Qatar Univ. Sci. J. (1995), 15 (2) : 319 - 331

PARTIAL CONNECTION : A NEW APPROACH TO IMPROVING THE BANDWIDTH OF BANYAN NETWORKS

Hamed Nassar
King Faisal University, Saudi Arabia

التوصيل الجزئي : طريقة جديدة لتحسين عرض نطاق شبكات بانيان

حـــامـد نصـــار

شبكات بانيان من شبكات التوصيل المرشحة للعمل في المعالجات المتوازية الكبيرة، أي التي تحتوي على مئات أو حتى آلاف من المعالجات. ويعود السبب في ذلك الى ميزتين هما قلة التكلفة والتوجيه الذاتي. بيد أن هذه الشبكات تعاني من مشكلة خطيرة هي قلة عرض النطاق. ولسوء الحظ فإن هذه المشكلة تزداد حدة مع زيادة حجم الشبكة أو تحميلها.

هذا البحث يقدم طريقة جديدة لحل هذه المشكلة هي طريقة التوصيل الجزئي. وتعتمد فكرة هذه الطريقة على حقيقة أنه عند ترك بعض أطراف الشبكة غير موصلة فإن كمية أقل من التطاحن تحدث، مما يحسن عرض نطاق الشبكة. ويستطلع البحث جدوى هذه الطريقة باستخدام نموذج تحليلي دقيق. وقد أظهر النموذج، الذي تم التأكد من صحة عمله بالمحاكاة، نتائج مشجعة جدا لطريقة التوصيل الجزئي، حيث يمكن الحصول بواسطتها على تحسين اختياري لعرض النطاق. وقد وجد أن قدر التحسين يعتمد ليس فقط على كمية الأطراف غير الموصلة، ولكن أيضا على الناحية التي نستخدم الطريقة فيها وعلى ترتيب الأطراف غير الموصلة بالنسبة لتلك الموصلة. كما يعتمد التحسين على حجم الشبكة وتحميلها.

طريقة التوصيل الجزئي لها ثلاث ميزات رئيسية. أولاً ، هي تعطي تحسينا اكبر كلما زاد حجم الشبكة أو تحميلها. ثانياً ، تحافظ على خاصية التوجيه الذاتي للشبكة الاصلية. ثالثاً، سهل جدا تنفيذها. ويمكن استخدام هذه الطريقة عندما يكون الأداء أكثر اهمية من التكلفة. كما يمكن أن تستخدم فقط حول الأطراف النشطة جداً، أو ما يعرف بالبؤر الساخنة إذا كانت التكلفة مهمة.

Key Works : Banyan Networks, Blocking Networks, Computer Architecture, Mutliprocessing, Multistage Interconnection Networks, Packet Switching, and Performance Analysis.

ABSTRACT

Banyan networks have been proposed as interconnection networks for large multiprocessors, those containing hundreds or even thousands of processors. Their attractiveness is attributed to two features : low manufacturing cost and self routing. However, these networks have a serious problem, low bandwidth. Unfortuntely, this problem gets worse as the size and/or the load of the network increases.

In this paper we introduce a novel approach to solve this problem, partial connection. The idea is that when some of

the network terminals are left unconnected, less blocking takes place, and therefore the bandwidth of the network improves. We explore this approach using an exact analytical model. The reslts obtained from the model, which have been validated by extensive simulation studies, are very promising. An arbitrary bandwidth improvement can be obtained. We have found that the exact amount of improvement depends not only on the amount of unconnected terminals, but also on their side and their arrangement with respect to the connected ones. It depends also on the network original size and the load.

The partial connection approach has three primary advantages. First, it works better as the size and / or the load of the network increases. Second, it preserves the same routing procedure of the normal network. Third, it is extremely easy to implement. The last advantage is due to the fact that the approach preserves the architectural structure of the normal network, and does not require any alteration in the design of its original components.

The approach can be a convenient option in situations where performance is more important than cost. It can also be applied only around very active terminals, e.g. hot spots, if cost is a concern.

1 Introduction :

An interconnection network is used in a multiprocessor to connect the processors to the memory modules [1]. Typically, the processors are regarded as sources and are thus connected to the network inlets, and the memory modules are regarded as destinations and connected to the network outlets. The role of the network is to establish communication paths between the processors and the memories as desired.

The crossbar switch is the ideal interconnection network as far as performance is concerned. A switch with n inlets and n outlets, can always establish up to n arbitrary paths between its inlets and outlets. Furthermore, if there are (x < n) paths currently established, the switch can establish up to n - x additional paths without disrupting, the existing ones. An interconnection network with this capability is called a blocking - free network; otherwise it is called a blocking network.

The performance excellence of the crossbar is marred by its cost, $O(n^2)$. This cost becomes excessive when n is large. In this case, Multistage Interconnection Networks (MINs) become an economically - better alternative [2].

A MIN is a set of small crossbar switches, called switching elements (SEs), arranged in stages. Each two successive stages are connected together with a set of links. The connection pattern is arbitrary, provided that each MIN inlet can communicate with every MIN outlet. Basically, a MIN is defined by three factors : the sizes of the SEs, the number of stages, and the connection patterns between the stages.

A large number of MINs have been proposed [3], differing in at least two of the three defining factors. Among these MINs, the ones that have drawn the most attention are banyan networks [4]. These network are characterized by having only one path between any inlet-outlet pair.

