ECE 538 Spring 2014

A Comparative Study of Interconnection Network

By Alejandro Flores

Advanced Computer Architecture

*Contents*

[*1.0 Abstract 3*](#_Toc405319115)

[*2.0 Introduction 3*](#_Toc405319116)

[*2.1 Background 4*](#_Toc405319117)

[*3.0 Interconnections Networks 5*](#_Toc405319118)

[*3.1 Buses 5*](#_Toc405319119)

[*3.2 Direct and Indirect Networks 7*](#_Toc405319120)

[*3.3 Butterfly Network 8*](#_Toc405319121)

[*3.4 Torus 9*](#_Toc405319122)

[*4.0 Comparison 10*](#_Toc405319123)

[*4.1 Throughput 10*](#_Toc405319124)

[*4.2 Latency 11*](#_Toc405319125)

[*4.3 Path Diversity 12*](#_Toc405319126)

[*4.4 Performance 14*](#_Toc405319127)

[*5.0 My Observations 16*](#_Toc405319128)

[*6.0 Conclusion 17*](#_Toc405319129)

[*7.0 Lessons Learned 17*](#_Toc405319130)

[*8.0 Works Cited 19*](#_Toc405319131)

[*9.0 Appendix A 20*](#_Toc405319132)

# *1.0 Abstract*

In recent years, processor speeds have hit a steady state that is limiting the growth of computing power in single processor systems. This is forcing other system components to evolve and become more efficient. A good example of this is interconnection networks. These are networks that are used to interconnect processors to memory and or I/O devices within a system. In the past, these systems where mainly connected via a crossbar interconnection network that are inefficient and slow.   
  
In this paper, we will go over to of the butterfly and tour family of interconnection networks. The paper will start by going over the background of how interconnection networks came about. It will then go over the details of each topology and analyze their performance in terms of throughput, latency and path diversity. Finally, a comparison of the two will be made and suggestions of what application might be best for each will be noted.

# *2.0 Introduction*

Communication networks have become key component to the many technologies we have come to rely on. Today, these networks expand into the area of single processor system communication and are often referred to as interconnection networks. In recent years, the topic of interconnection networks has gained some track due to communication issues that arise as designers pack more and more cores into a single machine.

To further understand the importance of interconnection networks, we must first study the evolutions of such networks and attempt to identify what has lead us to the current state.

## *2.1 Background*

Without a doubt, in the last 50 years, significant improvements have been made in the areas of computer systems and computer networks. If we read literature related to the early stages of computer systems, we find that the concept of interconnecting computer system quickly became apparent to computer designers. The idea of making use of a centralized “computer center” for processing work was soon replaced by interconnecting multiple independent computers [1]. Soon after, the concept of creating a network of computers became crucial to many business applications that required sharing of resources [1].

As the number of computers increased, due in part to the development of the microprocessor, the interconnections of these computer systems lead to the discovery of spectacular technologies such as the Internet [1]. By the mid 1980’s, computer performance was increasing at a speed of 52% per year and then by 2003, it began slowing down to 22% per year [2]. This reduction in performance was mainly attributed to reaching the maximum power dissipation of air cooled chips and lack of instructions that could take advantage of instruction-level parallelism in single processor systems [2]. As a result, companies such as Intel began concentrating their efforts on increasing performance via multi-processor chips, often referred to as multicore processors [2]. With the change to multicore processors, the issues of providing an efficient communication mechanism began to take root. Today, computers systems are built using several multi-core processors, putting the topic of interconnection networks at the forefront.

In this paper, I will analyze the most popular forms of interconnection networks and highlight the difference between them. Additionally, I will go over some of the applications that might benefit from the functionality of each network. Finally, I will provide a brief summery of my observations and close with the lessons learned in conducting this research.

# *3.0 Interconnections Networks*

Computer systems equipped with multicore processors need an efficient way of interconnecting the multicore processor to memory and I/O devices to I/O controllers. To be specific, these digital systems require an interconnection network that operate at very high data rates within very short paths. These characteristics make interconnection networks different from the networks we have come to learn about in computer networks [3].

