A Technology-aware and Energy-oriented Topology Exploration for On-chip Networks

Hangsheng Wang Li-Shiuan Peh Sharad Malik
Department of Electrical Engineering,
Princeton University, Princeton, NJ 08544
{hangshen,peh,sharad}@princeton.edu

ABSTRACT

As packet-switching interconnection networks replace buses and dedicated wires to become the standard on-chip interconnection fabric, reducing their power consumption has been identified to be a major design challenge. Network topologies have high impact on network power consumption. Technology scaling is another important factor that affects network power since each new technology changes semiconductor physical properties. As shown in this paper, these two aspects need to be considered synergistically.

In this paper, we characterize the impact of process technologies on network energy for a range of topologies, starting from 2-dimensional meshes/tori, to variants of meshes/tori that incorporate higher dimensions, multiple hierarchies and express channels. We present a method which uses an analytical model to predict the most energy-efficient topology based on network size and architecture parameters for future technologies. Our model is validated against cycle-accurate network power simulation and shown to arrive at the same predictions. We also show how our method can be applied to actual parallel benchmarks with a case study. We see this work as a starting point for defining a roadmap of future on-chip networks.[1]

1. INTRODUCTION

On-chip interconnection networks have been proposed [5, 6] and used [13, 15] in the past few years to replace buses and dedicated wires as the standard on-chip interconnection fabric. While many design methods inherited from chip-to-chip networks and computer cluster networks are performance-driven, the power impact of on-chip networks is becoming increasingly important [2, 9, 17, 18].

Network topologies define how nodes are connected to each other. More specifically, network topologies determine the number of hops and the wire length involved in each data transmission, both critically influencing the energy cost per transmission. At the same time, technology scaling not only leads to a globally increasing/decreasing trend of total network power, it also changes the power composition of different network components. As a result, an optimal topology at current technology may no longer be optimal after several technology generations. With rapid on-chip network adoption, arriving at the most suitable topology for the current process is not enough – it is critical for network designers to understand the impact of technology on network topologies and be able to predict the most energy-efficient topology for future technologies based on this understanding.

In this paper we provide the basis for this prediction by exploring the energy efficiency of several network topologies in a technology-aware manner. We use the average flit traversal energy as our metric to evaluate network energy efficiency. We then formulate this metric with a uniform analytical model which can be applied to a range of topologies and be used to quantitatively compare the energy efficiency of these topologies across multiple technologies.

The rest of the paper is organized as following: In Section 2 we explain the motivation to look at topologies and technology scaling. In Section 3 we describe our assumptions, models and metrics. Then in Section 4 we study the energy properties of four candidate topologies: 2-D meshes/tori, high-dimensional meshes/tori, hierarchical meshes/tori and express cubes. Next, we demonstrate the applicability and extensibility of this work with two case studies in Section 5, followed by the conclusion.

2. MOTIVATION

2.1 Energy-oriented topology optimization

In an on-chip network using minimal routing, the average energy to transmit one single flit can be expressed in two ways:

$$E_{flit} = H_{avg}(E_R + E_L) \quad (1)$$

$$E_{flit} = H_{avg} \cdot E_R + D_{avg} \cdot E_{L0} \quad (2)$$

where H_{avg} is the average hop count, D_{avg} is the average distance from source to destination, E_R is the average router traversal energy, E_{L0} is the average link traversal energy per unit length and E_L is the average link traversal energy per channel. Among these factors, E_R is determined by router microarchitecture and process technology, and E_{L0} is determined by signaling technique and process technology, H_{avg} and D_{avg} are both strongly influenced by the topology. Since the distance between two nodes is largely determined by their physical locations, little room exists for optimization. Hop count, however, varies greatly among different topologies. In this paper, we present a methodology and study that hinges on hop count to derive the inflection points at which a topology becomes more energy-efficient than another at each technology node.

2.2 Technology scaling

Semiconductor technology progress makes it feasible to build on-chip networks with very high bandwidth. According to data published by the International Technology Roadmap for Semiconductors (ITRS) [1], 13-15 layers of metal wires are expected at 70nm, 50nm and 35nm technologies, which translates to abundant wiring resources for network channels.

Technology scaling also strongly affects network power. A CMOS circuit has three types of power dissipation: dynamic power, static power and short-circuit power. Short-circuit power is negligible compared to the other two power types. Dynamic power has two sources: transistor capacitance and metal wire capacitance. As technology scales, these power sources scale differently. Table 1 lists their scaling factors for current and future technologies expected in five years. Each row represents one source of power consumption. These scaling factors are calculated based on data from ITRS and [8]. All numbers are measured as per unit width/length, and normalized to 0.1μm technology.

Table 1: Normalized power scaling factors.

	0.1μm	70nm	50nm	35nm
transistor capacitance	1	0.78	0.65	0.66
wire capacitance	1	0.94	0.89	0.85
static power	1	2.01	3.35	4.30

The table shows inconsistent scaling factors between dynamic power and static power, and between the two dynamic power sources. In particular, static power increases rapidly as technology scales.

The change in power composition can lead to subtle changes of network power characteristics. We use a common router component: FIFO buffer, as an example to illustrate how the optimal implementations are affected. All quantitative energy/power estimates are given by Orion's power models [17].

