-point networks, the probability of contention in the one-dimensional network is 4.2 times greater compared to the probability of contention in the two-dimensional network.

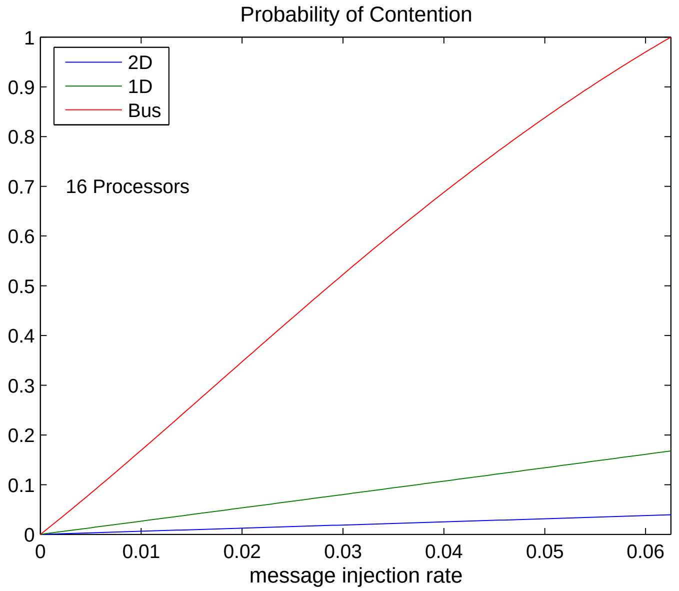


Fig. 15. Probability of Contention for a Two- and One-Dimensional Network, and a Bus-Based System.

VII. Energy Analysis Results

The previous section calculates the contention probabilities for the point-to-point networks and the bus. Using the analysis for contention, this section presents energy results in a two- and one-dimensional point-to-point network and a bus.

A. *Energy-Delay Product Assuming Uniform Distribution*

Table II presents the average energy costs (base and contention) for a message in a bus, a one-, and two-dimensional network, for a system with 16 and 64 processors and various message injection rate values. As the message injection rate (m) increases the networks reach the maximum channel utilization ($\rho = 1$) at different values of m. At that point the contention energy is maximized and every message injected in the network is stored in the input queuing buffer of every switch along the path from the originating processor to the destination processor. In the 16-processor systems the contention energy is maximized when $m > 0.25$ and $m > 0.1$ for the one-dimensional network and bus, respectively. In the 64-processor system the contention energy is maximized when $m > 0.1$ and $m > 0.035$ for the one-dimensional network and bus, respectively.

TABLE II

Energy for a two-, one-dimensional, and a bus-based network per message sent for different values of the message injection rate m

					m					
16 Processors	0.01	0.035	0.06	0.1	0.25	0.4	0.5	0.6	0.7	0.8
2D base (nJ)	0.137	0.137	0.137	0.137	0.137	0.137	0.137	0.137	0.137	0.137
2D contention (nJ)	0.0002	0.0007	0.0012	0.002	0.005	0.008	0.01	0.012	0.014	0.016
1D base (nJ)	0.292	0.292	0.292	0.292	0.292	0.292	0.292	0.292	0.292	0.292
1D contention (nJ)	0.0018	0.0063	0.0108	0.0452	0.068	0.068	0.068	0.068	0.068	0.068
Bus base (nJ)	0.535	0.535	0.535	0.535	0.535	0.535	0.535	0.535	0.535	0.535
Bus contention (nJ)	0.002	0.0672	0.01152	0.012	0.012	0.012	0.012	0.012	0.012	0.012
64 Processors										
2D base (nJ)	0.275	0.275	0.275	0.275	0.275	0.275	0.275	0.275	0.275	0.275
2D contention (nJ)	0.0008	0.0029	0.0050	0.0084	0.021	0.0336	0.042	0.0504	0.0588	0.064
1D base (nJ)	1.116	1.116	1.116	1.116	1.116	1.116	1.116	1.116	1.116	1.116
1D contention (nJ)	0.0277	0.097	0.1664	0.26	0.26	0.26	0.26	0.26	0.26	0.26
Bus base (nJ)	2.191	2.191	2.191	2.191	2.191	2.191	2.191	2.191	2.191	2.191
Bus contention (nJ)	0.0077	0.012	0.012	0.012	0.012	0.012	0.012	0.012	0.012	0.012

The network throughput is a linear function of the message injection rate (m), until the channel utilization reaches its maximum value ($\rho = 1$). After this point the network throughput remains constant.

This is shown in Fig. 16. The message throughout in a one-dimensional point-to-point network in a system with 64 processor follows the message injection rate until $m = 0.093$ and then it is held constant at 0.093. Fig. 16 also depicts the inverse of throughput which shows the delay between two injected messages. As the injection rate increases the delay between two injected messages by the processor decreases until again $\rho = 1$.