Of the many banyan networks reported in the literature, the most attractive has been the Binary Banyan (BB) network [5]. It is so called because it uses binary (i. e., 2 X 2) SEs. Figure la shows a 16 X 16 BB network. It has 4 stages, each with 8 SEs. In general, an n X n BB network has $\log_2 n$ stages, each with $\frac{n}{2}$ SEs. Since the BB network uses square SEs, (i.e., number of inputs equals number of outputs[1]) it is inherently square.

It is interesting to note that the BB network has been

[1] We will use the words inlets and outets to refer to the terminals of a switching network, whether a crossbar switch or a MIN, and the words inputs and outputs to refer to the terminals of an SE of a MIN. For the terminals of the outer stage SEs, the context will determine which word to use. If a terminal is being treated as an SE terminal, it is an input; if it is being treated as a MIN terminal, it is an inlet. Incidentally, we will say a crossbar switch if the switch is used as a switching network, and an SE if the switch is used as a building

reintroduced repeatedly in the literature, each time under a different name. The omega, baseline, shuffle and exchange, Delta - 2, cube, etc., are just variations of the BB network. They differ only in the connection pattern between the states, and they agree in the other two defining factors. Two networks are really different if they differ in at least two factors. It has been shown [3] that the basic properties of all the variations of the BB network are the same.

The importance of the BB network stems from two features : low cost and self routing. The low cost of the BB network is the result of using the smallest, SEs possible, 2x2. The cost of the BB network is $O(n \log_2 n)$.

The self routing capability is achieved in the BB network by using self routing SEs. A selfrouting SE is built such that any input can connect itself to any output, based on a routing bit placed on it. Typically, if the bit is 0 the input connects itself to the upper output, and if the bit is 1 the input connects itself to the lower output. This is depicted in Figure 1b. As can be seen, if the two inputs have different bits, there is no problem. However, if the two inputs have similar bits a conflict arises. In this case, only one input (usually chosen at random) gains access to the requested output, and the other is *blocked.*

To establish path in the BB network between a given inlet and a given outlet, the routing bits are placed on the inlet. The number of these bits should be equal to the number of stages, as one bit is used by each stage. The first SE, to which the inlet is connected, uses the first bit to set itself up. The SE then passes along the remaining bits to the next SE, in the next stage. The next SE repeats the same procedure. This is continued until the path is established.

There is a fascinating aspect to the self routing capability of the BB network : routing bits for a given path are just the digits of the number of the required outlet when this number is represented in binary. Figure la shows (in dark kine) a path established between inlet 3 and outlet 12 (binary 1100). The four routing bits are shown in bold face next to the respective switches.

The self routing capability is a major asset to the BB network. Networks without this capability, including the crossbar switch, require a dedicated central routing unit. This unit requires extra hardware, introduces delays, and presents a reliability weakness point.

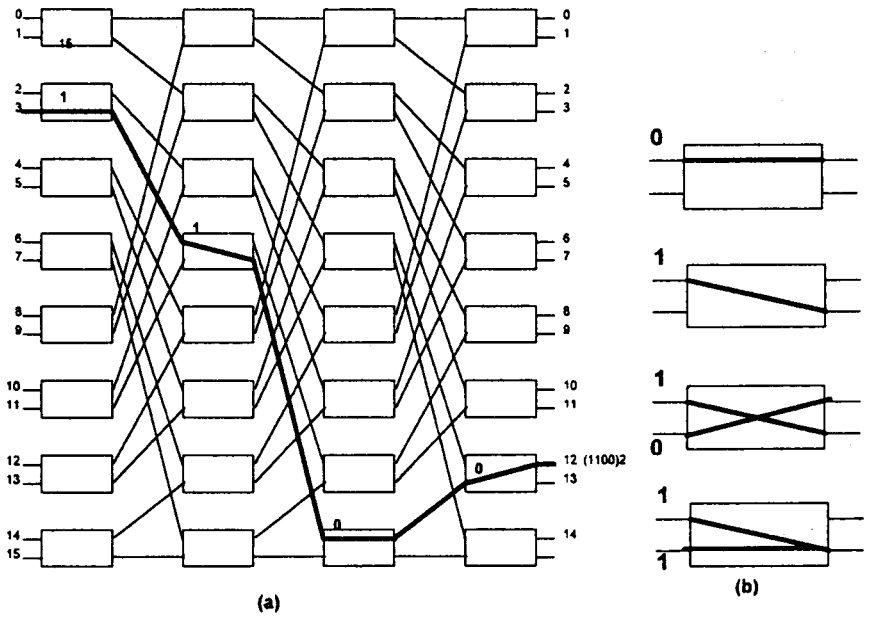

Figure 1 : a) 16 x 16 binary banyan (BB) network, with a path between intlet 3 and outlet 12. b) Switching element (SE) at different states

2 Low Bandwidth Problem

The self routing capability of the BB network is due primarily to the single-path property of this network. It is ironic that this same useful property is the cause of a major problem, low bandwidth. To illustrate how this problem comes about refer to Figure 1a again. One cannot establish a path, say from inlet 10 to outlet 14, in the presence of the shown path; the new path will conflict with the existing one in the second stagge. In general, each established path in the BB network blocks a set of other paths (the exact number of these paths depends on the size of the network).