Buffers can be implemented as either SRAMs or shift registers. The shift register implementation is usually more expensive in terms of energy because it uses more transistors than the SRAM implementation. But shift register operations (read, write, shift) only involve occupied cells, while SRAM operations (read, write) involve all cells

[1] This work is supported by the Gigascale Systems Research Center funded by MARCO/DARPA.

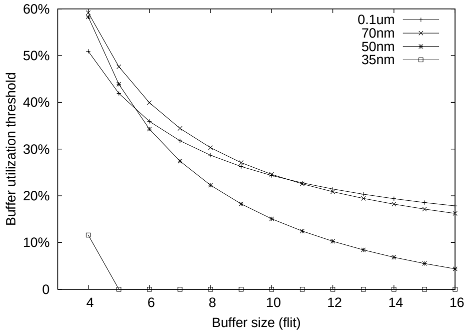

Figure 1: Buffer threshold utilization vs. buffer size.

due to the global bitline and wordline wiring. So shift registers may consume less energy than SRAMs when the buffer utilization is below a certain threshold, and the opposite is true when the buffer utilization is above this threshold.

Figure 1 shows the threshold utilization for different buffer sizes and technologies. The shift register implementation is still viable at 0.1μm technology with relatively smaller buffer size and lower buffer utilization, but is absolutely not an option at 35nm technology. This is mainly caused by the rapidly increasing static power, so at 35nm technology, the advantage of fewer activities is completely overwhelmed by the disadvantage of more transistors.

It is clear from this example that technology scaling must be considered when energy/power is concerned. In our following sections, we study and analyze network topologies in a technology-aware manner.

3. METHODOLOGY DESCRIPTION

Throughout this paper, we derive and compare the energy efficiency of different topologies solely based on network size and technology. In doing this, we need low level energy estimates from Orion's power models, and we make some architecture assumptions, *e.g.*, buffer size, to obtain these energy estimates. For a different set of architecture parameters, the exact quantitative results may differ, but relative trends will remain the same.

3.1 Assumptions

We assume that for an on-chip network, its nodes are placed as a square matrix, regardless of the topology. A network can be viewed as an $N \times N$ matrix, and we mostly use network edge size N as a proxy for network size, rather than the number of nodes N^2.

We also assume uniform random traffic in our discussion so as to reveal the "raw" capability of the topology. Uniform random traffic is the traffic pattern under which every node has equal probability to send data to every other node. Our methodology can be readily extended to support arbitrary traffic patterns, as shown in Section 5.

3.2 Metrics

We use the average flit traversal energy E_{flit} as the network energy efficiency metric. Since flit is the minimal flow control data unit, E_{flit} is the energy cost to transmit unit data. This metric encapsulates the energy efficiency of a network independently of its performance. For instance, a network with lower flit traversal energy and higher throughput can consume higher power than a network with higher flit traversal energy and lower throughput, as the former network can sustain more traffic.

3.3 Workload models

When using the flit traversal energy metric, handling static energy is not as straightforward as handling dynamic energy. Static energy is consumed by routers regardless of flit activities. To capture static energy with E_{flit}, we evenly apportion router static energy per cycle among all flits that pass through the router in that cycle.[2]

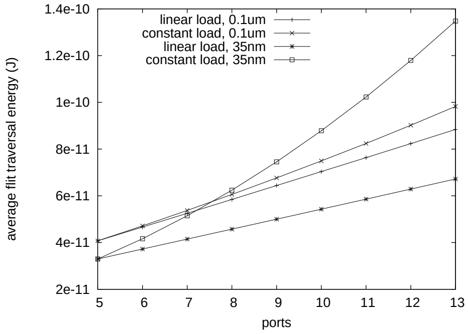

Figure 2: Average router flit traversal energy at 0.1μm and 35nm technologies.

Two workload models are used to approximate the number of flits passing through per cycle:

- Constant load. Assume the number of flits traversing a router in each cycle does not vary with the number of router ports. This assumption models the situation where the workload is determined by application demand rather than the network capacity.
- Linear load. Assume the number of flits passing a router in each cycle is linear in the number of ports. This assumption models the situation where the workload is constrained by the network capacity.

These two models could lead to vastly different flit energy costs. To illustrate with a concrete example, we assume 0.1μm and 35nm technologies, 2GHz frequency, 4-flit buffer size, 128-bit flit size and an input-queuing router. Figure 2 shows E_R given by both models at various port numbers. We choose the load so that the router is fully loaded when it has five ports.

The figure shows that with linear load, E_R is a linear function of the number of ports, where the linear part comes from the crossbar (both dynamic and static) and the constant part comes from the buffer (both dynamic and static). But with constant load, since no more traffic is available to offset the addition of more buffers and the increase of the crossbar size, the buffer static energy becomes a linear term and the crossbar static energy becomes a quadratic term. The quadratically increasing trend is more obvious at 35nm technology.

The two models represent two extremes in workloads.