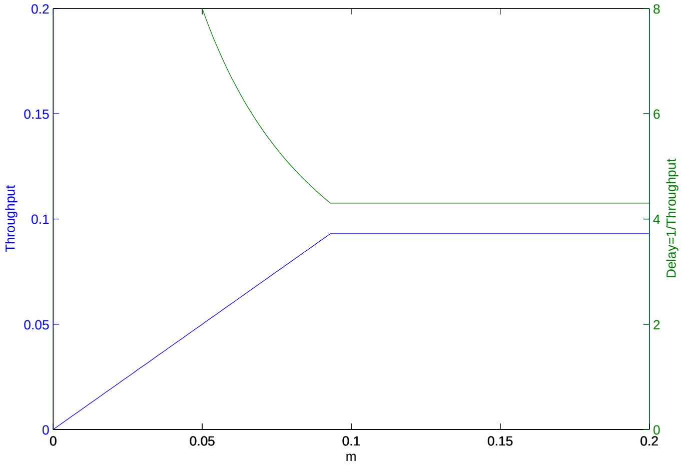


Fig. 16. Throughput and Delay in a one-dimensional point-to-point network in a 64-processor system.

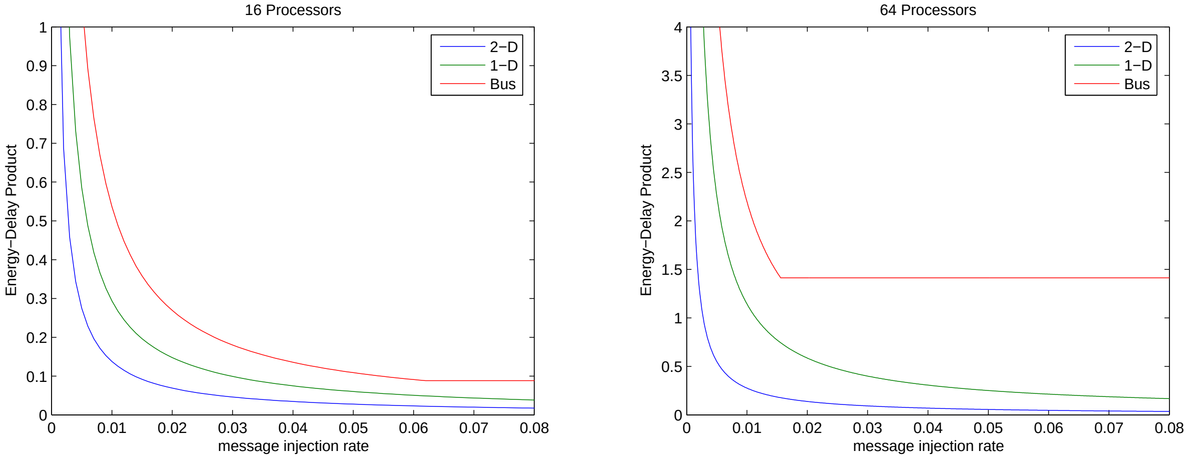


Fig. 17. Energy-Delay product in a two- and one-dimensional point-to-point network and a bus in a system with 16 and 64 processors assuming a uniform distribution.

We can use the energy predicted by the model and the delay between two messages to calculate the energy-delay product per message for the different networks and for different values of the message injection rate. Fig. 17 shows the energy-delay product for a two- and a one-dimensional point-to-point network and a bus in a system with 16 and 64 processors. It is clear from the graph that from the energy-delay standpoint it is beneficial to keep injecting messages into the network. This is mainly because the energy overhead that is paid when a message is stored in the input queuing buffer of the switch is fixed and small compared to the energy dissipated in the other network components.

B. Energy Consumption Using Network Traces

The previous section presented energy results for a two- and one-dimensional point-to-point network, and a bus assuming different values of the processor message injection rate (hence different values of the channel utilization), when processors communicate with each other with equal likelihood.

This section presents energy results using network traces from real applications running on the Raw tiled microprocessor. These traces are collected on both the dynamic and static networks of Raw. All applications are run on a 16-tile processor configuration.

We used the Raw cycle-accurate simulator to collect detailed network traces. Specifically, our traces indicated when a message entering a tile is stored into the input queue buffer. The total number of the messages stored in buffers along their path from the origin to destination provides information about the energy overhead due to contention.

We want to compare the total energy for each benchmark with the total energy predicted by the analytical model for the uniform distribution assuming similar channel utilization values and see the additional energy added by a non-uniform traffic

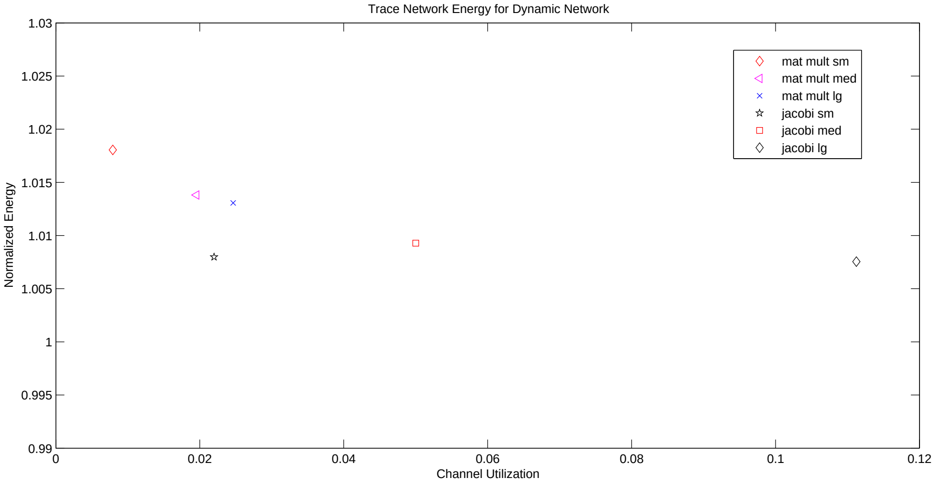


Fig. 18. Energy overhead due to contention on real benchmarks running on dynamic network collected on simulator.

pattern into the network. The graphs in the following sections plot the normalized energy ($E_{Contention}/E_{No\_Contention}$) for each benchmark and the expected normalized energy for the uniform distribution.

