Validation of Theoretical Power Analysis with a Network Power Performance Simulator

Emily Finn

Abstract:

Power analysis is becoming more and more important in interconnection networks. It is no longer sufficient to only look at performance when designing a network. It is important to look at both power and performance. It is especially important to observe the relationship between the two. This paper intends to validate a theory connecting the number of links used to the power consumed. It also discusses the trade-off between power and performance.

Introduction:

As interconnection networks become more prominent in systems, it is becoming important to analyze the power consumption in these networks. Minimizing power consumption is important because some systems have a limit on peak power. It is important to keep power analysis in mind when designing routing algorithms. Therefore, it would be useful to have a power model based on the number of links used by the network. Since link power is a significant part of total power, decreasing the number of links used would decrease the power. Ideally, we want to come up with a routing algorithm that will minimize peak power for any traffic permutation.

Background:

The power consumed by a network is the sum of the power consumed in each part of the network: link power, arbiter power, crossbar power, buffer read power, buffer write power, etc. Decreasing one of them, for example link power, decreases the total power consumed. A link only consumes power when a message is traversing it (load is greater than 0). Therefore, decreasing the number of active links decreases the total power consumed by the network. To get the minimal number of links, I used the minimal connectivity of k-ary 2-cube graphs, or the Hamiltonian cycle. My goal was to validate this theory using a network simulator called Orion. To do this I used a series of specific permutations and heuristic routing algorithms created by Noel Eisley.

Background on Orion and Popnet:

Orion is a network simulator. It has parameters that can be changed based on specific types of networks. The set of parameters that were used for these tests were based on the IMB Infiniband 8x8 12x router. I used Orion with another simulator called Popnet. Popnet inputs certain parameters about the simulated network and then uses Orion to calculate power consumption for that network. The parameters necessary to input into Popnet are:

ary size	k in k-ary n-cube
cube dimension	n in k-ary n-cube
virtual channel number	number of virtual channels in each physical channel
buffer size	size of buffer in flits
Outbuffer size	size of outbuffer in flits
flit size	flit size in bits
link length	length of link in micrometers
simulation length	time of simulation in cycles
trace file	list of messages being sent
routing algorithm	which built in algorithm to use

Figure 1: parameters for Popnet

Name	**Permutation**	**routing algorithm**
perm1a	each node sends to node below it	TXY
perm1b	each node sends to node to the left	TXY
perm2	2 hops in xdir and 2 hops in ydir	TXY
5x5_fewest	See Appendix 1	uses 40 links
5x5_most	See Appendix 1	uses 53 links
8x8_001a	See Appendix 2	uses 174 links
8x8_001b	See Appendix 2	uses 127 links
8x8_001c	See Appendix 2	uses 126 links
snake2	2 hops in xdir and 2 hops in ydir	Hamiltonian cycle
snake_001	See Appendix 2	Hamiltonian cycle

Figure 2: permutations and routing algorithms

The trace file has one line for each message, and each line contains the following information: time when message is sent, source, destination, and packet size. In using Orion and Popnet, I added routing algorithms to the code and ran the simulator with many different trace files.

Experiments:

I created a couple of different permutations and ran them with different algorithms. Each algorithm used a different number of links. Then, I compared the results to validate a connection between the number of links used and the power consumed. Figure 2 shows a table of permutations and algorithms used. Figure 3 shows the parameters used for the test runs.

ary size	8
cube dimension	2
virtual channel number	2
buffer size	16
outbuffer size	16
flit size	12
link length	3000
simulation length	1.00E+06
injection rate	0.005

Figure 3: Parameters used for tests.

Figure 4 shows the relationship between link power and total power for each of the permutations. Although the link power is less than fifty percent of the total power, it is significant enough that reducing the link power will reduce the total power.

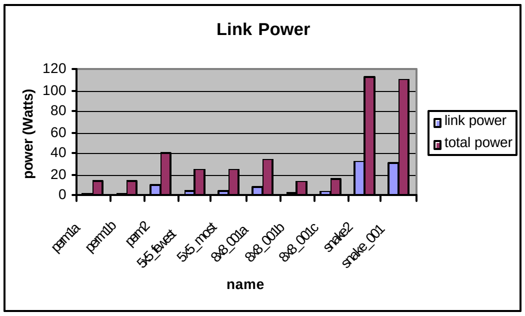

Figure 4: The relationship between link power and total power.

Figure 5 presents the peak power for each test run and compares that to using the TXY algorithm. Figure 5 shows that reducing the number of active links does in fact reduce the peak power. The case of the 8x8_001 algorithms seems to be the exception. These extraneous results occurred because the 8x8_001a, 8x8_001b, and 8x8_001c test runs experienced deadlock. The link power was 3.23822 for the 8x8_001b and 3.95248 for the 8x8_001c. Since the link power was higher for the 8x8_001c, the state of deadlock in the 8x8_001c most likely had more active links than the state of deadlock in the 8x8_001b. That would explain why the 8x8_001c consumed more power.

Name	Peak Power	Peak Power with TXY
perm1a	139.852	139.852
perm1b	139.91	139.91
perm2	407.983	407.983
5x5_fewest	145.897	204.447
5x5_most	233.841	204.447
8x8_001a	350.222	395.695
8x8_001b	180.089	395.695
8x8_001c	213.498	395.695
snake2	135.299	407.983
snake_001	162.436	395.695

Figure 5: Peak power for each test run compared to peak power with TXY algorithm.