This type of blocking is called network blocking. In switching networks, thhere is another type of blocking, destination blocking. Destination blocking arises when two or more sources reference the same destination. Assuming no network blocking, all the paths, except one will be blocked right at the destination.

In the crossbar switch network blocking is not present. One can always establish a path from an inlet to a free outlet, regardless of whether there are already established paths in the switch. However, when two or more sources reference the same destination, destination blocking occurs.

Bandwidth has been used as a measure of blocking, both network and desination, in switching network operating synchhrounously [2]. In synchronous operation, time is divided up into intervals, each called a cycle. At the beginning of each cycle, sources may reference destinations. A source that references a given destination is called an active source. When a source becomes active, the inlet to which it is connected becomes active as well. If the network being used is a crossbar switch, an active source proceeds by submitting the number of the referenced destination to a central routing unit. On the other hand, if the network is a BB network, the active source proceeds by submitting the routing bits to the network. In either case, the source waits for a confirmation that a path has been established to the desired destination. If a path is established it can last for only one cycle. Under independent source operation, it is likely that some paths will fail to be established because of blocking, either network or destination. The network bandwidth is then defined as the average number of successfully established paths per cycle. Throughout this paper, however, we will use normalized bandwidth. Normalized bandwiidth, denoted by BW, is the bandwidth just defined divided by the maximum number of possible paths, in this case n.

Assuming statistically independent and identical cycles, we can see that the bandwidth of a given network depends on three factors : network size, network load, and reference pattern. In the most general case, called the generalized model [9], the network load is represented by a vector.

$$P = [p_0, p_1, \ldots, p_{n-1}],$$

where p_i is the probability that inlet i will be active at the beginning of a cycle. On the other hand, the reference pattern in the generalized model is represented by a matrix.

$$R = \begin{bmatrix} r_{00} & r_{01} & \cdots & r_{0,n-1} \\ r_{10} & r_{11} & \cdots & r_{1,n-1} \\ \vdots & \ddots & & \cdots \\ r_{n-1,0} & r_{n-1,1} & \cdots & r_{n-1,n-1} \end{bmatrix}$$

where r_{ij} is the probability that source i references destination j.

Given P and R for an n x n switching network, one can evaluate the bandwidth BW under the generalized model as shown in the appendix.

There is a special case of the generalized model, called the equiprobable model [10]. In this model the elements of P are identical, i. e., $p_i = P$ for all i, and thus P represents the (normalized) load of the network. Furthermore, the elements of R are identical, i. e., $r_{ij} = r$ for all i and j, and thus r represents the connection pattern of the network. The evaluation of the bandwidth under this model is much simpler than under the generalized model and is also shown in the appendix.

Due to network blockung, the bandwidth of a BB network is inferior to that of a crossbar switch having the same size and having the same load. In fact, the bandwidth of the crossbar represents an upper bound for the bandwidth of any MIN. The bandwidth inferiority of the BB network compared to that of the crossbar increases as the size and / or the load of the network increases, as shown in Figures 2 and 3. In these two figures, the equiprobable model is assumed.

Figure 2 shows the bandwidth vs. load for both a crossbar switch and a BB network of the same size, 32 X 32. The bandwidth of both switching network increases as the load increases. However, the bandwidth of the crossbar at a greater rate than that of the BB network. Figure 3 shows the bandwidth vs. size for both the crossbar switch and the BB network under the same load, $P = 0.9$. The bandwidth of both switching networks decreases as the size increases. However, the bandwidth of the crossbar decreases at a greater rate than that of the BB network.

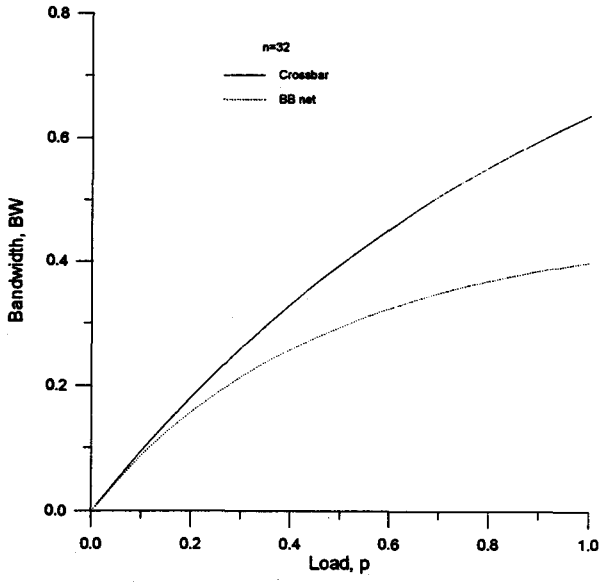

Figure 2 : Bandwidth vs. load for a crossbar and a BB network, both of size 32 x 32

