The K-Ring: a versatile model for the design of MIMD computer topology

P. Kuonen
Computer Science Department
Swiss Federal Institute of Technology
CH-1015 Lausanne, Switzerland
E-mail: Pierre.Kuonen@epfl.ch

Keywords: Parallel computers, MIMD architectures, intercommunication networks, graphs.

Abstract: In this article we present a new graph topology for the design of intercommunication networks of MIMD computers. We show that, compared to the most used topologies, this original topology can achieve better performance and is much more versatile and scaleable.

1 INTRODUCTION

One of the main contributions of graph theory to parallel computing is to provide suitable topologies for the design of the network of MIMD high performance parallel computers. Indeed as soon as the number of processors to connect increases it becomes too costly to use a fully connected network. In this situation we try to use other topologies having good properties, such as a small diameter for a small degree and a large size. Unfortunately all solutions proposed up to now lead to the use of very rigid topologies. By “rigid” we mean topologies that strongly limit the choice of the size and the degree of the graph. In this paper we will show that the K-Ring topology, first introduced by P.Kuonen in 1993 (Kuonen 93), is a suitable topology for building the interconnection network of MIMD computers.

The rest of this paper is organized as follows: Sections 2 introduces the K-Ring topology. Section 3 compares K-Ring topology to the most used topologies. Section 4 presents the major advantage of K-Rings: versatility. Finally, Section 5 concludes this presentation.

2 THE K-RING TOPOLOGY

2.1 Main characteristics of graphs

In order to be able to compare different graph topologies, we need to define the characteristics we consider as relevant. For our purpose we can summarize the relevant characteristics of a graph as listed below:

- the *size* is the number of nodes of the graph.
- the *diameter* is the maximum distance between nodes of the graph. The distance between two nodes is defined as the minimum number of edges to cross to go from one node to the other.
- the *degree* of a node is equal to the number of neighbors of this node.
- a *regular* graph is a graph where every node has the same degree.
- the *cost* of the graph is an “ad hoc” parameter defined for our purpose as the number of edges of the graph.

Let us first mention that regular graphs are our favorite candidate as they are best suited for avoiding hot spots in the communication network. Next, as the size of the graph is directly related to the number of processors of the parallel machine, we want to build graphs with a high size while keeping a low diameter and a low cost. Unfortunately, these different characteristics of graphs are not independent. For a given degree the diameter and the cost increase with the size and for a given size we can decrease the diameter by increasing the degree, but in such case we increase the cost.

2.2 Definition of the K-Ring topology

Intuitively, the K-Ring topology can be seen as a graph built using K>0 rings where each ring goes over all the nodes in a different order. The value K is called the *dimension* of the K-Ring.

More formally the definition of the K-Ring topology is the following:

A K-Ring is a graph defined by the values listed below:

- the size N is an integer >0;
- the dimension K is an integer >0;
- K different positive integers ($a_1,...,a_K$), prime[1] with N and smaller than (N+1)/2.

The corresponding K-Ring is constructed as follow:

- the nodes are numbered from 0 to N-1.
- each node *i* is connected to the nodes $(i+a_j)$ mod N, for j in [1..K].

Conventions:

1. With such a construction each a_j defines a ring on the N nodes, this ring will be identified as the $ring_j$ and a_j will be called the $step_j$.
2. By renumbering the nodes it is always possible, for any K–Ring, to obtain a $ring_j$ with a $step_j$ equal to 1. By convention, we decide that this ring is numbered by 1. Consequently we will only consider K-Rings with a_1=1.

K-Rings are regular graphs with an even degree equal to $2 \cdot K$.

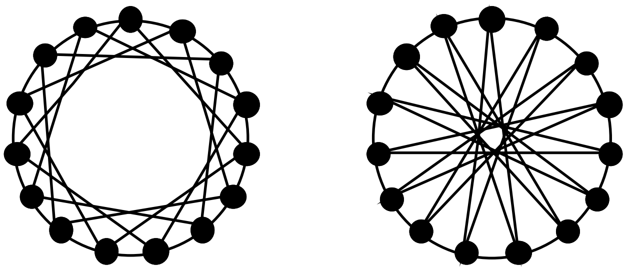

Figure 1: Two 2-Rings of size 15

[1] *a* is prime with *N* iff the LCM of *a* and *N* is equal to $a \cdot N$.