With the information in the network traces we estimate the message injection rate for each benchmark using the following equation:

$$m_t = \frac{Total\_Number\_of\_Messages}{Duration\_of\_Comm \cdot Number\_of\_Processors \cdot Average\_Message\_Length}. \tag{34}$$

1) Dynamic Network Traces: Fig. 18 shows the normalized communication energy overhead due to contention of messages on the Raw dynamic network. The points on the plot show the energy overhead due to network contention for the jacobi relaxation and matrix multiplication benchmarks, each with three different inputs sizes. These benchmarks were run using rMPI [24], the MPI library built for Raw. As expected, the channel utilization is greater for the larger input sizes, with the largest jacobi benchmark achieving the maximum channel utilization on this plot of 0.15. This can be attributed to the average message lengths sent for each of the benchmarks, which is larger for the larger benchmarks. For example, the averge message length for jacobi for the small, medium, and large input sizes are 21 words, 24 words, and 30 words, respectively. Note that the rMPI system breaks up logical messages larger than 31 words into 31 word messages, which is the largest message size allowed by Raw's dynamic network.

In general, matrix multiply exhibits higher contention than jacobi due to the manner in which data is distributed in each algorithm. With jacobi, only near neighbors need to communicate. However, with matrix multiply, data distribution from and reduction to single core induce contention. Accordingly, the matrix multiplication benchmarks have the lowest channel utilization but the highest energy overhead due to contention. With an improved matrix multiplication algorithm, the channel utlization could likely improve. While the algorithm was not adjusted for this evaluation, it is clear that such analysis is valuable in pinpointing such energy (and performance) inefficiencies.

2) Static Network Traces: Fig. 19 shows the normalized communication energy overhead due to contention of messages on the Raw static network. For this case we had to modify the analytical model to account for the unique characteristics of the Raw static networks.

The blue solid line shows the energy overhead predicted by the analytical model for the static network assuming a uniform distribution for the communication among processors. The red solid line is a first order polynomial fit on the energy overhead data points for each benchmark.

The graph reveals that only mxm and $life$ have energy overhead higher than the predicted one for the uniform distribution. The communication patterns revealed that a few processors in the system were responsible for a large percentage of the total communication, which explains the high overhead values due to contention. Specifically, in mxm we noticed that the communication was heavily targeted on two processors.

In the Raw static network most of the contention is suffered within the processor; the static scheduler holds data in the tile before it ensures that the path to destination is as clear as possible. This is the main reason the fitted curve falls below the values of the overhead predicted by the analytical model.

VIII. Related Work

Power and energy of on-chip networks have become a significant portion of the total power and energy budget [8], so studies that explore these issues are important for processor designers. Recently, there have been studies that address several important

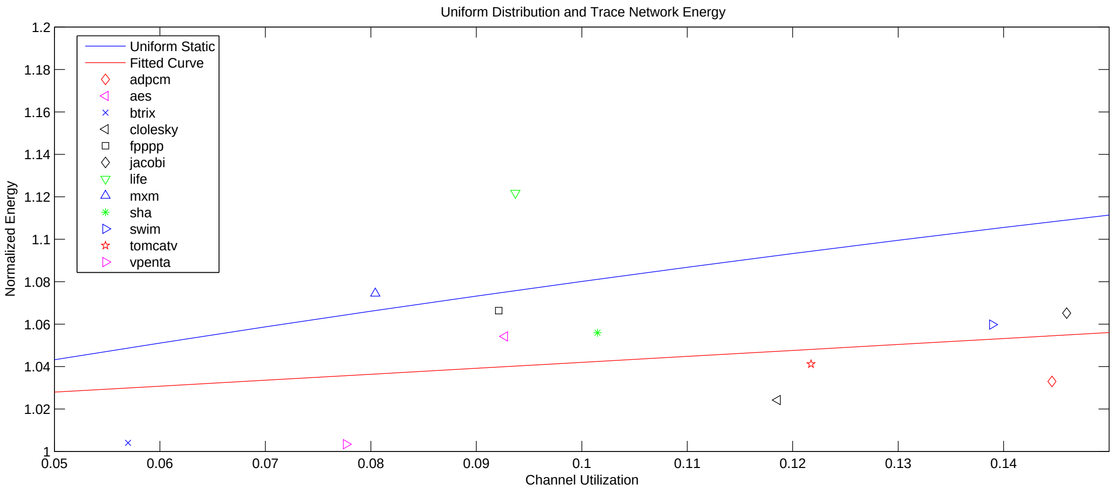


Fig. 19. Energy overhead due to contention assuming both uniform communication distribution (analytical model) and real benchmarks running on static network collected on simulator.

issues.