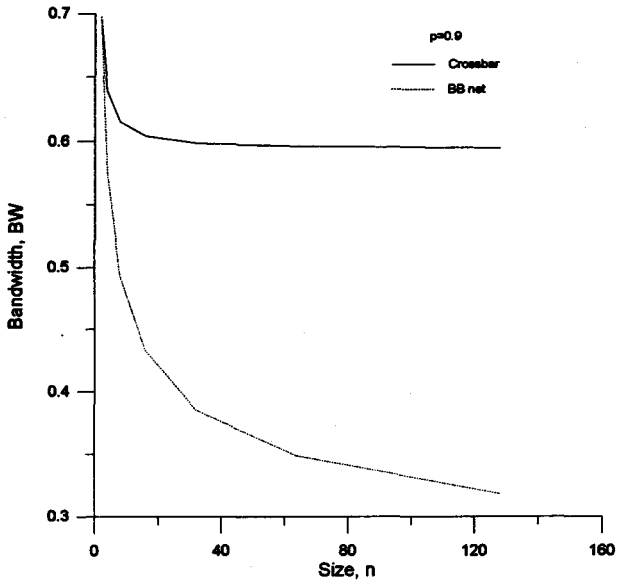

Figure 3 : Bandwidth vs. size for a crossbar switch and a BB network, both under the same load p = 0.9

Much research have been carried out to devise approaches to improve the bandwidth of the BB network. In the load distribution approach [5], extra stages are prepended to the network. These stages have been found useful in improving the bandwidth. But for each extra stage, a delay of one stage-passing-time is introduced. The extreme of this approach is to prepend a sorting network to the BB network. The idea is that the BB network is able to successfully establish a set of n paths to n outlets, provided these paths originate at n specific inlets. The role of the sorting network then is to receive the input traffic and present it to the BB network at the inlets that can be routed to the required outlets without blocking. If the sorting network is based on the Batcher Bitonic Sort network [6], it has $\frac{n}{4}(\log_2 n)^2 + \log_2 n$ stages. Aside from the large amount of extra hardware that such sorting network consumes, it introduces more delay than that caused by the BB network itself.

Another approach to alleviate the low bandwidth problem of the BB network is replication [7]. In this approach multiple copies of the network are used in parallel. This approach does not introduce delays, but requires a great deal of extra hardware, especially when more than two copies are used. It also requires intelligence at the sources and destinations to deliver and pick up the traffic to or from the right copy. A variation of replication, called dilation [7], suggests the use of multiple copies of the links only. However, this approach makes the intelligence required not only at the sources and destination, but also at the SEs themselves.

Still another approach is to use SEs wiith buffers[8]. Although this approach improves the bandwith greatly, it complicates the design of the SEs as well as the operation of the network. This approach is also not suitable for real-time applications.

3 The Partial Connection Approach

The idea of the partial connection approach is simple: Abandon some of the network terminals (inlets and/or outlets) and connect the sources and destinations to the remaining terminals. This creates a partially connected (PC) network. Since the connected terminals in a PC network are less than the existing terminals, less blocking takes place, thereby increasing the bandwidth. Figure 4 shows a 12 X 8 PC network, created from the 16 X 16 BB network of Figure 1a. Four inlets and six

outlets of the original network are abandoned. Notice that the terminals of the PC network carry the same numbers as in the original BB network. This is to preserve a useful aspect of the self routing capability, namely the direct generation of the routing bits from the number of the desired destination.

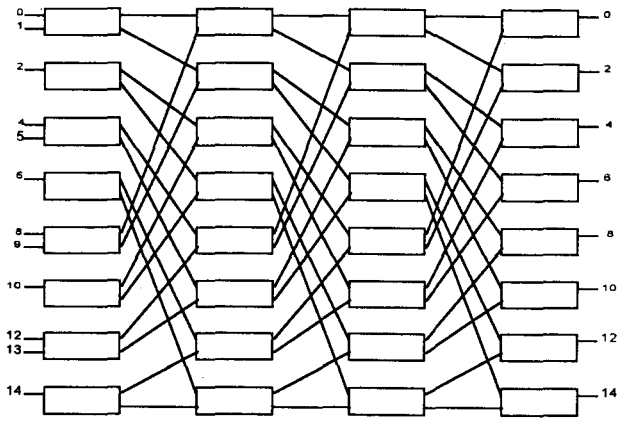

Figure 4 : 16/0.75 – 0.5 partially connected (PC) network

Now we will use the generalized model (given in the appendix) to study the performance of PC network. In this model, we substitute a 0 for the element Pi in the load vector P if inlet i is abandoned, and for the element r_{ij} in the reference matrix R if either the inlet i or the outlet j are abandoned. Otherwise the elements can assume arbitrary non zero values, with a constraint for R, namely $\sum_j r_{ij} = 1$, for all i.

From this study, it is has been found that three configuration factors affect the bandwidth of a PC network : the amount of partial connection, the side on which partial connection takes place, and the partial connection pattern.

The amount of partial connection affects the network bandwidth greatly. Taking a given side, the less terminals you connect the greater the bandwidth you obtain. In the extreme, when only one terminal is connected the bandwidth reaches a peak value under the given load. But clearly, no one would want to go to this extreme. In addition, the same amount of partial connection improves the bandwidth differently depending on whether it is done on the inlet side, the outlet side, or both. Usually the application at hand would determine which side to partially connect and at what amount. For example, if the number of sources is equal to the number of destinations, one would be forced to implement partial connection on both sides and with the same amount. The PC network of Figure 4 is partially connected on both sides, at the amount of 75% on the inlet side and 50% on the outlet side.