Figure 6 shows the results of comparing two algorithms for a 5x5 permutation. The 5x5_fewest algorithm uses 40 links, and the 5x5_most uses 53 links. In the beginning, the algorithm that uses more links consumes more power. After about 300,000 cycles, all of the messages have been received so the power levels out. At this point, the two simulations consume the same amount of power, because the only power still being consumed is leakage power.

The snake algorithm, the Hamiltonian cycle, is a form of non-minimal routing. It uses the fewest possible links, because it goes through each node only once. Since it uses the fewest links, we predict that it consumes the smallest amount of power. Figure 7 shows that the results were consistent with this prediction for perm2.

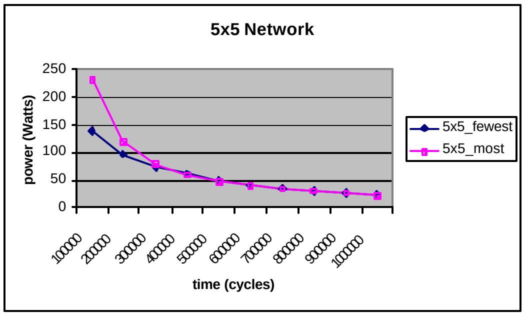

Figure 6: Comparison of two algorithms for a permutation on a 5x5 network.

In Figure 7, the power for perm2 stays consistent for the first 120,000 cycles and then dips down. This happens because the network is very congested while the messages are being sent out. Once all of the messages have been sent, the network begins to clear up, and the power begins to drop. This does not happen for the snake algorithm because the network stays congested after all of the messages have been sent. In fact, the network becomes slightly more congested after this turning point. As can be seen from these results, using fewer links is actually a trade-off between total power and average delay. Figure 8 shows the average delay for each permutation and algorithm.

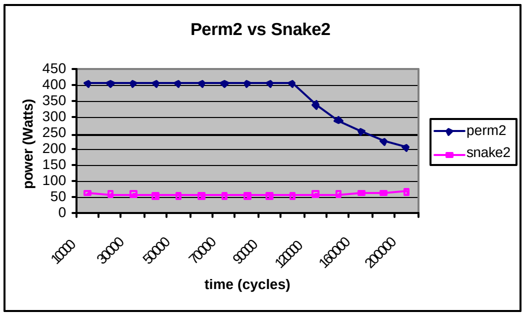

Figure 7: Comparison of two different algorithms for the permutation in perm2.

Name	Average Delay	Average Delay with TXY
perm1a	10.6158	10.6158
perm1b	10.6158	10.6158
perm2	26.1304	26.1304
5x5_fewest	55225.7	12121.5
5x5_most	1832.64	12121.5
8x8_001a	5748.71	2030.88
8x8_001b	25.2904	2030.88
8x8_001c	27.2537	2030.88
snake2	999771	26.1304
snake_001	910077	2030.88

Figure 8: Average delay for each test run.

As can be seen from Figure 8, using fewer links drastically increases the delay. The delay for the 8x8_001 algorithms is not comparable with what was expected because these test runs experienced deadlock. For these routing algorithms, there is no clear method for avoiding deadlock, so we could not get the desired results.

Although the peak power is much lower in the cases that used fewer links, the energy is actually larger because it takes much more time (E = P/t). However, this method of decreasing power is good for systems that have a limit on peak power.

Conclusions and Future Work:

The results validate the theory that using fewer links decreases the peak power. However, there are other factors that should be considered. Using fewer links also increases the average delay. In some cases the delay is so large that it may not be worth losing performance to decrease power.

Although my part of the project is finished, Noel Eisley will continue to work on validating this theory. He has more test runs using different numbers of active links. A direct correlation between number of active links and peak power is yet to be determined. He is also working on the problem of finding the algorithm that uses the minimum number of links for any permutation. He is trying to prove that this problem is NP-Complete.

References:

[1] N. Eisley. Theoretical Power Analysis of k-ary 2-cube Networks under Permutation Traffic.
[2] H. Wang, X. Zhu, L. Peh, and S. Malik. Orion: A Power Performance Simulator for Interconnection Networks, In Proceedings of the 35th International Symposium on Microarchitecture (MICRO), Istanbul, Turkey, November 2002.

Appendix 1:
Permutation for 5x5_fewest and 5x5_most

0→17
1→7
2→21
3→4
4→2
5→18
6→20
7→22
8→13
9→14
10→10
11→8
12→5
13→6
14→1
15→16
16→0
17→15
18→24
19→19
20→3
21→11
22→23
23→12
24→9

Appendix 2:
Permutation for 8x8_001a, 8x8_001b, 8x8_001c, and snake_001

0→30
1→41
2→45
3→28
4→31
5→34
6→14
7→26
8→29
9→42
10→13
11→63
12→59
13→36
14→25
15→18
16→8
17→55
18→35
19→58
20→38
21→49
22→12
23→47
24→60
25→16
26→10
27→56
28→62
29→40
30→6
31→52
32→39
33→20
34→19
35→44
36→48
37→54
38→24
39→0
40→46
41→4
42→43
43→23
44→3
45→27
46→7
47→53
48→2
49→22
50→50
51→5
52→51
53→15
54→37
55→33
56→32
57→57
58→1
59→21
60→9
61→11
62→61
63→17