To our knowledge, Orion [9] was the first power/performance interconnection network simulator, capable of providing detailed power and performance reports. The simulator is triggered on every cycle by different activities and calculates the network power based on power models for each on the components of the network. Orion simulates various network fabrics and workloads, providing power and performance network characteristics. Our results, derived both from an analytical model and traces of real benchmarks (that are gathered from a single execution run), provide energy estimates for various types of communication traffic and can be used early in the design cycle, with no need for time-consuming architectural simulation. Further, models provide a different level of insight compared to a simulator, particularly as they relate to asymptotics and in many cases can provide an intuitive basis from which energy characteristics may be extrapolated to any benchmark without simulating it.

Eisley and Peh [10] addressed the need for high-level network power analysis by proposing a framework that uses link utilization as the unit of abstraction for network power. The analysis takes as input the message flows across nodes, the specific topology and the routing protocol to derive a power profile of the network. Instead, our framework uses hop count as the unit of abstraction to measure the energy efficiency of the network. Additionally, our model takes communication locality into account. Link utilization is a valid unit of abstraction, however (1) hop-counts and (2) probability distributions that model traffic patterns assist intuition on communication locality, especially since the applications that we studied have patterns that roughly match the modeled distributions. Link utilization cannot infer communication locality since different locality patterns can produce the same link utilization. Communication locality is not addressed in the work of Eisley and Peh. The framework we present can be used to derive link utilizations from locality distributions or real traces. We further extract information about traffic patterns and locality from traces of benchmarks running on a tiled processor and use the results both to validate our model and to assess the impact of communication locality. Furthermore, the analysis of Eisley and Peh does not take into account the effect of buffering at each individual node. This buffering occurs at the switch when an output port is not free and the message has to wait in the input queue buffer for the port to be available. In this work we present a detailed analysis of the effect of contention and the overhead of message queueing to the total energy dissipated in the interconnection network. Additionally, this work correlates the energy results to real switches (Raw microprocessor) and the energy consumption of the switch hardware components.

Wang et al. [11] present a method to evaluate the energy efficiency of different two-dimensional square network topologies and predict the most energy-efficient network topology based on the network size, technology predictions, and architectural parameters. The methodology uses the *average* flow control unit traversal energy as the network energy efficient metric. Our model is based on precise distributions. Therefore it can (1) capture arbitrary topologies, (2) incorporate actual network traces, and (3) use arbitrary analytical distributions. We believe our work is the first to address communication locality in analyzing multicore interconnect energy and to use real multicore interconnect traces extensively. Further, the results presented in Wang assume uniform distributions, which we showed that is not enough to choose the most energy efficient network topology. Our work is not limited to one distribution to model traffic patterns. Rather we present a framework that can incorporate any traffic pattern, including those from real traces, and use many distributions to model various forms of communication locality. Interestingly, for some simple cases, we show that our distribution-based approach asymptotically yields the same results as an approach using averages. One of our major results is that network energy is heavily related to communication locality, which was not captured in Wang's paper. Additionally, contention of network resources is not addresses in that work. We have presented an detailed analysis on the probability of contention in the interconnection network and the effect it has on the energy dissipation. Our framework can be used for the energy estimation for high- dimensional networks for any mapping the compiler chooses, with subsequent changes of the model parameters.

Moreover, the real appeal of our approach lies in the ease of incorporating actual network traces in our framework. Additionally, our work provides energy comparison of several interconnection networks with buses. Buses are currently popular in commercial multicore chips and therefore merit special attention.

Kim et al. [12] present measurements of energy consumption in an experimental tiled processor containing a mesh OCN. This work was an experimental study of a real tiled processor. In their study, they present measurements of energy consumption in the Raw microprocessor and examine the energy costs of communication over the two types of network in the Raw multicore processor. The measured numbers for the static and dynamic networks are $85pJ$ and $90pJ$, respectively. Both numbers are measured for a maximum toggle-rate per word; consecutive words injected to network would cause the channel lines to alternate on every cycle, so these values correspond to the maximum energy dissipation per cycle. Our work estimates energy consumption of $86pJ$/hop for the dynamic network when we assume maximum toggle rate.

IX. Conclusions

This paper develops an analytical framework for estimating the energy of on-chip interconnection networks. It presents results for the energy of networks such as buses, one-, two-, and high-dimensional OCNs based on both analytical communication models and real network traces from benchmarks running on a tiled multicore processor.

The paper makes the following major contributions. First, it develops an analytical framework for the energy analysis of OCNs and compares the energy of one-, two-, and multi-dimensional OCNs (mapped to a two-dimensional physical on-chip substrate) to that of buses. The model includes the impact of switch energy, wire lengths and related capacitance, communication locality, and network contention.

Second, the paper uses real network traces from benchmarks running on a tiled microprocessor to compare the energy performance of OCNs, and to validate the analytical model. As an example, the paper shows that the analytical model using a uniform communication distribution provides results that are similar to that of the application *tomcatv* (the data arrays are distributed evenly across all the tiles).

Third, the paper quantifies using both the analytical model and network traces the positive impact of communication locality. As an example, the results using the analytical model and confirmed with traces of real benchmarks show that the energy of a two-dimensional OCN can be 10 times better than that of a bus for 16 processors when the applications show communication locality (e.g. *adpcm*), and about 6 times better when the communication follows a uniform distribution (e.g. *btrix*).