4. DESIGN SPACE EXPLORATION OF ON-CHIP NETWORK TOPOLOGIES

The 2-D mesh/torus is currently the most popular topology used by on-chip networks, because it perfectly matches the 2-D silicon surface and is also easy to implement. Channels of a 2-D mesh/torus only connect neighboring nodes, so while this topology supports local traffic well, it suffers large hop counts for long-distance traffic. Some more complicated topologies have been used by or proposed for other interconnection networks, such as high-dimensional meshes/tori, hypercubes, hierarchical meshes/tori [7], express cubes [4] and fat-trees [11]. Hyper-cubes are a subset of high-dimensional tori where each dimension has two nodes. To keep the scope manageable, in this paper we consider high-dimensional meshes/tori, hierarchical meshes/tori and express cubes.

One common feature of these topologies is that they use long wires to build express channels (channels that cross multiple hops) so that the hop count for long-distance traffic is effectively reduced. But using express channels inevitably increases router complexity, which increases E_R, and some topologies increase D_{avg} as well, so these topologies may not always reduce E_{flit}, and exploiting them to save energy becomes a complicated trade-off between hop count and router complexity. Intuitively, the larger the network is, the more long-distance traffic can benefit from express channels. So there exists a minimal network edge size N_{min} for each topology beyond which the topology is more energy-efficient than a 2-D mesh/torus.

In the following sections, we first discuss the baseline 2-D mesh/torus, then investigate the other three topologies one by one with a 5-step procedure.

[2] It is only for modeling convenience to assign static energy to active flits.

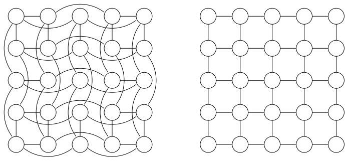

Figure 3: A 5×5 folded torus (left) and a 5×5 mesh (right).

1. Derive the average hop count H_{avg} for the topology based on the expected traffic pattern.
2. Derive the average flit traversal energy E_{flit} as a well-formed analytical function of H_{avg}.
3. Derive the minimal network edge size N_{min}, at which this topology becomes more energy-efficient than the 2-D torus.
4. Derive the optimal (most energy-efficient) configuration of the particular topology.
5. Discuss how N_{min} and the optimal configuration vary with process technologies.

4.1 2-D meshes and tori

Figure 3 shows a mesh and a folded torus. The torus is folded to ensure even link latency.

For an $N \times N$ 2-D mesh, $H_{avg} = \frac{2N}{3}$, so that

$$E^M_{flit} = \frac{2N}{3}(E_R + E_L) = \frac{2N}{3} \cdot E_R + \frac{2N}{3} \cdot E_L$$

For an $N \times N$ 2-D torus, $H_{avg} = \frac{N}{2}$ and it has double E_L due to folding, so that

$$E^T_{flit} = \frac{N}{2}(E_R + 2E_L) = \frac{N}{2} \cdot E_R + N \cdot E_L$$

So a 2-D torus consumes 25% less router energy and 50% more link energy compared to an equal size 2-D mesh, and the ratio between E_R and E_L determines which one is more energy-efficient.

Since all topologies discussed in the rest of this section have mesh variants and torus variants, we only study their torus variants in this paper, and results for the mesh variants can be similarly derived. We use their mesh variants in figures for better legibility.

4.2 High-dimensional meshes/tori

Figure 4 shows how a 9×9 network can be realized as either a 2-D 9×9 mesh (9-ary 2-mesh) or a 3-ary 4-mesh. While the 3-ary 4-mesh reduces the average hop count from 9 to 6, the mapping of a 4-D topology onto a 2-D surface also doubles the average channel length, so the energy property of a high-dimensional mesh/torus is complicated by the trade-off between router energy and link energy.

In general, when an $N \times N$ network of $M = N^2$ nodes is realized by a $\sqrt[n]{M}$-ary n-cube, the average hop count is

$$H_{avg} = \frac{\sqrt[n]{M}}{4} \cdot n$$

and the average channel length relatively increases by

$$l = \frac{1}{n} \cdot \sum_{i=0}^{n-1} \left(\sqrt[n]{M}\right)^{\lfloor \frac{i}{2} \rfloor} = \begin{cases} \frac{2\sqrt{M}-2}{n(\sqrt[n]{M}-1)} & \text{n is even} \\ \frac{\sqrt{M}^{\frac{n-1}{n}}+\sqrt{M}^{\frac{n+1}{n}}-2}{n(\sqrt[n]{M}-1)} & \text{n is odd} \end{cases}$$

Since $\sqrt{M}^{\frac{n-1}{n}} + \sqrt{M}^{\frac{n+1}{n}} > 2\sqrt{M}$ always holds, we loosen the condition a little by letting

$$l \simeq \frac{2\sqrt{M}-2}{n\left(\sqrt[n]{M}-1\right)}$$

in following discussion, and the subsequent results will be a little optimistic.