They are Caley graphs (Rumeur94) and they belong to the General Corodal Rings (GCR)(Bermond86). The size of a K-Ring can be any positive integer. For a given size and a given dimension there are many different K-Rings. Figure 1 shows two 2-Rings of size 15.

Readers interested in the classification of K-Rings can find more details in (Kuonen95).

3 K-RING VERSUS OTHER TOPOLOGIES

3.1 Currently used topologies

One of the main issue which is addressed by the designers of parallel computers is the diameter of the topology used to build the intercommunication network. This was the main reason of the success, a few years ago, of the hypercube topology. It was often mentioned that the diameter of an hypercube grows with the logarithm of the size. If this assertion is correct, it nevertheless lacks to mention that, for keeping this low diameter, we also have to increase the degree with the logarithm of the size. In other words, for a given size there exists only one hypercube with a given degree. Moreover the size of an hypercube must be a power of two. Because of this limitation hypercubes have been largely abandoned and, today, the designers of parallel computers use more “flexible” topologies such as the grid or the torus[2]. More recently, Meiko company has designed a parallel machine using a fat-tree topology (Meiko94).

In order to compare these topologies with the K-Ring we need to define clearly which graphs should be compared. For our analysis we decided to compare graphs of the same size and of the same degree. The motivation of such a choice is the following: when building a parallel computer the size is directly related to the number of processors and the degree to the number of I/O ports by node. Processors and I/O ports are hardware elements which have a price. Therefore the problem of designers is to decide, for a given size and for a given degree (i.e. for a given price), which are the best topologies for connecting the processors.

3.2 Comparison of diameters

Fat-tree topology is used to build multi-stage networks. In these networks not all the nodes of the

[2] It can be demonstrate that under some hypothesis, hypercubes are special cases of toruses.

graph correspond to computing nodes, but some nodes of the graph correspond to computing nodes while other nodes correspond to switch nodes. Figure 2 presents a fat-tree of size 12. Square nodes are computing nodes while round nodes are switch nodes. Computing nodes are those which contain processors. More details on the fat-tree topology can be found in (Leiserson85).

As it appears in Figure 2, fat-trees are not regular graphs. In order to compare this topology with a regular one, we have to decide which degree we assume for a fat-tree. In order to be fair in our comparison, we based our choice on the degree of the computing nodes. Indeed this degree determines how many I/O ports the processors (or the processor cards) have to possess. With this hypothesis the graph presented on figure 2 has a degree of 2.

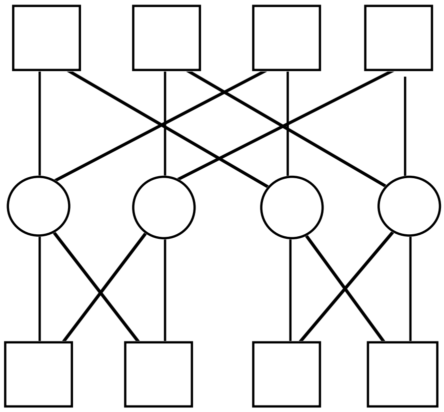

Figure 2: A fat-tree of size 12

In figure 3 we represent the diameter of toruses, fat-trees and K-Rings of degrees of 4 and 6 in relation to the amount of computing nodes .

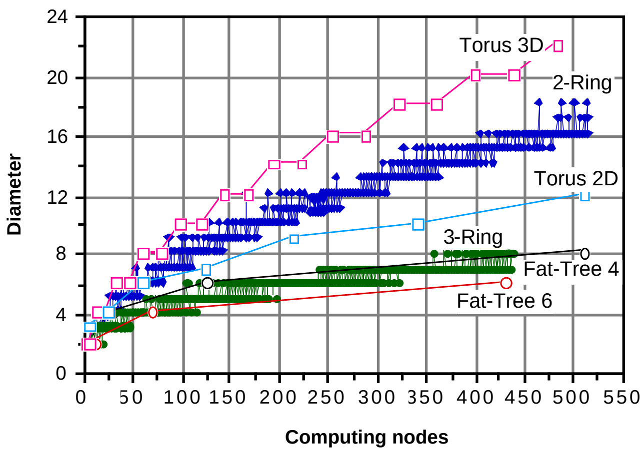

Figure 3: Comparison of diameters.

