Network Topology Scalability

Tze Jen Leong Melvyn Lim

Abstract: Scalability is arguably the *raison d'être* for interconnection networks. As demand for more higher performance supercomputers and higher bandwidth switches increases, there is increased interest in scalable networks. In this paper, we look at several distinct approaches to designing network topologies for scalability. Each emphasizes a different aspect of network topology scalability, thus resulting in disparate topology designs. We discuss the characteristics and strengths of each design and compare the designs.

1. Introduction

The size (number of processors) of computer systems has increased steadily over the last two decades as a result of rising demand for processing power. Ideally, the performance of a multi-processor system would scale linearly with the number of processors it has. This quality is virtually impossible to achieve with traditional bus-based systems as they grow larger. Interconnection networks emerged as the solution to system scalability. The increasing disparity between logic (on-chip) and wire (off-chip) speeds also makes interconnection networks essential to system performance.

In addition, packet switches used in the Internet are growing in size (number of ports, aggregate bandwidth). The internal fabrics of such switches, currently implemented as crossbars, do not scale well and thus are increasingly being implemented in multiple chips. Some form of interconnection network (e.g. Clos network) is then required to stitch the component switches together into a large fabric.

Such demand further propels the study of interconnection network scalability. Accordingly, there has been a shift in research emphasis towards networks that can efficiently accommodate large numbers of nodes. Given the breadth of issues and factors involved in the design of interconnection networks, researchers usually select one (or a few) preferred parameters, such as network diameter, to optimize and use as a basis of the superiority of their designs over other designs [Ni96]. In this paper, we select several attributes of network scalability that researchers have chosen to employ in their designs and examine some new network topologies that emphasize these attributes.

Section 2 briefly discusses several key considerations in making a topology scalable. Section 3 looks at a few topologies with different approaches to achieving network scalability. Section 4 compares these topologies and Section 5 concludes.

2. Scalability Considerations

What makes a topology scalable? There are many factors that are involved in scaling a network topology, including change in communication latency, bisection bandwidth, change in network throughput, packaging and layout cost, size and cost of expansion, change in routing and flow control, and change in performance to cost ratio. Many of these conventional factors can only be improved at the expense of others, thus requiring designs to either optimize for one or find an optimal compromise between many. We now briefly discuss several somewhat atypical factors that have been the basis of a few very different proposed scalable topologies.

2.1 Link Complexity

The link complexity of a topology refers to the number of links needed in the topology to be connected to a node as the network is scaled up. It depends on the node degree, which is the number of channels incident to a node. Link complexity has a direct correlation to hardware cost and complexity. Channels are implemented either as wide buses or optical cables for high bandwidth. Buses occupy a significant amount space on circuit boards and suffer bandwidth limitations, while optical cables are expensive and require optical-electrical interfaces. Since integrated chips and circuit boards have limited pin bandwidth, high node degree means narrower

channels, which means higher serialization latency. For scalability, a network topology should ideally have constant node degree. That is, the node degree of each node stays the same regardless of the number of times the network is scaled up. For example, a binary tree topology has low link complexity because of its constant node degree of 3, while a torus has relatively high link complexity because its node degree is 4 times its number of dimensions (4n for a k-ary n-cube).

2.2 Incremental Scalability

Incremental scalability refers to the number of additional nodes that have to be added to a topology when it is scaled up in order to maintain the same topology. (Strictly speaking, incremental scalability includes link complexity, but we consider just the increase in number of nodes here.) For example, each time a k-ary n-cube (torus) is scaled up (i.e. n is increased by 1), the number of nodes increases by a factor of k. Incremental scalability is important because it affects the ease and cost of increasing the size of a network. There might be a need for the processing power (number of nodes) of a system to be expanded but not by the amount dictated by the topology. If a network topology can be scaled up in small or constant increments, then the amount of additional wiring needed is probably small. The amount of modification to the routing algorithm needed, if any, is also of concern, since it might require significant software and/or hardware changes to the existing network for proper functioning.

2.3 Traffic Pattern

To examine the effect of traffic patterns on scalability, we first make a distinction between space and time division networks. As its name suggests, the communication bandwidth of a space division network is divided spatially among the links that traverse the network, since different links connect to different nodes in space. Nodes are connected using multiple links and data is transferred within the network by routing and switching traffic through these links. The mesh topology is an example of such a space division network. In contrast, a time division network divides the available bandwidth into time slices, which are then allocated to data traversing the network. In time division networks such as bus based networks, multiple nodes share a common link.