Finally, the pattern of partial connection has a prominent effect on the bandwidth of a PC network. The connection pattern on the inlet side of an $n \times n$ PC network can be described by the connection vector.

$$C_{in} = [C_0, C_1, ..., C_{N-1}]$$

where

$$C_i = \begin{cases} 1 & \text{if inlet } i \text{ is connected} \\ 0 & \text{if inlet } i \text{ is abandoned} \end{cases}$$

Similarly, the connection pattern on the outlet side of an $n \times n$ PC network can be described by the connection vector C_{out}, having a similar definition to that of C_{in}. Given C_{in} and C_{out} for a network, one can have a complete picture of the connection pattern of that network. For example, if C_{in} = [1, 1, 0, 1] and C_{out} = [1, 0, 0, 1] for a particular network, one concludes that inlet 2 and outlets 1 and 2 are abandoned, and all the other inlets and outlets are connected.

There are two types of connection patterns, regular and irregular. A pattern is regular if the 1s and 0s of its repesentative vector are arranged in groups of equal size each, and is irregular otherwise. Thus, both of the two vectors just mentioned, namely C_{in} = [1, 1, 0, 1] and C_{in} = [1, 0, 0, 1], are irregular, whereas both of the connection vectors of the PC network of Figure 4.

$$C_{in} = [1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0]$$

$$C_{out} = [1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0]$$

are regular

Note that the regularity of a pattern allows expressing its vector by an equation for its general element. For example, in Figure 4, the i th element of C_{in} is

$$C_i = \begin{cases} 1 & \text{if } ((i+1) \bmod 4) \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

To specify a PC network fully, one should mention the original size of the network, i. e., before adopting the partial connection approach. In addition, one should mention the side, amount and pattern of partial connection. To specify these factors in shorthand, we will use the notation.

$$n / x_{in} - x_{out} / C_{in}, C_{out}$$

where

- n is the size of the original network, i. e., before adopting the partial connection approach.
- x_{in} - x_{out} are two values, less than or equal to unity, representing the amount of partial connection on the inlet and outlet side, respectively. For example, 0. 5 - 0. 25 means that 50% of the inlets and 25% of the outlets are connected, and the rest of the terminals are abandoned.
- C_{in} and C_{out} are the pattern connection vectors defined above.

It should be noted that the last factor in the notation, i. e., the connection vectors provide the information given by the other two factors. However these two factors are still written for convenience.

Among all regular patterns, one particular pattern is of interest, the base pattern. This pattern is achieved when the group size of the connected terminals and that of the abandoned terminals are the minimum possible under the stated partial connection amount. For example, given a partial connection amount of 50% on the inlet side of an 8 X 8 network, then the regular patterns C_{in} = [1, 1, 1, 1, 0, 0, 0, 0] and C_{in} = [1, 1, 0, 0, 1, 1, 0, 0] are not base patterns, whereas C_{in} = [1, 0, 1, 0, 1, 0, 1, 0] is. For another example, the two patterns on the two sides of the PC network of Figure 4 are base patterns. Throughout this paper when the third factor (connection sets) of a PC network definition is omitted, the base pattern is assumed.

The dashed curves in Figure 5 show the bandwidth vs. load for an assortment of PC networks, all developed from a 32 X 32 BB network. All the patterns for these networks are base regular patterns. For each network, all the connected inlets have the same load, P, and the load of any given inlet has equal probability of going to any outlet. Thus, if we ignore the abandoned terminals, the PC networks shown in the figure operate under the equiprobable model. The main performance measure in this study is the bandwidth. It should be noted that the bandwidth, BW , used throughout this paper is the normalized bandwidth, defined as the average number of successful paths per cycle divided by the maximum possible such number. Thus, for example, if the average number of successful paths per cycle for a 32/0.5-1 PC network is 10, then BW = $\frac{10}{16}$ .In Figure 5, the bandwidth of the original network, i. e., with 100% connection on both sides, is represented by the dotted (lowermost) curve. This figure reveals a number of interesting observations.

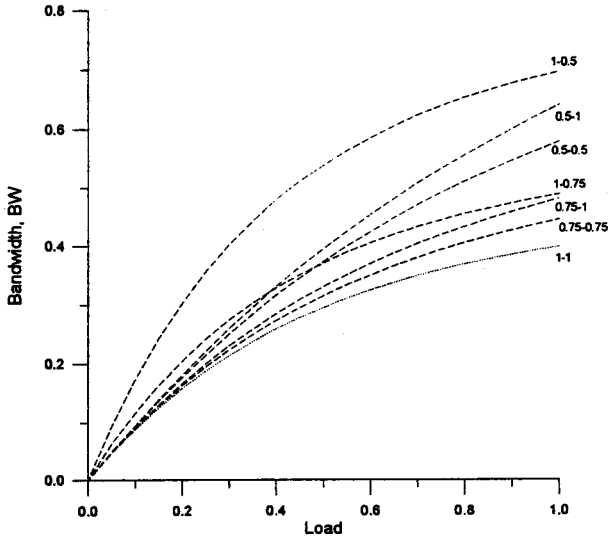

Figure 5 : Bandwidth vs. load for a 32 x 32 BB network fully connected (dotted curve) and partially connected at different amount (dashed curves).

First, we notice that partial connection improves the bandwidth more when implemented on the outlet side. This is clear from the top two curves. Both curves represent the same absolute amount of partial connection, yet the choice of the side to implement the partial connection makes a noticeable difference. The interpretation of this is that blocking is more likely in the

stages closer to the outlets (with some blocking taking place even at the outlets themselves) than in the stages closer to the inlets. By reducing the number of outlets we provide more empty space in the congested stages, thus allowing more paths to be established. On the contrary, since the stages closer to the inlets are less congested any way, partial connection there has a lighter, though still noticeable, effect.