In a $\sqrt[n]{M}$-ary n-cube, each router has $2n$ interconnection ports and one local inject/eject port. Let $E_R(n)$ denote the average router traversal energy of an n-dimensional torus. According to Section 3.3, $E_R(n)$ is a quadratic function of n

$$E_R(n) = An^2 + Bn + C$$

where A, B and C are positive constants. So we have

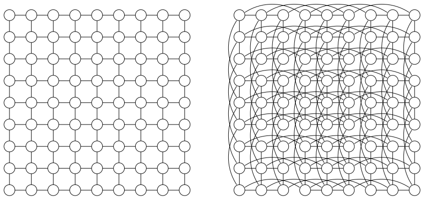

Figure 4: A 9-ary 2-mesh (left) and a 3-ary 4-mesh (right).

$$\begin{aligned} E_{flit}(n) &= H_{avg}(E_R(n) + l \cdot E_L) \\ &= \frac{C}{4}\sqrt[n]{M} \cdot n + \frac{B}{4}\sqrt[n]{M} \cdot n^2 + \frac{A}{4}\sqrt[n]{M} \cdot n^3 \\ &\quad + \frac{\left(\sqrt{M}-1\right)\sqrt[n]{M}}{2\left(\sqrt[n]{M}-1\right)} \cdot E_L \\ &= E_1(n) + E_2(n) + E_3(n) + E_4(n) \end{aligned}$$

Minimal network size

Because

$$\frac{d^2E_1(n)}{dn^2} = \frac{C}{4} \cdot \frac{\sqrt[n]{M} \cdot \ln^2 M}{n^3} > 0$$

$E_1(n)$ is a concave function of n, similarly we can prove that $E_2(n)$, $E_3(n)$ and $E_4(n)$ are all concave functions of n. So $E_{flit}(n)$ is concave, which has the property

$$E_{flit}(3) > E_{flit}(2) \Rightarrow \forall n > 3, E_{flit}(n) > E_{flit}(2)$$

In other words, if the network can benefit from high-dimensional tori, it can benefit from a 3-D torus. So $E_{flit}(3) < E_{flit}(2)$ is the sufficient and necessary condition to determine if pursuing high-dimensional tori is beneficial for network energy.

E_L is determined by the link length, which is in turn determined by the core size. To avoid making assumptions about cores and still get some approximate results, we ignore E_L and only consider $E_R(n)$, and the resultant N_{min} is a lower bound of the minimal network size.

With this simplification, $E_{flit}(3) < E_{flit}(2)$ is equivalent to

$$M > \left(\frac{3}{2} \cdot \frac{E_R(3)}{E_R(2)}\right)^6$$

With $E_R(2)$ and $E_R(3)$ calculated by Orion's power models, Table 2 lists the minimal network size (lower bound) in the form of (M, N_{min}). Here we use 128-bit flit size and 4-flit, 16-flit buffer sizes as representative parameters.

Table 2: Minimal network size for high-dimensional tori, in the form of (M, N_{min}).

	Linear load		Constant load	
buffer size	4-flit	16-flit	4-flit	16-flit
0.1μm	53, 8	44, 7	61, 8	51, 8
70nm	49, 7	42, 7	64, 8	55, 8
50nm	46, 7	39, 7	82, 10	73, 9
35nm	46, 7	36, 6	165, 13	140, 12

Several insights can be extracted. A linearly loaded network requires smaller network size to benefit from high-dimensional tori due to its slower increasing of $E_R(n)$ compared to a constantly loaded network. A network with larger buffer size also requires smaller network size to benefit from high-dimensional tori due to its larger constant term in $E_R(n)$.

The effect of technology scaling is interesting. With linear load, a network at 35nm technology requires smaller network size to benefit from high-dimensional tori than at 0.1μm technology, which can be explained by the smaller slope of $E_R(n)$ at 35nm technology (Figure 2). But with constant load, the opposite is true due to higher energy cost per added port at 35nm technology (Figure 2). So at future technologies, accurate workload prediction is more important for making a correct topology decision.

Optimal dimension

The optimal dimension is the positive root of equation $(E_{flit}(n))'_n = 0$. We cannot find the exact value of the root, but we prove that

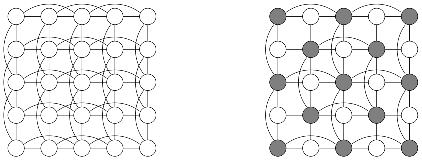

Figure 5: A 5×5 hierarchical mesh (left) and a 5×5 express cube mesh (right), with express interval being 2.

$$3 \leqslant n_{opt} < \ln M$$

So for a network with 1000 nodes, $n_{opt} \in \{3,4,5,6\}$. Note that this is a technology-independent small range convenient for enumeration.

4.3 Hierarchical meshes/tori

In hierarchical meshes/tori, channels not only connect adjacent nodes, but also connect v-node away neighbors in each dimension, as shown in Figure 5. We call the longer channels express channels and call v the express interval.

Routing in an N-node hierarchical ring (one-dimensional hierarchical torus) is equivalent to first routing in a $\frac{N}{v}$-node ring, then routing from the edge nodes of a v-ary 1-mesh to its inner nodes, assuming $v|N$. So the average hop count of an $N \times N$ hierarchical torus is

$$H_{avg} = 2\left(\frac{N}{4v} + \frac{v}{4}\right) = \frac{N}{2v} + \frac{v}{2}$$

Because the minimal distance between any two nodes is achievable in a hierarchical torus, the link energy is not altered by the topology compared to a 2-D torus (this differs from the high-dimensional torus, in which some physically adjacent nodes have no direct connection). So we only need to compare router energy.