These results show that:

1. toruses always have the worst diameter;
2. fat-trees appear to have the best diameter;

Because K-Rings and toruses are both regular graphs, they have the same cost for the same size and the same degree. Figure 3 clearly show that, for the same cost, toruses have a much larger diameter than K-Rings.

To compare K-Rings and fat-trees we need to take into account the cost of graphs. Fat-trees have switch nodes and computing nodes therefore, for a given number of computing nodes (i.e. for the given number of processors) we have to use a fat-tree topology with more nodes than the corresponding K-Ring. Figure 4 compares the cost of fat-trees and K-Rings in relation to the amount of computing nodes. We can notice that the cost of fat-trees are significantly higher than for equivalent K-Rings. In other words, fat-trees provide a better diameter but at a very high cost.

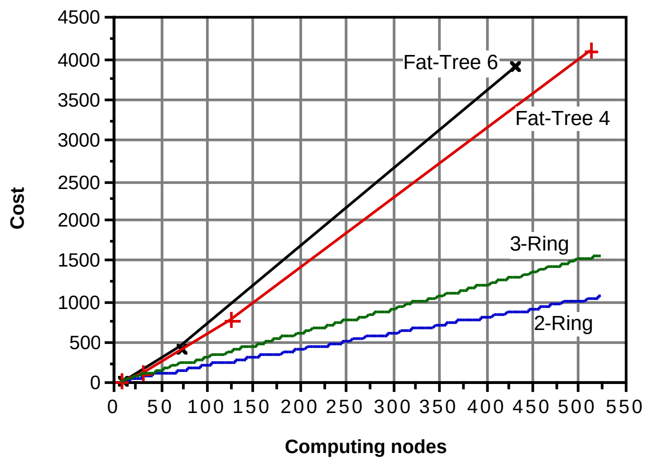

Figure 4: Comparison of costs

		Diameter		
Size	**Cost**	**Fat-tree**	**K-Ring**	**K**
32	256	4	2	8
128	1536	6	3	12
512	8192	8	4	16
2048	40960	10	5	20

Figure 5: Comparison of K-Rings and fat-trees of same cost.

It can be argued that at least fat-trees are able to provide a very low diameter and that, for high performance computers, the cost could be a secondary factor compared to the performance of the network which is the very critical element of a MIMD computers. Therefore in order to be able to decide which is the best topology, we must compare the diameter of topologies of the same cost.

Figure 5 presents such a comparison for fat-trees and K-Rings of same size and same cost. These results show that, at same cost, K-rings have a better diameter than fat-trees.

4. THE K-RING: A VERSATILE TOPOLOGY

The analysis presented in the previous section has shown that K-Rings are, in term of diameter and cost, a better topology than the most used topologies for designing the interconnection network of MIMD computers. Nevertheless K-Rings have another major advantage : it is a much more versatile topology.

First we can noticed that fat-trees and toruses does not exist for any size. For a torus of dimension K, the size N must be equal to $a_1 \cdot a_2 \cdot a_3 \ldots \cdot a_k$ where a_i are positive integers. The choice of a_i has a great influence on the diameter of the torus. The optimal diameter is obtained when $a_1=a_2=a_3\ldots=a_k=A$. Therefore we can assume that for a torus of dimension K the size should be A^K, with A>2. (when A=2, we obtain hypercubes). As the degree of a torus of dimension K is equal to 2K, size and degree of toruses are not independent values.

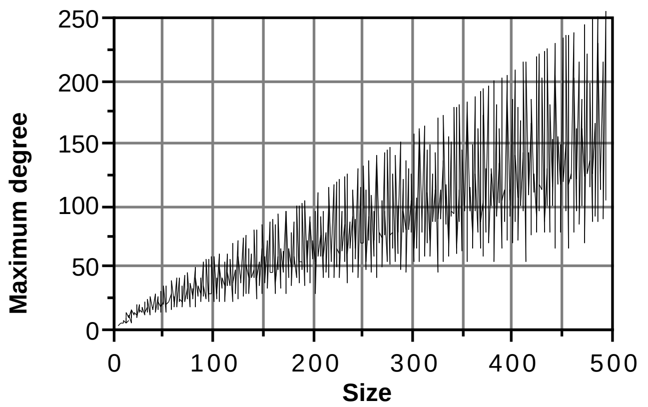

Figure 6: Maximum degree of K-Rings

Fat-trees are based on trees. The choice of the degree determines the possible sizes of the fat-tree. In fact the degree (as we have defined it in section 3) is equivalent to the branching factor of the fat-tree. Therefore the possible numbers of computing nodes for a fat-tree of degree d is: $2(d^K)$ where K is a non-null positive integer. As for toruses, degree and size are not independent values.