Fourth, this paper presents an contention energy analysis. Contention of network resources results in message delays and increased energy dissipation in the switch, when messages are written into queueing buffers waiting to be serviced by a specific output port. We examined the effect of contention on the energy dissipated in interconnection networks and derived a closed-form solution for the energy for various channel utilization values assuming processors communicate to each other with equal likelihood. We used energy estimates for the energy dissipated in the interconnection networks in the Raw microprocessor to quantify the energy overhead and showed that the maximum amount of additional overhead paid is 23.3%. Additionally, we showed that the energy-delay product per message decreases as the processor message injection rate into the network increases.

X. Appendix

A. Calculation of Network Components Energy Costs

This section describes the methodology for estimating the energy costs E_C, E_T, and E_Q that we use in our analysis. We follow a low level approach in our methodology based on capacitance values from the Raw microprocessor dynamic networks. For wiring and metal capacitance values, we use the extracted capacitance values generated by the IBM ChipEdit capacitance extractor tool for the final layout of the Raw microprocessor [25]. For the cell input and output capacitances we use the values provided by IBM for their cells in the SA-27E process.

1) Link Capacitance Energy Cost Estimation: The energy contributing capacitances of the data path (the output of one tile to the input of a neighboring one) consist of:

a) The metal capacitance of the metal wires. We obtained those values from the extracted capacitance information based on the final layout of Raw. The IBM ChipEdit tool reports for every single node of the chip the best and worst case capacitance values, $25.5pF$ and 16.4 [25], respectively.

b) The input capacitance of the inverters[9].

c) The internal and output capacitance of the inverters.

Taking into account those contributing components the equivalent capacitance for a 32-bit data transfer is $47.8pF$ and $38.7pF$ for the worst and best case respectively, resulting in an average energy cost of $34.5pJ$ and a maximum cost of $69pJ$ per transfer for an approximate path length of $4mm$ (for a 0.18u Technology and 1.8V power supply).

[9] The input and output capacitance values for the IBM standard cells are intentionally not listed, because those values are IBM's proprietary information

2) Crossbar and Control Logic Energy Cost Estimation: Our switch model lumps together the energy dissipated on the crossbar and the control logic circuitry, since this energy is always expended when a message enters the switch.

a) Crossbar - The crossbar energy consists of the energy for the propagation of the data signals from the output of the input queueing buffer multiplexor to one of the output ports of the switch; and the energy dissipated on the multiplexors and drivers (input, internal, and output capacitances) that direct the data to the input of the neighboring tile. The average energy expended for a message going through the crossbar from west to east is $10.01pJ$.

b) Control Logic - The control circuitry consists of the input and output control logic. The input control logic provides the logic signals for the selection of the data at the input queue buffer multiplexors, generates request signals for the output control logic, and signals to the output logic the end of the message. The input control has counters and comparators that determine the direction of the message (whether the message has to travel more hops on the X or Y directions, or it has reached its destination).

This logic is responsible for requesting service to the output control logic for a specific input port. Some major components include six flip-flops, eight comparators, and a counter. The average energy expended in this block is $3.47pJ$.

The output control logic maintains the control of the destination for the data signals. It controls the output multiplexors for the network port and performs the scheduling for the output. The output logic handles all requests for the output port and is responsible for the logic signals that maintain the route requested by a message until the whole message is transmitted and detects any new message requests based on information from the input control logic.

Additionally, the output control logic includes the logic circuitry that implements a five input random function for the arbitration among the five possible (four input ports and the processor) requests. The output control logic consists mainly of logic gates and four flip- flops that hold the state of the current route of a message (3 bits for choosing among five possible routes) and another state that reveals whether a route is planned or not. The average energy expended in the output control logic block is $3.64pJ$.

3) Input Queue Buffer Energy Cost Estimation: The last component of the switch is the input queue buffer. This is the energy cost expended when there is contention for the output port and the message needs to be stored until it can be serviced. The Raw input buffer queue is a 4-entry deep, 32-bit wide FIFO. The total energy cost consists of the energy to write the data in one of the four positions and the energy for reading the stored data.

The flip-flops are standard-cell master-slave implementations with different clocks for the two stages (master and slave). The input data are either propagated through the multiplexer or stored in the flip-flops. The logic signals for the multiplexor selection are provided by the input control logic block.

A significant portion of the total energy is due to the distribution of the two clocks to the two inputs of the flip-flops for the 32 data bits. Moreover, from the available standard-cell implementations in the IBM library, the flip-flops in the Raw network input buffers are the ones that correspond to the highest performance level, resulting in high input, internal, and output cell capacitances. The energy cost to write new data and read the data from the input buffer in Raw is $12pJ$.

B. One-Dimensional Networks - Derivation of the Model

1) Uniform Distribution: The total expected energy cost of transmitting the data (Eq. 6) is

$$E_N = \sum_{k=1}^{N} E_k = \sum_{i=1}^{N}(\sum_{j=1}^{N} E_{i,j}) = \frac{m}{N-1} E_l \sum_{i=1}^{N}\sum_{j=1}^{N} H_{i,j} \tag{35}$$