The average flit traversal energy is

$$E^H_{flit} = \left(\frac{N}{2v} + \frac{v}{2}\right) E_{R9} + D_{avg} \cdot E_{L0} \quad (3)$$

Minimal network size and minimal express interval

A 2-D torus router has 5 ports, including the inject/eject port, and a hierarchical torus router has 9 ports. Let E_{Rm} denote the average router traversal energy of a router with m ports. In order for a hierarchical torus to save energy, the following inequality must hold

$$E_{R9}\left(\frac{N}{2v} + \frac{v}{2}\right) < E_{R5} \cdot \frac{N}{2} \Rightarrow N(vE_{R5} - E_{R9}) > v^2 E_{R9}$$

which is equivalent to

$$v > \frac{E_{R9}}{E_{R5}} \quad (4)$$

$$N > \frac{v^2 E_{R9}}{vE_{R5} - E_{R9}} \quad (5)$$

Inequality (4) determines the minimal express interval for a hierarchical torus to achieve better energy efficiency than a 2-D torus, and inequality (5) determines the minimal network size for a certain express interval. Table 3 lists the minimal express interval and corresponding minimal network size in the form of (v, N_{min}).

Table 3: Minimal express interval and corresponding minimal network size for hierarchical tori, in the form of (v, N_{min}).

	Linear load		Constant load	
buffer size	4-flit	16-flit	4-flit	16-flit
0.1μm	2, 16	2, 13	2, 20	2, 16
70nm	2, 14	2, 12	2, 22	2, 18
50nm	2, 13	2, 11	2, 46	2, 32
35nm	2, 13	2, 10	3, 28	3, 23

From (5), it can be derived that when v is the integer closest to $\frac{2E_{R9}}{E_{R5}}$,

$$N_{min} > 4\left(\frac{E_{R9}}{E_{R5}}\right)^2 \quad (6)$$

defines the minimal network size, and Table 4 lists its value.

Table 4: Minimal network size and corresponding express interval for hierarchical tori, in the form of (v, N_{min}).

	Linear load		Constant load	
buffer size	4-flit	16-flit	4-flit	16-flit
0.1μm	3, 11	3, 10	3, 12	3, 10
70nm	3, 10	3, 9	3, 12	3, 11
50nm	3, 10	3, 9	4, 14	4, 13
35nm	3, 10	3, 9	4, 21	4, 19

Optimal express interval

For an $N \times N$ hierarchical tori, the optimal express interval is one that leads to minimal average hop count. Let $H'_{avg} = 0$ and simple calculus reveals that

$$v_{opt} = \sqrt{N} \quad (7)$$

and the corresponding theoretical minimal flit traversal energy is

$$E^H_{opt} = \sqrt{N} \cdot E_{R9} + D_{avg} \cdot E_{L0}$$

This theoretical minimal value may not be achievable since $\sqrt{N}$ may not be an integer.

(4) is satisfied when (6) and (7) are true, so $\sqrt{N}$ is indeed the optimal express interval. Note that the optimal express interval is technology-independent.

Table 3 and 4 show similar trends to those for high-dimensional tori. Linear load, larger buffer size and technology progress all make networks more likely to benefit from topology improvements, but technology progress also makes networks more sensitive to workload.

4.4 Express cubes

The express cube [4] is another hierarchical topology, which provides express channels for every other v nodes (Figure 5). Although not all nodes are connected with express channels, long-distance traffic can still use express channels to bypass intermediate nodes, but the network does not pay router complexity penalty at every node.

Due to the similarity between the express cube and the hierarchical torus, we only present the results without detailed derivation.

Equations for average flit traversal energy, minimal network size and theoretical minimal flit traversal energy are

$$E^E_{flit} = v \cdot E_{R5} + \frac{N}{2v} \cdot E_{R9} + D_{avg} \cdot E_{L0} \quad (8)$$

$$N_{min} > 8 \cdot \frac{E_{R9}}{E_{R5}} \quad (9)$$

$$E^E_{opt} = \sqrt{2N \cdot E_{R5} \cdot E_{R9}} + D_{avg} \cdot E_{L0} \quad (10)$$

Table 5 lists the minimal express interval and corresponding minimal network size for several technologies, and Table 6 lists the minimal network size and corresponding express interval.

Table 5: Minimal express interval and corresponding minimal network size for express cubes, in the form of (v, N_{min}).

	Linear load		Constant load	
buffer size	4-flit	16-flit	4-flit	16-flit
0.1μm	2, 20	2, 17	2, 24	2, 20
70nm	2, 18	2, 16	2, 26	2, 22
50nm	2, 17	2, 15	2, 50	2, 36
35nm	2, 17	2, 14	3, 25	3, 22

Table 6: Minimal network size and corresponding express interval for express cubes, in the form of (v, N_{min}).