Space division networks have a tendency to leave certain links idle while other links in the network experience congestion. Time division networks perform considerably better than space division networks in this aspect, managing to utilize a greater fraction of the bandwidth by recycling bandwidth that is not used by idle nodes, and which would otherwise be wasted if the bandwidth could not be shared with other nodes. Thus, time division networks are better suited to handle dynamic nonuniform traffic.

Time division networks, however, do not scale up well. With more nodes on the shared link, it would require more substantial overhead to arbitrate for bandwidth on the link, effectively reducing throughput. Given these tradeoffs, we suggest that truly scalable networks should be able to accommodate bursty, unpredictable traffic, which is characteristic of certain applications, while somehow getting around the scalability problem of time division networks.

3. Scalable Topologies

In this section, we examine 3 distinct topologies that were designed with some aspect of scalability as their objective. We chose these topologies to reflect the diversity of approaches and ideas in scalable topology design. Each targets a different aspect of interconnection networks to optimize, and hence are rather specialized and may not represent the best or most authoritative method for specialized or generic network topology design.

3.1 Hybrid Tree

The Hybrid Tree topology proposed by E. John, Hudson, and L. John is a combination of binary trees and fat trees [JHJ98]. A binary tree topology exactly resembles a binary tree structure in graph theory and consists of a number of processing nodes, usually a power of 2, forming the leaves of the tree, and routers at the non-leaf nodes. The binary tree topology has a constant node degree of 3 and therefore requires minimal wiring even for larger numbers of processing nodes. However, the traffic on the top- level channels and the root router is very heavy, making the binary very bisection limited. The fat tree, proposed by Leiserson [Lei85], is a tree topology with channels of increasing width from the leaves to the root. The fat tree solves the bisection problem of the binary tree with higher bandwidth where traffic is likely to be

heaver. (Interestingly, fat trees more closely resemble real trees since the branches get thicker towards the root.) However, as the number of processing nodes increases, the bandwidth required at the root of the fat tree in order to eliminate the bottleneck there tends to exceed the practical wiring and electrical transmission limits.

The hybrid tree proposed by the authors consists of a fat tree on the upper levels of the tree and identical binary trees each connected to a leaf of the fat tree. The authors suggest using optical interconnects to overcome the bandwidth limitation problem at the root of the tree. Optical fibers offer bandwidth many orders of magnitude higher than that of electrical cables. Although the authors assert that they are not introducing a new topology, but rather exploiting new technology to realize the potential of an existing topology, we choose to showcase this topology because of the scalability it claims to offer.

The heights of the binary tree components are kept small to minimize the bottleneck effect, while the fat tree connecting the binary trees helps overcome the bottleneck. The single biggest advantage of the hybrid tree is the low link complexity afforded by the constant node degree within the binary tree portions of the topology. For each node added, only 3 channels need to be added. (This is assuming the fat tree is no modified.) However, if the topology is scaled up too much, then the fat tree portion must be augmented for it to continue to serve its purpose well. The hybrid tree could conceivably be expanded one processing node at a time, though this would create an unbalanced tree structure and require exceptions in its routing relation. Rather, to take advantage of the simple routing of a tree topology, the tree should be built with a complete set of leaf (processing) nodes. This means that the number of processing nodes is doubled each time the hybrid is scaled up one level. Also, in the case of expanding an existing hybrid tree network (with a complete set of leaf nodes), a large amount of rewiring must be done. The processing nodes must be disconnected, new routers must be connected where the processing nodes were before, then existing and new processing nodes must be connected at the leaves.

The authors assert that optical binary trees perform satisfactorily for 7 to 10 levels more than electrical binary trees, and that binary trees will suffice if built with optical interconnects for systems of up to 1024 processing nodes. They further state that a network of 16384 processors can be feasibly built with a hybrid tree consisting of a fat tree with 4 or 5 levels and binary trees of 9 or 10 levels each. We feel that the hybrid tree topology is scalable to the extent that it retains the desirable quality of low link complexity in a network of a significant number of nodes, although it is dependent on using optical interconnects, which may not always be economically viable.

3.2 Extended Incomplete Mesh

The Extended Incomplete Mesh is a specific case of a general approach to incremental design of scalable interconnection networks using basic building blocks proposed by Yang and Ni [YN00]. The networks are constructed in such a way that they can be scaled up by as few as one building block each time while requiring little or no rewiring and maintaining high bisection bandwidth and short diameter. Though the authors intend for this incremental design approach to be generalized for any topology, we refer only to the extended incomplete mesh for ease of describing the approach.