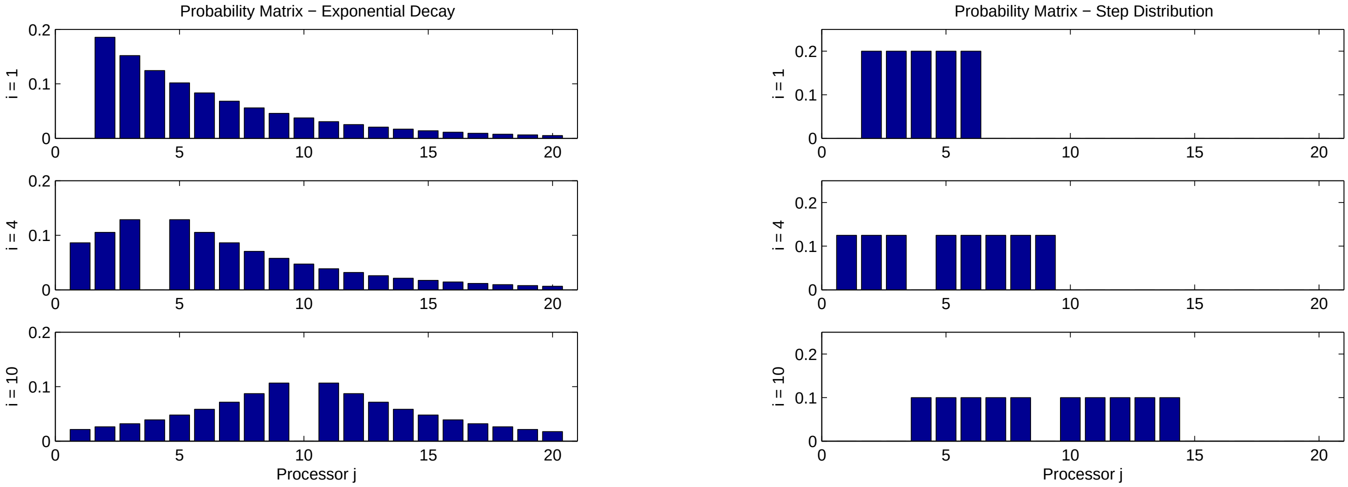
For a one-dimensional network matrix with no end-around connections H is given by

$$H = \begin{bmatrix} 0 & 1 & \cdots & N-2 & N-1 \\ 1 & 0 & \cdots & \cdots & N-2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ N-2 & \cdots & \cdots & 0 & 1 \\ N-1 & N-2 & \cdots & 1 & 0 \end{bmatrix} \tag{36}$$

Matrix H (Eq. 36) is symmetric, so the sum of its elements is the sum of the elements of the upper triangle multiplied by two.

$$\sum_{i=1}^{N}\sum_{j=1}^{N} H_{i,j} = 2 \cdot \sum_{i=1}^{N-1}(N-i)\cdot i = 2\sum_{i=1}^{N-1}(N\cdot i - i^2) = 2(\sum_{i=1}^{N-1} N\cdot i - \sum_{i=1}^{N-1} i^2) = \tag{37}$$

$$= 2(N\sum_{i=1}^{N-1} i - \sum_{i=1}^{N-1} i^2) = 2(N[\frac{N}{2}(N-1)] - [\frac{N}{6}(N-1)(2N-1)]) = \tag{38}$$



(a) Exponential Decay Probability Distribution

(b) Step Probability Distribution

Fig. 20. Probability Distributions

$$= 2N(N-1)(\frac{N}{2} - \frac{2N}{6} + \frac{1}{6}) = \frac{N(N-1)(N+1)}{3} \tag{39}$$

and the total expected energy cost is (Eq. 35) is

$$E_N = \frac{m \cdot E_l}{N-1} \cdot \frac{N(N-1)(N+1)}{3} = m \cdot E_l \cdot \frac{N(N+1)}{3} \tag{40}$$

2) Localized Distribution Models: The non-zero elements of the probability matrices are shown for each distribution. (The parameters a, b, and d of the equations determine the shape of the probability distributions)

Linear Decay

$$p_{i,j} = \frac{|b - (a \cdot |i-j|)|}{\sum_{j=1}^{i-1} |b - a(i-j)| + \sum_{j=i+1}^{N} |b - a(j-i)|}, i \neq j \tag{41}$$

Exponential Decay

$$p_{i,j} = \frac{\frac{1}{d} \cdot b^{-\frac{|i-j|}{d}}}{\sum_{j=1}^{i-1} \frac{1}{d} \cdot b^{-\frac{i-j}{d}} + \sum_{j=i+1}^{N} \frac{1}{d} \cdot b^{-\frac{j-i}{d}}}, i \neq j \tag{42}$$

Step Distribution

$$p_{i,j} = \begin{cases} 1/(r+i-1) & , \quad i \leq r \\ 1/2r & , \quad r < i < N-r \\ 1/(r+N-i) & , \quad N-r \leq i \end{cases} \tag{43}$$

for $|i-j| \leq r$, $i \neq j$ and $N > 2r$

Truncated Linear Decay

$$p_{i,j} = \frac{|b - (a \cdot |i-j|)|}{\sum_{j=i-r}^{i-1} |b - a(i-j)| + \sum_{j=i+1}^{i+r} |b - a(j-i)|}, i \neq j \tag{44}$$

Truncated Exponential Decay

$$p_{i,j} = \frac{\frac{1}{d} \cdot b^{-\frac{|i-j|}{d}}}{\sum_{j=i-r}^{i-1} \frac{1}{d} \cdot b^{-\frac{i-j}{d}} + \sum_{j=i+1}^{i+r} \frac{1}{d} \cdot b^{-\frac{j-i}{d}}}, i \neq j \tag{45}$$

Figures 20 and 21 present the probability distribution of the communication between two processors in a system with twenty processors for three different processors $(i = 1, i = 4, i = 10)$ for the exponential decay, step, and truncated linear decay distributions.