## *3.1 Buses*

Buses are one of the simplest and perhaps the most used interconnection network in early implementation of the single processor computer systems [3]. As illustrated in Figure 1, a bus is a simple means of interconnecting multiple components via a single channel.

BUS

Component   
A

Component   
B

Component   
C

*Figure 1: Bus*

This type of interconnection network poses the power of easily broadcasting a message (logical unit of data exchange) in an inexpensive serialized manner. Because the buses can only process one message at a time, it is important that the traffic in the bus be synchronized (3). As expected, this limits the number operations that could be processed per clock cycle. In a more advance implementation of a bus network, designers make use of multiple parallel busses. These parallel busses can separate the traffic into control, address, and data bits. Figure 2 illustrates the basic structure of a parallel bus interconnecting a processor to two memory modules.

*Control*

P1

M1

M2

*Address*

*Data*

Figure 2: Parallel Bus

While this might work great for basic single processor systems, today’s high-speed computers running at approximately 10^9 instructions per second require a more efficient ways of handling multi-message operations [3]. Furthermore, as stated in the Interconnection Networks textbook, performance for interconnection networks are being forced to improve as other components have already reached their limit. Additionally, the bus model does not scale well when we increase the number memory banks. For this reason, buses have recently been replaced with point-to-point interconnection networks that are faster and offer concurrency [3].

## *3.2 Direct and Indirect Networks*

After looking into the most basic version of an interconnection networks, we must recognize that the bus topology will not yield significant improvements in speed.

Alternatively, we must look at other options that take into consideration the design constraints that lead to the most appropriate network design.   
The two major categories for network topology are the direct (or point-to-point) and indirect networks. In a direct network, each node or device is both a terminal and a switch while in indirect networks; the node can be a terminal or switch. Figure 3 should help better understand the concept between direct and indirect networks.



Figure 3: Butterfly and Torus Networks

There are many variations for each of these networks but the butterfly (indirect) and torus (direct) have become somewhat popular because of their unique characteristics. Moreover, the authors of the Interconnection Networks textbook appear to agree.

## *3.3 Butterfly Network*

What makes the butterfly network unique is its simplicity in routing and the fact that the hop count stays the same regardless of the destination location (Hops= logkN+1). To fully understand these two concepts, let’s go over an example.   
  
Figure 3a has a red line marked to illustrate how a message would get routed from node 3 (011) to node 6(110). To get to node 6, we simply take the route with the 110 patterns. To determine the amount of hops we need to obtain N = kn where k=2(ary) and n=3(fly) to get N=8. Plugging these values in the hops equations, we get: *Hops= log 28+1 = 4.* This means that it will take us no more than 4 hops to get from the source (left circle) to the destination (right circle) in all cases regardless of the source and destination we choose.

The butterfly network does however have a few disadvantages that are worth mentioning. To start with, the butterfly network suffers from low path diversity; meaning there are few route options to get to a destination. This lack of routes could lead to systems degradation if a node fails [4]. This network topology also requires long wires that limit its capability to perform well in large interconnection networks because the speed decreases as the distance increases. A detailed picture of what a butterfly network is available in Appendix A.

## *3.4 Torus*

Another network topology that is often used for interconnection networks is the torus network. This network structure comes in two options with one simply referred to as torus while the other is called mesh torus. To better identify the difference between the two, an illustration of these two networks is available in Appendix A.

The torus topology offers really good diversity; leading to good load balancing even with permutation traffic [3]. To better explain this notion, Figure 4 illustrates how using this topology will provide multiple paths for getting from one node to another.



*Figure 4: Torus Network Path Diversity*

Another advantage of using a torus networks is that all routes are bidirectional and therefore can take advantage of bidirectional signaling. More importantly, torus networks have uniform short wires that make it ideal for high-speed operations [3].

As with any other network topology, torus also has its disadvantages. Perhaps the most significant issue arises when a considerable amount of nodes are used. Having too many nodes could potentially create temporary memory. In addition, a high latency is experienced with this network topology.