The rather surprising observation is that the bandwidth of a PC network with partial connection on one side only, i. e., 1 - x or x - 1, is higher than that of a PC network with partial connection at the same amount on both sides, i. e., x - x . A lock at the top three curves clarifies this observation. The fact that the bandwidth of the 0.5 - 0.5 PC network is less than that of the 1-0.5 PC network could be interpreted as follows. First, let *P* denote the load of each connected inlet, i. e., the probability that the inlet will receive traffic from its source in each cycle. Now, assuming the same load *P* for the inlets of both networks, the sources of the latter network pour more traffic into network than do the fewer sources of the former. Since the bandwidth is directly increases with the input traffic, as shown in all the bandwidth vs. load curves throughout this paper, the bandwidth of the 0.5 - 0,5 PC network is less than that of the 1-0.5 PC network.

On the other hand, the fact that the bandwidth of the 0.5- 0.5 PC network is less than that of the 0.5-1 PC network could be interpreted as follows. Assuming the same load *P* for both networks, the traffic in the latter is distributed over a large number of outlets. Since the destinations for any source are distributed uniformly, the more destinations there are, the less contention for a given destination. This again results in a higher bandwidth.

It should not be construed from the above argument that the bandwiidth improvement is slight if partial connection is done on both sides of the network. A comparison between the curves of the 0.5- 0.5 and the 1-1 PC networks shows that the bandwidth of the former network at high load is about 45% higher than that of the latter.

Another evidence that it is not just the absolute amount of partial connection that determines the bandwidth improvement for a PC network can be seen from the two curves 1-0.5 and 0.75-0.75. In both cases the absolute amount of connected terminals is 75% of the original number. However the bandwidth of the former network is about 70% higher than that of the latter.

One more interesting observation is that the bandwidth curves "level off" when no partial connection is done on the inlet side. The curves 1-1, 1-0.5 and 1-0.75 show that vividly. The reason is that the heavy traffic caused by connecting all the inlets results in bandwidth saturation when the load starts to get high. The opposite of this observation is also true; the bandwidth curves become more linear when no partial connection is done on the outlet side and a large amount of partial connection is done on the inlet side. This is evident from the curve 0.5-1. The reason here is that the traffic poured into the network is distributed over a large number of destination, resulting in a light traffic density within the network. Thus, any increase in the input load produces a proportional improvemment in the network bandwidth, with no developmment of saturation due to the light traffic intensity within the network.

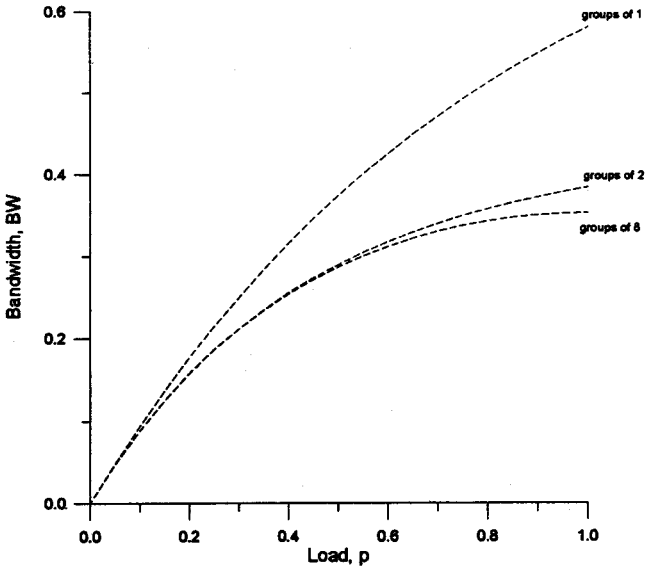

Figure 6 : Bandwidth vs. load for a 32 / 0.5 – 0.5-0.5 PC network with regular connection pattern on both sides, for three different group sizes.

Besides the amount of partial connection and the side on which partial connection employed, the connection pattern has a profound effect on the bandwidth of a PC network. This is clear in Figure 6, which includes three curves representing the bandwidth vs. load for three 32/0.5 - 0.5 PC networks. The patterns on both sides of each network are the same, and are regular with the

group size of the connected terminals being equal to that of the abandoned terminals. The difference between the three networks is the group size, which is one for the top curve, two for the middle curve, and eight for the bottom curve. Thus, the top curve represents the base pattern. It can be seen that as the group size increases the bandwidth decreases. This is logical, since each connected terminal suffers less blocking hence performs better, if the neighbor terminals are either abandoned or lightly active.

4 0.5 - 0.5 PC Networks

Having introduced the approach of partial connection in the previous section, we will focus in this section only on the 0.5 - 0.5 PC networks. What makes this amount special is that it produces native BB network sizes, i. e., 4, 8, 16, etc. To create a PC network with any of these sizes we apply a 0.5 - 0.5 partial connection operation, with base regular pattern, to the immediately higher size. With this technique we can reproduce the entire BB network size family. We then end up with two networks for each size : one with normal bandwidth and one with high bandwidth. We will call the latter a *0.5-0.5* PC network, since it has twice the number of inlets and outlets found in the former. Thus, when we say, for example, a 32 X 32 0.5- 0.5 PC network, or a 0.5 - 0.5 PC network and a BB network of size 32, it should be understood that the 0.5- 0.5 PC network was originally a 64 X 64 BB network then was subjected to a 0.5 - 0.5 partial connection operation.