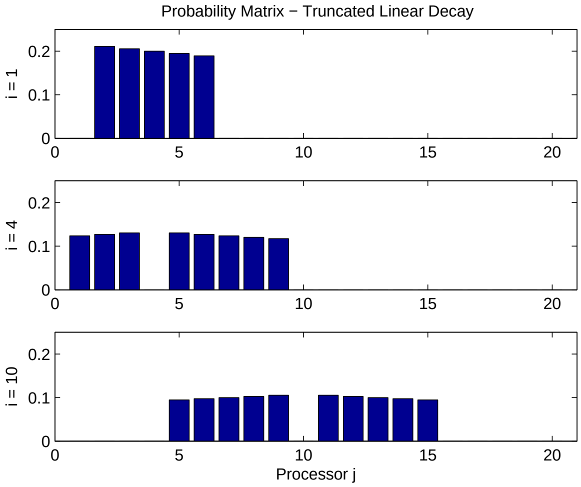


Fig. 21. Truncated Linear Decay Probability Distribution

C. Two-Dimensional Interconnection Networks

Each element of matrix H in Eq. **??** gives the number of hops between two processors as they are laid out in the plane. The processor i.d.s increase column-wise. H is an $rc-by-rc$ symmetrical matrix and looks like:

$$H = \begin{bmatrix} H_X & H_X + A & \cdots & H_X + A(Y-1) \\ H_X + A & H_X & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ H_X + A(Y-1) & \cdots & \cdots & H_X \end{bmatrix} = \tag{46}$$

$$= \begin{bmatrix} H_X & H_X & \cdots & H_X \\ H_X & H_X & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ H_X & \cdots & \cdots & H_X \end{bmatrix} + \begin{bmatrix} 0 & A & \cdots & (Y-1)A \\ A & 0 & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ \cdots & \cdots & \cdots & 0 \end{bmatrix} = \overline{H} + \overline{A} \tag{47}$$

where

$$H_X = \begin{bmatrix} 0 & 1 & \cdots & X-1 \\ 1 & 0 & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ X-1 & \cdots & 1 & 0 \end{bmatrix} \tag{48}$$

shows the distance between two processors in the same column in the system and

$$A = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}. \tag{49}$$

Therefore,

$$\sum_{i=1}^{N}\sum_{j=1}^{N} H_{i,j} = \sum_{i=1}^{N}\sum_{j=1}^{N} \overline{H}_{i,j} + \sum_{i=1}^{N}\sum_{j=1}^{N} \overline{A}_{i,j} = \tag{50}$$

$$= Y^2 \sum_{i=1}^{X}\sum_{j=1}^{X} H_{Xi,j} + (\sum_{i=1}^{X}\sum_{j=1}^{X} A_{i,j}) \cdot sum \begin{bmatrix} 0 & 1 & \cdots & Y-1 \\ 1 & 0 & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ \cdots & \cdots & 1 & 0 \end{bmatrix} \tag{51}$$

where $sum[M] = \sum H_{i,j}$.

So

$$\sum_{i=1}^{N}\sum_{j=1}^{N} H_{i,j} = Y^2 \cdot \frac{X(X-1)(X+1)}{3} + X^2 \cdot \frac{Y(Y-1)(Y+1)}{3} = \tag{52}$$

$$= \frac{XY[Y(X-1)(X+1) + X(Y-1)(Y+1)]}{3} = \frac{XY[Y(X^2-1) + X(Y^2-1)]}{3} \tag{53}$$

$$= \frac{XY[YX^2 - Y + XY^2 - X)]}{3} = \frac{XY[Y(XY-1) + X(XY-1)]}{3} = \frac{XY(XY-1)(X+Y)}{3}, \tag{54}$$

and the total expected energy cost for the uniform distribution is

$$E_N = \frac{m \cdot E_l}{N-1} \cdot \sum_{i=1}^{N}\sum_{j=1}^{N} H_{i,j} = \frac{m \cdot E_l}{XY-1} \cdot \frac{XY(XY-1)(X+Y)}{3} = m \cdot E_l \cdot \frac{XY(X+Y)}{3}. \tag{55}$$

D. Switch Energy

We present an algorithm implemented in Matlab that calculates matrices D and H for four-dimensional networks.

```
   function [D H] = four_d(d1,d2,d3,d4)
% FOUR_D Physical and logical distance for a 4-D network mapped to plane.
%     D(i,j) returns the physical distance from processor $P_i$ to
%     processor $P_j$ for a 4-d mesh after it is mapped to 2-d.
%     H(i,j) returns the logical distance from processor $P_i$ to
%     processor $P_j$ for a 4-d mesh.

   N = d1*d2*d3*d4;
   [X,Y,Z,W] = ndgrid(1:d1,1:d2,1:d3,1:d4);
   x = repmat(X(:),1,N);
   y = repmat(Y(:),1,N);
   z = repmat(Z(:),1,N);
   w = repmat(W(:),1,N);
   s = sort([d1 d2]);
   D = abs(x-x')+abs(y-y')+abs(z-z')*s(1)+abs(w-w')*s(2);
   H = abs(x-x')+abs(y-y')+abs(z-z')+abs(w-w');
```

REFERENCES

[1] M. Horowitz and W. Dally, “How scaling will change processor architecture,” in *Proceedings of the International Solid-State Circuits Conference*, 2004, pp. 132–133.