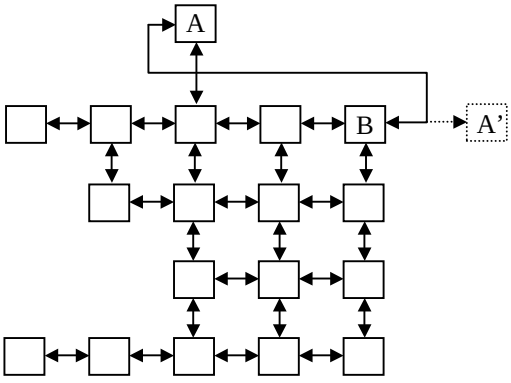

Figure 1: A 2-D Extended Incomplete Mesh

An incomplete mesh is formed by removing some nodes in a mesh, such that no remaining node is unconnected and only the channels connecting the remaining nodes in the original mesh remain. An extended incomplete mesh is an incomplete mesh with some additional channels between some pairs of boundary nodes along the same dimension but in opposite directions. An example of an extended incomplete mesh is shown in Figure 1, where the

west channel of A is connected to the east channel of B. Without A, the topology is an incomplete mesh. The extended incomplete mesh is designed to use X,Y dimensional routing.

When A is added to form an extended incomplete mesh, it is connected to its southerly neighbor and also B. As a result, A can also be seen as both being in its position shown in the figure and in position A'. This allows A to send messages to other nodes by X,Y dimensional routing.

When sending a message from A to another node using X,Y dimensional routing, the position A' can be used to compute the message header. A can be distinguished from other nodes using a flag bit.

If the additional channels that map A to A' are treated as X-direction channels, then there is no cycle in the channel dependency graph based on X,Y dimensional routing, and therefore the incomplete mesh is deadlock free. When more nodes are added, there are cases where the additional channels introduce alternative non-minimal routes, which need to be disallowed to ensure that the network remains deadlock free. When adding nodes, care must be taken to avoid deadlock.

Nodes must be added in a systematic way, shown in Figure 2. Suppose there is already a square or rectangular 2-D mesh. To add a node, suppose there is an axis *x* that lies in the X dimension and divides the mesh into two equal halves. The next node is added to the east side of the mesh on or beside *x* from inside to outside until a complete mesh is obtained. The same

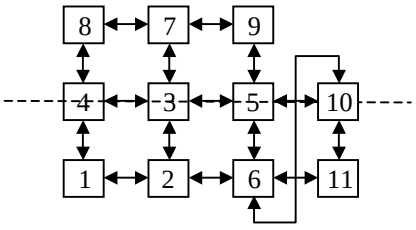

Figure 2: Order of adding nodes

method is applied for adding nodes on the north side of a complete mesh. Figure 2 shows the order in which nodes are added to obtain the incomplete mesh.

The incremental building approach obviously emphasizes incremental scalability indeed does well. It deserves credit as an innovative method to tackling the problem. The extended incomplete mesh can maintain bisection bandwidth and short diameter, and can also avoid deadlock. Rewiring is almost always unnecessary when adding nodes to the topology. However, routing becomes a little more complicated, since extra care must be taken to prevent deadlock even with previously deadlock free routing. Consequently, more information must be maintained at each node and routing tables may lose the simplicity provided by X,Y dimensional routing. Although incomplete meshes give a lot of freedom in constructing irregular topologies, a regular topology is still desired to simplify routing. The extended incomplete mesh is very scalable, though more on a small, incremental scale than on a large, expansionary scale, since it is ultimately similar to a mesh topology.

3.3 Virtual Bus

The Virtual Bus architecture was designed by K. C. Lee to handle the bursty traffic generated by the parallel processing of data intensive applications while placing a high priority on scalability [Lee93]. While many previous instances of parallel processors using time division networks have glossed over the scalability problem mentioned in Section 2.3 by exploiting the locality of data accesses to reduce the amount of traffic on the interconnection network, certain applications do not exhibit this property. Hierarchical topologies that take advantage of the locality property are rendered useless in these cases. Yet all applications that are supported must have their performance requirements met by the interconnection network. With these considerations in mind, the author proposes an architecture to tackle afresh the scalability problem for dynamic nonuniform communication.

The virtual bus employs a packet-parallel time-space-time switching architecture, consisting of multiple time division switches that are connected by a single nonblocking space division switch. The architecture exploits the ability of time division networks to share bandwidth among multiple nodes and handle nonuniform traffic without significant performance degradation, while using the central space division switch to scale up the time

division switches, which do not scale well themselves. The object of the architecture is to enable the entire interconnection network to appear virtually as a bus to the nodes connected to it.