We now compare the bandwidth of a 0.5 - 0.5 PC network to those of a crossbar and a BB network of the same size. Fixing the size makes the effect of partial connection vividly clear. Figures 7 and 8 below are the same as Figures 2 and 3, respectively, only this time a curve for a 0.5- 0.5 PC network is included.

In Figure 7, we can see that, at all loads, the bandwidth of the 0.5 - 0.5 PC network is higher than that of the BB network and is closer to the bandwidth of the, much more costly, crossbar switch. The more important observation is that the bandwidth of the 0.5 - 0.5 PC network increases as the load increases at a greater rate than does the bandwidth of the BB network.

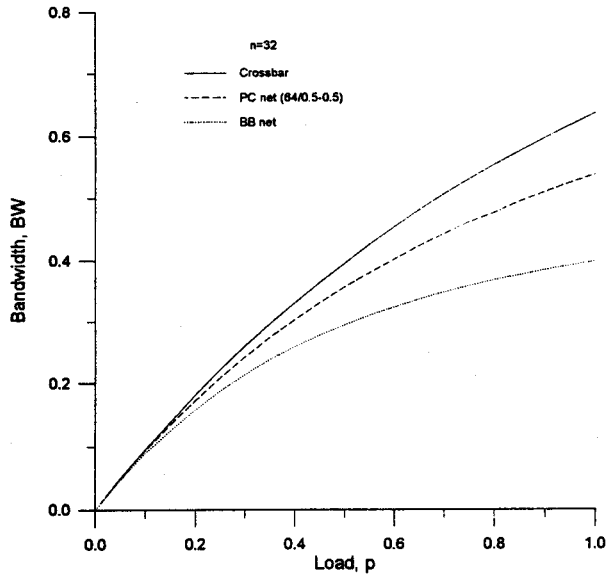

Figure 7 : Bandwidth vs. load for a crossbar, a 0.5-0.5 PC network and a BB network, all of size 32

In Figure 8, on the other hand, we can see that at all sizes the bandwidth of the 0.5- 0.5 PC network is higher than that of the BB network. Again, it is closer to the bandwidth of the, much more costly, crossbar switch. Also, we notice that the bandwidth of the 0.5 - 0.5 PC network decreases as the size increases at a smaller rate than does that of the BB network.

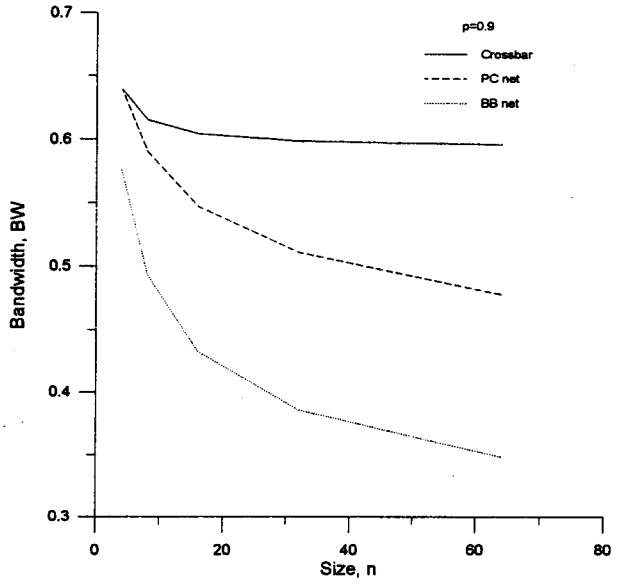

Figure 8 : Bandwidth vs. size for a crossbar, a 0.5-0.5 network , all under the same load, p = 0.9

Figure 9 summarizes the findings of this paper. It shows the bandwidth gain of the 0.5 - 0.5 PC network, defined as the ratio of the bandwidth of the 0.5 - 0.5 PC network and that of a BB network of the same size. Bearing in mind that the cost of a 0.5- 0.5 PC network is

almost twice that of a BB network, the same curves may be looked upon as representing the performance Karice cost curve of 0.5 - 0.5 PC networks. As can be seen, the curves are monotonically increasing. This means that the larger the size, the more costeffective 0.5 - 0.5 PC networks become, a desirable feature.