[2] G. Moore, “No exponential is forever: but ”forever” can be delayed,” in *Proceedings of the International Solid-State Circuits Conference*, 2003, pp. 20–23.

[3] W. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003.

[4] J. Duato, S. Yalamanchili, and L. Ni, *Interconnection Networks*. Morgan Kaufmann Publishers Inc., 2002.

[5] W. J. Dally, “Performance analysis of k-ary n-cube interconnection networks,” *IEEE Trans. Comput.*, vol. 39, no. 6, pp. 775–785, 1990.

[6] A. Agarwal, “Limits on interconnection network performance,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 2, no. 4, pp. 398–412, 1991.

[7] V. S. Adve and M. K. Vernon, “Performance analysis of mesh interconnection networks with deterministic routing,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 5, no. 3, pp. 225–246, 1994.

[8] V. Raghunathan, M. Srivastava, and R. Gupta, “A survey of techniques for energy efficient on-chip communication,” in *DAC*, 2003, pp. 900–905.

[9] H.-S. Wang, X. Zhu, L.-S. Peh, and S. Malik, “Orion: a power-performance simulator for interconnection networks,” in *MICRO 35: Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture*. Los Alamitos, CA, USA: IEEE Computer Society Press, 2002, pp. 294–305.

[10] N. Eisley and L.-S. Peh, “High-level power analysis for on-chip networks,” in *CASES ’04: Proceedings of the 2004 international conference on Compilers, architecture, and synthesis for embedded systems*. New York, NY, USA: ACM Press, 2004, pp. 104–115.

[11] H. Wang, L.-S. Peh, and S. Malik, “A technology-aware and energy-oriented topology exploration for on-chip networks,” in *Design, Automation and Test in Europe*, March 2005, pp. 1238–1243.

[12] J. Kim, M. B. Taylor, J. Miller, and D. Wentzlaff, “Energy Characterization of a Tiled Architecture Processor with On-Chip Networks,” in *2003 ISLPED*, 2003, pp. 424–427.

[13] M. B. Taylor, J. Kim, J. Miller, D. Wentzlaff, F. Ghodrat, B. Greenwald, H. Hoffman, J.-W. Lee, P. Johnson, W. Lee, A. Ma, A. Saraf, M. Seneski, N. Shnidman, V. Strumpen, M. Frank, S. Amarasinghe, and A. Agarwal, “The Raw Microprocessor: A Computational Fabric for Software Circuits and General-Purpose Programs,” *IEEE Micro*, pp. 25–35, Mar 2002.

[14] K. Mai, T. Paaske, N. Jayasena, R. Ho, W. J. Dally, and M. Horowitz, “Smart Memories: A Modular Reconfigurable Architecture,” in *2000 ISCA*, 2000, pp. 161–171.

[15] S. Swanson, K. Michelson, A. Schwerin, and M. Oskin, “Wavescalar,” in *MICRO 36: Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture*. Washington, DC, USA: IEEE Computer Society, 2003, p. 291.

[16] K. Sankaralingam, R. Nagarajan, H. Liu, C. Kim, J. Huh, N. Ranganathan, D. Burger, S. W. Keckler, R. G. McDonald, and C. R. Moore, “Trips: A polymorphous architecture for exploiting ilp, tlp, and dlp,” *ACM Trans. Archit. Code Optim.*, vol. 1, no. 1, pp. 62–93, 2004.

[17] J. Oliver, R. Rao, P. Sultana, J. Crandall, E. Czernikowski, L. JonesIV, D. Franklin, V. Akella, and F. T. Chong, “Synchroscalar: A multiple clock domain, power-aware, tile-based embedded processor,” in *International Symposium on Computer Architecture*, Jun 2004.

[18] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective, 2nd Edition*. Prentice Hall, 2003.

[19] T. K. Konstantakopoulos, "Energy scalability of on-chip interconnection networks," Ph.D. dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, June 2007.

[20] W. Lee, R. Barua, M. Frank, D. Srikrishna, J. Babb, V. Sarkar, and S. Amarasinghe, "Space-time scheduling of instruction-level parallelism on a raw machine," in *ASPLOS-VIII: Proceedings of the eighth international conference on Architectural support for programming languages and operating systems*. New York, NY, USA: ACM Press, 1998, pp. 46–57.

[21] M. Taylor, W. Lee, J. Miller, D. Wentzlaff, I. Bratt, B. Greenwald, H. Hoffman, P. Johnson, J. Kim, J. Psota, A. Saraf, N. Shnidman, V. Strumpen, M. Frank, S. Amarasinghe, and A. Agarwal, "Evaluation of the raw microprocessor: An exposed-wire-delay architecture for ilp and streams," in *International Symposium on Computer Architecture*, Jun 2004.

[22] Kleinrock, *Queueing Systems*. New York: Wiley, 1975.

[23] E. Gelenbe and G. Pujolle, *Introduction to Queueing Networks*. John Wiley & Sons, 1998.

[24] J. Psota and A. Agarwal, "rmpi: Message passing on multicore processors with on-chip interconnect," in *High-Performance Embedded Architecture and Compilation*, Jan 2008.

[25] "MIT RAW Group, Personal Communication," 2007.