	Linear load		Constant load	
buffer size	4-flit	16-flit	4-flit	16-flit
0.1μm	3, 13	3, 13	3, 14	3, 13
70nm	3, 13	3, 12	3, 14	3, 14
50nm	3, 13	3, 12	4, 15	4, 15
35nm	3, 13	3, 12	4, 19	4, 18

4.5 A comparison of high-dimensional tori, hierarchical tori and express cubes

4.5.1 *Hierarchical tori vs. express cubes*

We first compare the case where the two topologies have the same express interval v. From (3) and (9) we have

$$E^H_{flit} - E^E_{flit} = v\left(\frac{E_{R9}}{2} - E_{R5}\right)$$

So when $\frac{E_{R9}}{E_{R5}} > 2$, the hierarchical torus has higher flit traversal energy than the express cube, and the opposite is true when $\frac{E_{R9}}{E_{R5}} < 2$.

We then compare their theoretical minimal flit traversal energy. Since they have identical link energy, we only need to compare the router energy part.

$$\frac{E^H_{opt}}{E^E_{opt}} = \sqrt{\frac{E_{R9}}{2E_{R5}}}$$

And again, the value of $\frac{E_{R9}}{E_{R5}}$ determines which topology is better.

For hierarchical tori and express cubes, $\frac{E_{R9}}{E_{R5}}$ is also the minimal express interval. From Table 3 and 5, the minimal express interval switches between 2 and 3 at 35nm technology for different load models, so which topology is better depends on which load model is closer to reality.

4.5.2 *Hierarchical tori vs. high-dimensional tori*

In this section we will prove that hierarchical tori practically always have smaller flit traversal energy than high-dimensional tori. Since high-dimensional tori have longer average link length than hierarchical tori, we can back our claim by proving that high-dimensional tori always have higher or equal router energy.

Ignoring link energy, we have

$$E^H_{opt} = \sqrt{N} \cdot E_{R9}$$

We make a practical assumption that the network has fewer than 1000 nodes, so for a high-dimensional torus, the optimal dimension $n_{opt} \in \{3, 4, 5, 6\}$. For an $N \times N$ network, the flit traversal energy of a 3-D, 4-D, 5-D and 6-D torus is

$$\begin{aligned}
E^{3D}_{flit} &= \frac{\sqrt[3]{N^2}}{4} \cdot 3 \cdot E_R(3) \\
E^{4D}_{flit} &= \frac{\sqrt[4]{N^2}}{4} \cdot 4 \cdot E_R(4) = \sqrt{N} \cdot E_R(4) \\
E^{5D}_{flit} &= \frac{\sqrt[5]{N^2}}{4} \cdot 5 \cdot E_R(5) \\
E^{6D}_{flit} &= \frac{\sqrt[6]{N^2}}{4} \cdot 6 \cdot E_R(6)
\end{aligned}$$

Since a router of an n-dimensional torus has $2n+1$ ports, $E_R(4) = E_{R9}$, and $E^{4D}_{flit} = E^H_{opt}$. We conclude that hierarchical tori are better than 4-D tori.

For 5-D tori, if we assume $E^{5D}_{flit} < E^H_{opt}$, which implies $N > \left(\frac{5}{4} \cdot \frac{E_{R11}}{E_{R9}}\right)^{10}$, then by substituting energy estimates from Orion, we have $N \geqslant 45$. This conflicts with our assumption that the network has less than 1000 nodes. Similar reasoning can be applied to 6-D tori, and we conclude that hierarchical tori are better than 5-D and 6-D tori.

For 3-D tori, we can derive that $E^{3D}_{flit} < E^H_{opt}$ holds when $N < 19$. But this requires the network to be organized as a $\sqrt[3]{N^2} \times \sqrt[3]{N^2} \times \sqrt[3]{N^2}$ matrix, which is not square when mapped to a 2-D surface. Since chips are usually square, with this constraint, the network needs to be organized as an $N \times \sqrt{N} \times \sqrt{N}$ matrix, therefore

$$E^{3D}_{flit} = \left(\frac{N}{4} + \frac{\sqrt{N}}{4} + \frac{\sqrt{N}}{4}\right) \cdot E_R(3)$$

and the condition to satisfy $E^{3D}_{flit} < E^H_{opt}$ becomes

$$N < \left(\frac{4E_{R9}}{E_{R7}} - 2\right)^2$$

By substituting energy estimates from Orion, we have $N < 9$, which is smaller than the minimal network size for hierarchical tori to save energy. This implies that when a hierarchical torus is able to save energy, it saves more than a 3-D torus.

So we prove that when hierarchical tori can save energy over 2-D tori, they always have lower flit traversal energy than high-dimensional tori.

4.5.3 *Express cubes vs. high-dimensional tori*

By using the same method as used in the previous section, we conclude that when express cubes can save energy over 2-D tori, they also have smaller flit traversal energy than high-dimensional tori. In short, from an energy standpoint, high-dimensional tori should never be selected over hierarchical or express cubes.

5. CASE STUDIES

5.1 Topology scaling

To demonstrate how this work can help predict the right network topology for future technologies, we construct a hypothetical case study based on a real design.