The configuration proposed by the author and shown in Figure 3 involves a three-dimensional space division switch, implemented as an output-buffered crossbar switch, that facilitates communication between M bus clusters, which are in fact time division networks and can be further partitioned as necessary into local buses. Figure 4 aids in the visualization of the functionality and structure of the space division switch. The packet size of the network is also the height of the switch, H. Since switching entire packets in parallel requires very fast arbitration, using a high speed variable round-robin arbiter is suggested.

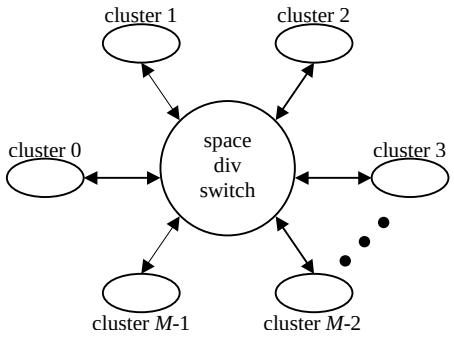

Figure 3: Virtual bus configuration

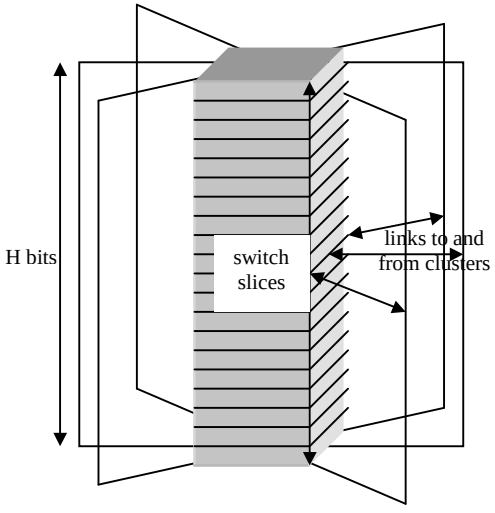

Figure 4: Datapath for 3-D space switch

To evaluate the scalability of the virtual bus, we assume that the packet size is L and that the datapath width, H is bounded by $1 \leq H \leq L$. Everything else being equal, a N/H by N/H switch with a H bit datapath, like the one employed in the virtual bus architecture, is equivalent to a N input by N output 1-bit wide switch in terms of switching throughput. The space complexity of a N by N bit-serial crossbar switch is $O(N^2)$ as opposed to $O(N^2/H)$ for our space switch of height, H.

For the time switched portion of the network, since each of the N nodes requires H drivers to drive the packet-wide datapath, we have complexity of $O(NH)$. Setting the constants for the asymptotic complexities of the space and time switched networks equal, we arrive at a total complexity of $O(N(N/H + H))$. (Discarding the constants in the asymptotic complexities gives us just a ballpark figure for the total complexity, but this will suffice for our scalability evaluation.) From this figure, we see that the optimal H is roughly $N^{1/2}$, which puts the optimal complexity of the virtual bus at $O(NN^{1/2})$, which is a much better result than the $O(N^2)$ we would expect from the simple crossbar switch that has similar functionality and throughput.

The virtual bus scores points just for its simple and novel idea of fitting together dissimilar networks in an attempt to harness the positives that each type of network brings. It is also easy to incrementally scale up the system by adding further cluster buses and space division switches if needed to facilitate more nodes. However, it appears that the network is not truly scalable if the packet size is not scaled accordingly as well. Keeping the packet size constant and adding more nodes does not increase the bisection bandwidth, while adding more space division switches will increase the network complexity significantly. We feel that the scalability of this system is overly dependent on the usage of an optimal packet size.

The architecture suffers from several disadvantages like the short amount of time in which it has to perform arbitration and the increased likelihood of electrical issues regarding the many packet-wide buses in the system, which may arise when scaling up, especially if signals have to go off chip. In this scenario, the number of pins on a chip and the clock frequency may limit performance.

Although we note that the author relied on many assumptions in arriving at a favorable result, the architecture does have some merit and, while probably not achieving the ideal scalability outlined in the paper, still represents an improvement in scalability over that of purely space or time division networks. While far from perfect, we tended to be more lenient when examining the virtual bus because it does not claim to be the answer to all scalability concerns, but was designed specifically to meet performance requirements for certain types of applications.

4. Comparisons

As in all engineering activity, there exist tradeoffs between the various architectures presented. A summary of these advantages and disadvantages are presented in Figure 5.