# *4.0 Comparison*

After going over both of the leading topologies for interconnection networks, we can now compare the performance to each other.  
To effectively make this comparison, we need to define the parameters to evaluate each technology on. In this paper, we will mainly focus on three major parameters:

1. Throughput – the data rate per second accepted by input port
2. Latency – the time between the data being send and being received
3. Path Diversity – number of path that lead to the same node

While these are the parameters I selected for this paper, there are others that might also impact the performance of the networks such as delay, power consumption, error detection, and message size among others.

## *4.1 Throughput*

When selecting an interconnection network, it is imperative that we select a network design that supports our throughput needs. This parameter becomes particularly important when working with high-speed networks.

To be clear, we are not talking about bandwidth in this section. While bandwidth specifies the width of our median, the throughput is the actual rate of transfer (speed) within the media.

The ideal throughput of a butterfly and tour networks with uniform load can be calculated as follows:

Torus

Butterfly

(1.0)

Where Bn is the maximum bandwidth and Bs is the maximum bandwidth across the bisection of the system.

The question now is: what butterfly or torus configuration will yield the maximum bandwidth and minimize latency. Well it turn out there is a formula that can help determine the best values for n and k.

(1.1)

k=

Butterfly

nk

Torus

Obtaining the best values to choose for our n and k will result in a better throughput performance in our network.

## *4.2 Latency*

Regardless of the network we decide to use for a specific problem, latency is also an important aspect to analyze. A network with high latency levels can cause the system to become unstable and or simply crash. For this reason, we need to make sure a system is designed to tolerate a predefined latency value.   
  
When using the butterfly configuration, we need to consider the serialization Ts, hop Th, and wire Tw latencies. The serialization latency is derived from the network bandwidth. Hop latency Th is the latency obtained from the lowest possible hop count in the network. Lastly, the wire latency Tw, is depended on the choice of n and k we make.

In the torus network, latency is also highly depended on the dimensions of the network. The interesting part in the torus network is that we get latency no matter what we change. If we increase the dimensions, the extra nodes will create latency. On the other hand, if we reduce the dimension, latency is attributed to the hop count. Therefore, it is very important that we make use of Equation 1.1 to generate the proper network dimensions for our specific problem. Later in the paper, we will see how the torus network performs under different dimensions.

## *4.3 Path Diversity*

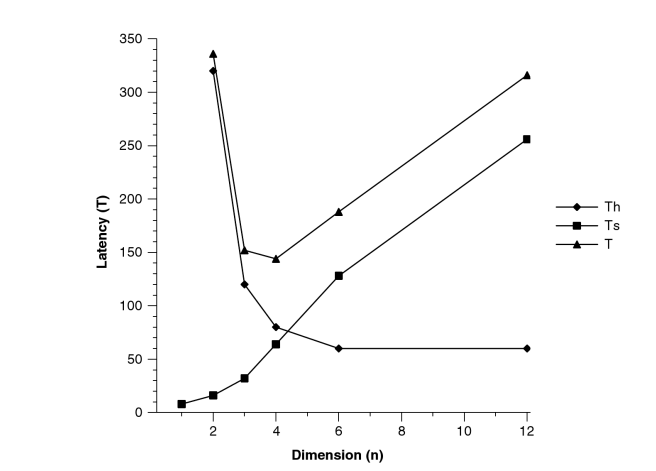
The last parameter we consider in this paper is path diversity. After looking at the topology of both the butterfly and torus networks, one might think that path diversity is a non-issue. On the contrary, path diversity becomes crucial when experimenting with arbitrary permutation traffic (3). There is a possibility of creating a bottleneck when not using a uniform distribution traffic pattern. Additionally, path diversity is what gives networks like the torus an advantage over others by providing alternate route when presented with a faulty channel.   
Let’s first start by analyzing the path diversity for a butterfly network.   
  
If look back to Figure 3, we can see that in multiple occasions, we run into a single point of failure issues. Meaning that if the particular path fails; our network fails. There are ways to help minimize the impact of a single point path failure but they are not the ultimate solutions. Single points of failure can still occur.



*Figure 5: Butterfly Network 2-ary 3-fly with extra stages*

Figure 5 illustrates how adding an additional stage to the butterfly networks can provide better path diversity (3). Studies have been done that prove that butterfly networks with additional stages improve networks performance under a uniform traffic pattern (4).