The TRIPS [13] operand network is a 5×5 mesh-like network connecting ALUs and memory components. To better match our presented results, we assume torus topology in this case study. TRIPS is designed at 0.1μm technology. Suppose we want to scale this design to 70nm, 50nm and 35nm technologies. According to the ITRS, the transistor count doubles for each technology transition listed above, while the chip size stays constant. To facilitate reuse of the cores as technology scales, we will build larger networks with the same cores, so the network size will be 7×7 at 70nm technology, 10×10 at 50nm technology and 14×14 at 35nm technology. We assume 4-flit buffer size, 128-bit flit size and $E_L = 0.53E_{R5}$, as they are in the current design. We also assume uniform random traffic and linear load.

Table 7 lists the most energy-efficient topologies predicted by our analytical approach. H-v stands for "hierarchical torus with express interval v", and E-v stands for "express cube with express interval v".

Table 7: Predicted optimal topologies.

technology	70nm	50nm	35nm
optimal topology	2-D torus	H-3 torus	H-4 torus

To validate the accuracy of our predictions, we use Orion to estimate the average flit traversal energy of these topologies at 70nm, 50nm and 35nm technologies, and the results are listed in Table 8. The top half of the table lists the results for high-dimensional tori, and the bottom half lists the results for hierarchical tori and express cubes. It can be seen that the simulation results match our predictions.

Table 8: Simulated average flit traversal energy (10^{-10}J) of tori, hierarchical tori and express cubes, with the minimal energy at each technology highlighted.

torus dimension	2	3	4	5	6
70nm	**1.59**	1.79	1.98	2.44	N/A
50nm	1.98	2.14	2.21	2.67	2.83
35nm	3.53	3.88	3.40	4.03	4.27

topology	H-2	H-3	H-4	E-2	E-3	E-4
70nm	1.82	1.78	1.88	1.96	1.98	2.15
50nm	2.07	**1.94**	1.97	2.19	2.12	2.22
35nm	3.47	3.14	**3.01**	3.63	3.38	3.42

This example demonstrates how our methodology can be applied to predict the optimal network topology. It also highlights that when current network design is scaled to future technologies, simply preserving the current topology will not optimize energy efficiency. The most energy-efficient topology should be chosen with regard to network size and the technology node.

5.2 Arbitrary traffic patterns

To handle arbitrary traffic patterns other than uniform random traffic, we only need to change the computation of the average hop count H_{avg} based on the specific traffic pattern, then apply this new H_{avg} to the 5-step procedure.

If the traffic pattern can be expressed by a probability matrix or some other mathematical form, then H_{avg} can be easily computed.

For example, assume a traffic pattern which is the aggregation of an $N \times N$ global uniform random traffic and a $v \times v$ local uniform random traffic, with probability p and q respectively, then for a 2-D torus

$$H_{avg}^{T} = p \cdot \frac{N}{2} + q \cdot \frac{v}{2}$$

and the average hop count of other topologies can be similarly derived.

For some applications, such well-formed mathematical expressions are hard to obtain, and we have to resort to network simulation to derive hop count information.

Consider the 10×10 network at 50nm technology and three SPLASH benchmarks: *LU*, *mp3* and *water* [14]. We choose *LU* and *water* because they can be scaled to a 10×10 network. Mp3 does not have that scalability, and we choose it for comparison purpose. We use RSIM [12] to simulate the benchmarks and collect traffic traces to compute the average hop count. Then we apply the hop count values numerically to a variety of topologies to find the optimal topology. Table 9 lists (H_{avg}, E_{flit}) values under these traffic patterns. We do not include high-dimensional tori due to limitations of RSIM, but it suffices for demonstration.

Table 9: Average hop count and flit traversal energy under SPLASH benchmark traces.

benchmarks	*LU*	*water*	*mp3*
2-D torus	5.07, 201pJ	4.93, 196pJ	1.55, **61pJ**
H-2	3.05, **190pJ**	2.97, **185pJ**	1.55, 82pJ
H-3	3.07, 191pJ	3.00, 186pJ	1.55, 82pJ
H-4	3.26, 198pJ	3.20, 194pJ	1.55, 82pJ
E-2	4.26, 212pJ	4.09, 207pJ	1.55, 65pJ
E-3	4.81, 213pJ	4.58, 207pJ	1.55, 65pJ
E-4	4.87, 211pJ	4.67, 205pJ	1.55, 65pJ

H-3 is the optimal topology under random traffic at 50nm technology. Now H-2 becomes the optimal topology for both *LU* and *water* since these two benchmarks scale well to the large network size and can take advantage of express channels. But no energy reduction can be achieved for *mp3* benchmark, since *mp3* benchmark only uses a 3×3 subset of the network, hence express channels cannot be utilized. So for real applications to benefit from these energy-efficient topologies, good scalability is required.

6. RELATED WORK

The concept of on-chip interconnection networks was introduced fairly recently [3, 5, 6]. While network topologies have been extensively studied, with many topologies proposed for either performance or hardware efficiency [4, 7, 11], optimizing topologies for power/energy is more recent. In 1999, Zhang *et al.* exploited generalized meshes to reduce the energy dissipation of DSP interconnects [19]. In 2003, Wang *et al.* proposed and explored power-efficient network architectures, and investigated the energy saving capability of express cubes through simulations [16]. In 2004, Jalabert *et al.* developed ×pipesCompiler, a tool for instantiating application-specific on-chip networks with customized topologies to reduce both power and area [10].