With K-Rings the situation is completely different. Size and degree are almost independent factor. There is only a limitation on the maximum value for the degree. This maximum value is equal to the number of positive integers prime with N and smaller than (N+1)/2. Figure 6 gives the maximum degree of K-Rings for sizes up to 500. On this figure we can see that this limitation is not very strong as soon as the size of the graph is not too small. For example, for a size greater that 60 we can guarantee a degree greater or equal to 10. Even if we do not have an analytic formula which allows us to calculate the maximum degree, figure 6 clearly shows that this value grows linearly with the size.

In fact, the main difficulty with K-Rings is that, up to now, we do not have many analytic formulas to determine their characteristics. For example, there are many different K-Rings having the lowest diameter for a given dimension and a given size, but we are not able to determine analytically these K-Rings and if these K-Rings correspond to the same graphs. On the practical point of view this ignorance is not very restrictive. Results presented in this article were obtained by enumerating the K-Rings. For reasonable sizes and dimensions this enumeration is possible on standard workstations. Nevertheless this ignorance on K-Ring is a motivating intellectual challenge.

Lets us come back to our original goal: to build the interconnection network of an MIMD computer. Thanks to K-Rings we are now able to build a MIMD computer of any size (of any number of processors) with a very low diameter. But we are able to do more than that; we can adapt the configuration of the machine to the user needs. For a given size, i.e. for a given power of the parallel computer, we can choose the dimension of the K-Ring according to the desired network performance. If, subsequently, the user needs to increase the performance of the network we can increase the dimension of the K-Ring without changing the number of processors. The opposite modification is also possible, we can increase the number of processors without changing the degree of the K-Ring. This flexibility which was not possible with the other topologies, allows us to optimize the ratio price/performance according the user needs.

CONCLUSION

In this article we have shown that K-Ring is a high-performance and flexible topology suitable for building the intercommunication network of MIMD computers. We demonstrated that K-Rings have better characteristics than the currently used topologies. K-Rings are seriously considering to be used for the design of the SwissTx-serie parallel computers (Gruber98).

K-Rings are still not well known graphs. Especially there is almost no analytic formula allowing to compute their characteristics (such as the diameter). Because of their flexibility it is possible to build a lot of, apparently, different K-Rings but we are still not able to determine which of those K-Rings correspond to same graphs. Some work has been undertaken at the mathematical department of EPFL in order to classify the 2-Rings.

REFERENCES

Bermond, J.-C.; and C. Delorme. 1986. “Strategies for Interconnection Networks: Some Methods from Graph Theory”, Journal of Parallel and Distributed Computing, No.3, p.433-449.

Gruber; R. and Dubois-Pèlerin; Y. 1998. “Swiss-Tx: First experience on the T0 system”, EPFL-Supercomputing Review No.10, November, pp.19-23.

Kuonen, P; 1995. “The K-Ring”, published in the Proceeding of the European Research Seminar on Advances in Distributed Systems (ERSADS), April .

Kuonen, P; 1993 “La programmation parallèle asynchrone et son application aux problèmes combinatoires”, PhD. thesis 1205, Swiss Federal Institute of Technology - Lausanne (EPFL), Computer Science Department, Lausanne, Switzerland, December.

Leiserson, C.E; 1985. “Fat-Trees: Universal Network for Hardware-Efficient Supercomputing”, IEEE Transaction on Computers, Vol. c-34, No.10, October, pp.892-901.

Meiko, 94; “Communication Network Overview” Http://www.meiko.com/info/NetworlOverview/Network/Overview.html.

Rumeur; J. de. 1994. “Communications dans les réseaux de processeurs”, études et recherches en informatique, Masson, pp.17-19.

BIOGRAPHY

Pierre KUONEN was born in Switzerland in July, 1958. He obtained a degree in Electrical Engineering from the Swiss Federal Institute of Technology (EPFL) in 1982. After six years of experience in industry he joined the Computer Science Theory Laboratory at the EPFL in 1988. He then started working in the field of parallel computing, and received his Ph.D. degree in 1993. Since 1994 he has been a scientific collaborator, heading the Parallel Computing Research Group of the Computer Science department. He is a lecturer in parallel computation courses.