The one thing we do need to keep in mind is that adding diverse paths to a network also impacts other components of the network. If you recall, in the torus networks, all nodes are interconnected to each other providing the ultimate diversity path for the networks. This however, comes with significant impact to the overall latency.

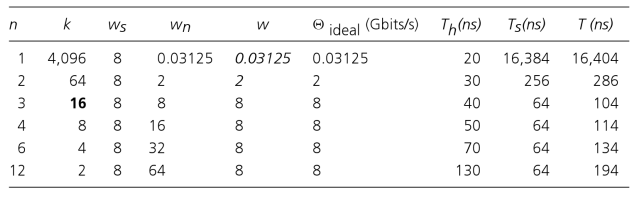
In the book, Principles and Practices of Interconnected Networks, the authors provide an excellent graph that depicts how the path diversity impacts the torus network. Therefore, leaving us to decide the proper path diversity combination that will provide the best result for our application.

*Figure 6: Tour Network Dimension vs. Latency Graph*

Figure 6, shows how hop, serialized, and overall latency is impacted as the dimensions for the torus network increases. If for instance, our application can handle an overall latency of approximately 200 ns, if an n value between 2 and 6 is selected for the torus network dimension.

## *4.4 Performance*

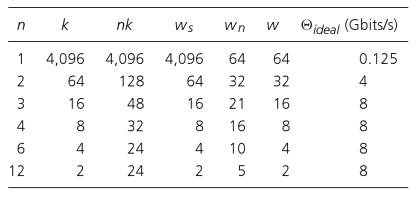
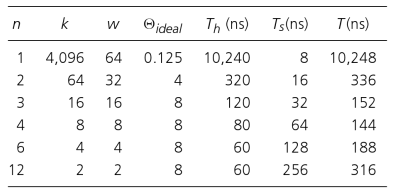
As a final analysis of these two networks, I would like to present the throughput and latency performance for each of these network configurations as presented in the Interconnection Networks textbook. These result where based on a 4096 node network using uniform traffic patters.



*Table 1: Butterfly Throughput and Latency*

In Table 1, we see the throughout and latency of a butterfly network with 4096 nodes in various values of n and k (3). Choosing a butterfly network with n=3 and k=16 will result in obtaining the best throughput (8Gbits/s) and lower latency (114ns).

A similar graph is also available for the torus network (3). In the torus case, choosing n=4 and k=8 will yields the larges throughput of 8Gbit/s and the lowest total latency of 144ns.



Torus Throughput Torus Latency

*Table 2: Throughput and Latency*

Note that if our design permits a slightly higher latency value, then the same could be done with a different combination of n and k.

# *5.0 My Observations*

Without a doubt, both of these network topologies are quite powerful and flexible. However, each has unique characteristics that make them ideal for certain applications. In this section of the paper, I will go over my observations and suggestions as to when to use which topology.

When selecting the butterfly interconnection network, we not only get a system that has a predictable amount of hops but also one with very simple routing protocols. This makes the butterfly network an ideal upgrade candidate to the traditional bus interconnection. Most importantly, the butterfly network makes it possible to run a multiprocessor system configuration, leading to a level of parallelism that we were lacking with the conventional bus implementation. Lastly, because butterfly networks can become quite powerful when additional stages are used, it also makes it a good option for when bandwidth is a design constraint (4).

With that being said, there are some significant issues with the butterfly network that prevent us from using it in high-speed systems. Specifically, the long wires required to build a butterfly network can cause signal degradation that might render the overall system to its knees. Another concerning factor to a designer might also be the lack of path diversity that the butterfly network possesses in its most basic configuration. As previously mentioned, a single node failure could disrupt the network flow and possibly cause the system to crash.

Alternatively, torus interconnection networks can be used in digital systems with more than a single multi-core processor. By design, torus networks have great path diversity that makes them ideal for systems requiring a certain level of redundancy and or load balancing. Torus channels are also bidirectional, therefore making better use of its resources. Ultimately, the torus networks perform well in high speed-speed systems due to its short wiring requirements.