Our work differs from these prior works in several key aspects. First, our goal is to derive a criteria for prescribing the most energy-efficient topology at a particular process technology. Second, we use a single analytical model to study a wide range of topologies, and our methodology can be easily applied to other topologies as well. Third, we do not rely on simulation tools and all results in this paper are mathematically provable with the support of low level power models. We consider topology customization an orthogonal direction of our work, so pursuing that direction based on the topology suggested by our method can lead to more energy savings.

7. CONCLUSIONS

In this paper, we perform a technology-aware and energy-oriented topology exploration for on-chip networks. We use 2-D meshes/tori as the baseline, and consider high-dimensional meshes/tori, hierarchical meshes/tori and express cubes as candidate energy-saving topologies. For each of them, we derive the minimal network size and other conditions under which that topology can save energy compared to the baseline. We also derive the criteria to compare these topologies and use a case study to demonstrate how our method can help choose the most energy-efficient topology.

Our method only requires information of network size, technology and a limited set of architecture parameters. Network simulation is usually not required, so our method can help make early-stage decisions that will guide later tasks such as placement and routing. With the advent of on-chip networking, a roadmap that guides on-chip network designers just as ITRS guides circuit designers is critically needed. We are currently extending this work to derive a complete roadmap for on-chip networks, which will support more traffic distributions, topologies and other aspects of network design.

8. REFERENCES

[1] International technology roadmap for semiconductors. `http://public.itrs.net`.

[2] L. Benini and G. D. Micheli. Powering networks on chips. In *Proc. International System Synthesis Symposium*, pages 33–38, 2001.

[3] L. Benini and G. D. Micheli. Networks on chips: A new SoC paradigm. *IEEE Computer*, 35(1):70–78, 2002.

[4] W. J. Dally. Express cubes: Improving the performance of k-ary n-cube interconnection networks. *IEEE Transactions on Computers*, 40(9):1016–1023, 1991.

[5] W. J. Dally and B. Towles. Route packets, not wires: On-chip interconnection networks. In *Proc. Design Automation Conference*, pages 684–689, 2001.

[6] P. Guerrier and A. Greiner. A generic architecture for on-chip packet-switched interconnections. In *Proc. Conference on Design Automation and Test in Europe*, 2000.

[7] S. Hauck, G. Borriello, and C. Ebeling. Mesh routing topologies for multi-FPGA systems. *IEEE Transactions on VLSI Systems*, 6(3):400–408, 1998.

[8] R. Ho, K. W. Mai, and M. A. Horowitz. The future of wires. *Proceedings of the IEEE*, 89(4):490–504, 2001.

[9] J. Hu and R. Marculescu. Energy-aware mapping for tile-based NoC architectures under performance constraints. In *Proc. Asia-South Pacific Design Automation Conference*, 2003.

[10] A. Jalabert, S. Murali, L. Benini, and G. D. Micheli. ×pipescompiler: A tool for instantiating application specific networks on chip. In *Proc. Conference on Design Automation and Test in Europe*, 2004.

[11] C. E. Leiserson. Fat-trees: Universal networks for hardware-efficient supercomputing. *IEEE Transactions on Computers*, 34(10):892–901, 1985.

[12] V. S. Pai, P. Ranganathan, and S. V. Adve. RSIM: An execution-driven simulator for ILP-based shared-memory multiprocessors and uniprocessors. In *Proc. International Symposium on High Performance Computer Architecture*, pages 72–83, 1997.

[13] K. Sankaralingam, R. Nagarajan, H. Liu, C. Kim, J. Huh, D. Burger, S. W. Keckler, and C. R. Moore. Exploiting ILP, TLP, and DLP with the polymorphous TRIPS architecture. In *Proc. International Symposium on Computer Architecture*, pages 422–433, 2003.

[14] J. P. Singh, W.-D. Weber, and A. Gupta. SPLASH: Stanford parallel applications for shared-memory. *Computer Architecture News*, 20(1):5–44, 1992.

[15] M. B. Taylor, W. Lee, S. Amarasinghe, and A. Agarwal. Scalar operand networks: On-chip interconnect for ILP in partitioned architectures. In *Proc. International Symposium on High Performance Computer Architecture*, pages 341–353, 2003.

[16] H. Wang, L.-S. Peh, and S. Malik. Power-driven design of router microarchitectures in on-chip networks. In *Proc. International Symposium on Microarchitecture*, 2003.

[17] H.-S. Wang, X. Zhu, L.-S. Peh, and S. Malik. Orion: A power-performance simulator for interconnection networks. In *Proc. International Symposium on Microarchitecture*, pages 294–305, 2002.

[18] T. T. Ye, L. Benini, and G. D. Micheli. Analysis of power consumption on switch fabrics in network routers. In *Proc. Design Automation Conference*, 2002.

[19] H. Zhang, M. Wan, V. George, and J. Rabaey. Interconnect architecture exploration for low-energy reconfigurable single-chip DSPs. In *Proc. IEEE Computer Society Workshop on VLSI*, 1999.