The hybrid tree has constant node degree, giving rise to low link complexity, low serialization latency and fewer channels needing to be added per additional node. Its binary tree structure allows for simple routing and its fat tree structure provides good bisection bandwidth. However, it is not truly scalable because channels on the fat tree must be modified to have increased bandwidth whenever levels of nodes are added in order to prevent bisection bottlenecks. It is not incrementally scalable because the number of nodes should be doubled each time to make a complete binary tree in order to keep routing simple. Significant rewiring is required to expand the network and it is dependent on optical interconnects beyond some small number of levels.

The extended incomplete mesh is very incrementally scalable, as one node can be added at a time. Because nodes are added in a symmetric way, it retains good bisection bandwidth and short network diameter. It is possible to maintain a deadlock-free mesh. Complicated routing, which requires significant extra information to be kept at each node, is needed to fully exploit the good qualities of the network. The topology is limited to mesh qualities since it must be restricted to simple topologies to retain incremental scalability.

The virtual bus handles bursty traffic well because of its use of time division switches close to the nodes. Being a bus network, one node can be added at a time, making it very incrementally scalable. However, its scalability is somewhat dependent on the network using an optimum packet size, only a short period of time is available for arbitration, and having a packet-wide bus structure makes packaging harder and gives rise to the possibility of electrical issues, both of which may affect performance.

It is difficult to assess the relative worth of these three systems, especially in the absence of realistic network traffic and performance measurements with respect to scalability. Each have their strengths and weaknesses and would probably perform well or poorly relative to the other two based on the environment in which it is used. Hopefully, we will be able to predict the architecture that works best based on the characteristics of the networks that we have researched.

We can, however, safely say that the extended incomplete mesh does give the best results with respect to general-purpose scalability among the three. The hybrid tree is not very scalable and depends on optical interconnects beyond a certain size. While the virtual bus works well with certain applications and traffic patterns, it is not truly scalable either. The mesh also has its weaknesses and does not yet give an entirely satisfactory solution to the scalability problem, but appears to be the topology with the widest application among the three.

	Hybrid Tree	Extended Incomplete Tree	Virtual Bus
Advantages	constant node degree simple routing good bisection bandwidth	very incrementally scalable retains good bisection bandwidth retains short network diameter deadlock can be avoided	handles nonuniform traffic well very incrementally scalable
Disadvantages	not truly scalable not incrementally scalable significant rewiring needed dependent on optical interconnects	routing is complicated extra information kept at each node limited to mesh qualities	dependent on packet size short time for arbitration physical and electrical constraints

Figure 5: Table of pros and cons

5. Conclusion

Issues regarding scalability will continue to grow in importance as interconnection networks attempt to match the advances in processing power. This problem is not new and is undoubtedly the right one. Not only will networks have to provide more bandwidth between chips, but also as transistors shrink, manufacturers will be able to squeeze more onto a chip, requiring efficient interconnection networks running both on and off chip, handing interconnection network designers problems on two fronts. The solutions put forth in the papers address different areas of a large problem and vary in scope. To give a more easily assessable and usable solution, it would help to have the original problem recast as a more closely defined problem, like in the virtual bus example, based on expected operating conditions, although papers that are more general in scope could be of use as repositories of knowledge for subsequent researchers and designers.

There are many issues connected with the design of scalable networks and the papers did their best to address issues that their authors deemed most critical. Ni suggests a list of considerations for scalable networks such as range of scalability, incremental scalability, performance metrics, switching techniques, routing and flow control, and reliability [Ni96]. Undoubtedly many key issues were not elaborated on in the papers and a solution that touches on all these aspects would be ideal. What we hope to see is a solution that attempts to answer the fundamental question of scalability while managing to preserve desirable properties and not compromise other aspects of the network, but that would be a tall order.

6. References

[Ni96] L. Ni, "Issues in Designing Truly Scalable Interconnection Networks", in Proceedings of the ICPP Workshop on Challenges for Parallel Processing, pp. 74-83, 1996.

[JHJ98] E. John, F. Hudson, and L.K. John, "Hybrid Tree: A Scalable Optoelectronic Interconnection Network for Computing", Proceedings of the Thirty-First Hawaii International Conference on System Sciences, volume 7, pp. 466-474, 1998.

[YN00] M. Yang, and L. Ni, "Incremental Design of Scalable Interconnection Networks Using Basic Building Blocks ", IEEE Transactions on Parallel and Distributed Systems, volume 11 issue 11, pp. 1126-1140, 2000.

[Lee93] K. C. Lee, "A virtual bus architecture for dynamic parallel processing", IEEE Transactions on Parallel and Distributed Systems, volume 4 issue 2, pp. 121-130, 1993.