# *6.0 Conclusion*

In my opinion, the torus network is a more powerful alternative to interconnection networks. To me, speed, is perhaps the one constraint that led us to conduct research in the area of interconnection networks in the first place. We need a solution that can help improve the overall performance of a multi-processor system. Furthermore, we need a network that can provide redundancy, load balancing and flexibility. Therefore, I consider torus networks the best option to replace the interconnection bus used in current or legacy systems.

# *7.0 Lessons Learned*

In conducting and writing this research paper, I learned so many things that will become very valuable as I start on my master thesis this upcoming semester. This paper gave me the opportunity to become familiar with the process of finding sources from the UNM library and Internet trusted sources such as IEEE. With this being the biggest writing assignment so far during of my masters degree, it also has given me a good sense what I am capable of doing in a few weeks worth of research work. I had the chance to practice my reading comprehension skills for interpreting what I read. But perhaps the most valuable lesson learned form this assignment was being able to use my critical thinking skills throughout the paper. Everything from selecting the topic, to conducting the research, and writing the paper made me think of the why behind every decision I made.

Overall, I think this was a great exercise, especially for those like me that will move on to doing a more in debt analysis in our master’s thesis topic.

# *8.0 Works Cited*

1. Andrews S. Tanenbaum, Computer Networks Age. 4th ed. Prentice Hall, 2003.

2. Hennessy, John, and David Patterson. Computer Architecture A Quantitative Approach. 5th ed., Morgan Kaufmann, 2011.

3. Dally, William J., and Brian Towles. Principles and Practices of Interconnection Networks. Morgan Kaufmann, IEEE 2004.

4. Arjomand, Mohammad, and Hamid Sarbazi-Azad. Performance Evaluation of Butterfly on-Chip Network for MPSoC’s, International SoC Design Conference (2008): 978-1-4244-2599-0.

5. Yao, Qing, Du XU, Hong Wang, Guo Jian, Lei Song. The Effect of Dimension on Performance and Complexity Comparison of Torus Interconnection Networks, Sanya (2008): 978-1-4244-1686-8

6. Peterson, Larry, and Bruce C. Davie. Computer Networks A Systems Approach Age 3rd Ed., Morgan Laufmann, 2003.

7. Ross, Kurose. Computer Networking A Top-Down Approach Age. 6th ed. Pearson, 2013.

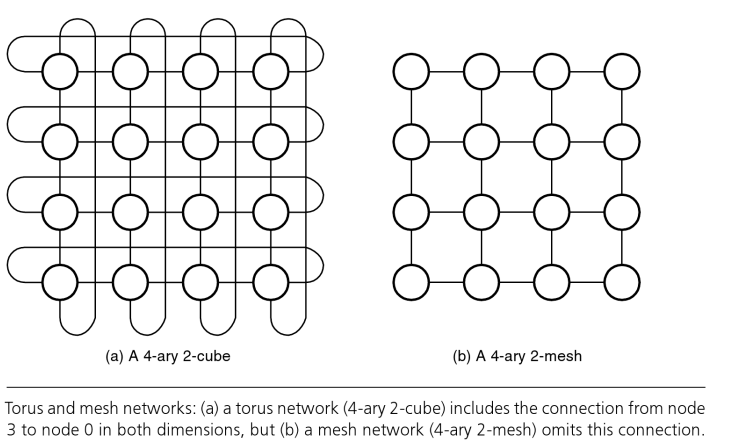
8. Mohanty, Ram Prasad, Ashok Kumar Turuk, Bibhudatta Sahoo. Performace Evaluation of Multi-Core Processors with Varied Interconnect Networks. IEEE Computer Society(2013): 978-0-7695-5127-2

9. Litz, Heiner, Maximilian Thuermer, Ulrich Bruening. TCLuster: A cluster Architecture Utilizing the Processor Host Interface as a Network Interconnect. IEEE International Conference on Cluster Computing (2010): 978-0-7695-4220-1.

# *9.0 Appendix A*

# HD:Users:alejflor:Desktop:Screen Shot 2014-11-30 at 8.42.21 PM.png

Butterfly Network  
Interconnection Networks Textbook



Torus Network

Interconnection Networks Textbook