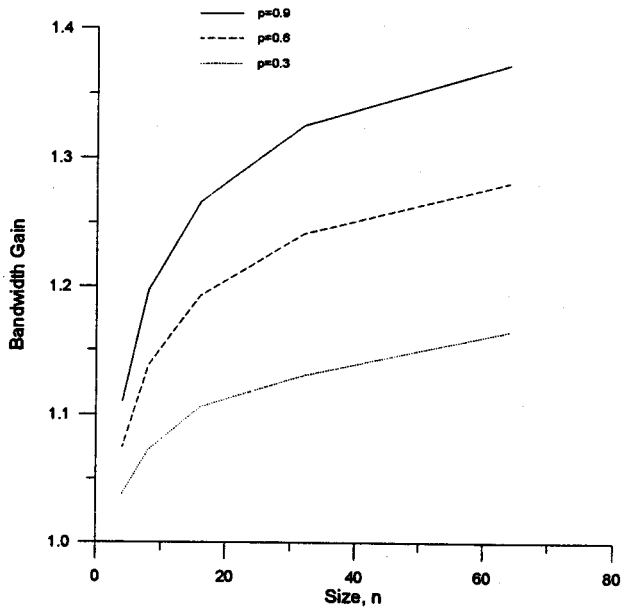

Figure 9 : Bandwidth gain vs. size for a 0.5-0.5 PC network, at three different loads

An interesting observation in Figure 9 is that partial connection is more effective at high loads, another desirable feature. This behavior is logical because at low loads there is little blocking to begin with, and therefore there is a little role for partial connection to play. At high loads, on the other hand, there is more blocking for partial connection to relieve, hence the noticeable bandwidth increases. We note that a peak bandwidth gain of about 40% is obtained at load p = 0.9 for a network of size 64 x 64 .

5 Conclusions

In this paper we have introduced a novel approach to improve the bandwidth of interconnection networks, partial connection. It has been found to be a cost effective solution to the bandwidth problem of the BB network, especially at large sizes and heavy loads. The salient features of this approach is that it is extremely easy to implement, preserves the modularity of the BB network, increases the delay by only one - stage - passing time (i.e., by the ratio $\frac{1}{\log_2 n}$), does not require any specifically designed hardware or software, and does not require any alteration in the design of the network SEs.

This technique has been found to be more effective on the outlet side than on the inlet side. Thus, the technique is well suited to solve the disturbing "hot spot" problem [11]. This problem arises when one outlet is requested more than the others, creating a hot spot in the network . When this outlet is surrounded by two connected outlets, as is the case in normal networks, the problem is exacerbated. The reason is that the traffic going to the surrounding outlets is likely to block the (already heavy) traffic going to the hot spot, making it even 'hotter'. The partial connection approach can be used here as follows. Abandon the two outlets surrounding the hot spot, thus reducing the blocking of the traffic going to the spot.

Although it is theoretically possible to implement partial connection at any amount, practically the amount of 0.5 seems to be a reasonable upper bound. A lower amount than this would necessitate using a BB network other than that of twice the size. This in turn would 1) raise the cost of partial connection to more than double, and 2) introduce delays more than one stage-time. We examined closely the case when partial connection is implemented on both sides at the amount 0.5 .

Finally, although this paper focuses on the BB network, its findings are applicable to all types of banyan network, and even many other interconnection networks.

Appendix

Given P and R for an nxn switching network, crossbar switch or banyan network, one can evaluate the bandwidth, BW, for the network. The course of the evaluation differs according to which one of two operation models is used : the generalized model or the equiprobable model. Below we will show how the bandwidth is evaluated under both models.

The Generalized Model :

It is the generalized model [9] that we use in this paper to analyze the performance of PC networks. In this

model the elements of the load vector P are arbitrary, and so are the elements of the reference matrix R (with the constraint that each row in R should sum to unity). Our aim is to find the bandwidth BW.

Let a stage input or ouput be called active if it carries an outlet access request from a given source (i.e., if it forms an end point to a partially established path). Let us define for each input i of a given stage L an activity vector, P^{l}_{i}, where the jth element of this vector, P^{l}_{ij}, represents the probability that input i will be active due to source j. The probability that the input is active is just the sum of all the elements of this vector. Note that the activity vector for any first stage input i contains only one nonzero element, equal to p_i.

In a similar manner, let us define for each stage output i an activity vector, $\hat{P}^{L}_{i}$, with details like above. Now, if we can find $\hat{P}^{l-1}_{i}$, then we can sum its elements to find the probability that outlet i is active. Then we can find the average number of active outlets, divide it by the maximum possible such number, hence find the bandwidth. The problem now is how to find $\hat{P}^{L-1}_{i}$. This problem can be solved as follows.

First, from the activity vectors of the inputs of the first stage, which are obtained from P as mentioned above, we generate the activity vectors of the outputs of that stage. And from the activity vectors of the outputs of the first stage, we generate the activity vectors of the inputs of the second stage. If we carry out this activity vector generation process, recursively, until the final stage, we end up with the activity vectors for the ouputs of the last stage, $\hat{P}^{L-1}_{i}$.

The problem is now reduced to finding a means to

1. generate the activity vector of some ouput of a given stage, given the activity vectors of the inputs of that stage.
2. generate the activity vector of some input of a given stage, given the activity vectors of the outputs of the previous stage.

The second generation process is easy. After obtaining the activity vector for a given stage output, assign it to the next-stage input connected to this output.

The first generation process is somewhat involved. To illustrate how it is performed, consider a typical switch in some stage. In the four equations below we will use the letter u to denote the word upper and the letter l to denote the word lower (do not confuse this l with that which denotes the stage number). These equations will specify a certain input or output of a switch, but can be adapted easily (by symmetry) to the other input or ouput of that switch.

Suppose we know, say, $\tilde{P}_{l_k}$ the probability that the lower input is active due to source k, then we can find, say, $\hat{P}_{u_k}$, the probability that the upper ouput is active due to this source, as.

$$\hat{P}_{u_k} = \hat{P}_{u,\ell} \frac{\tilde{P}_{l_k}}{\tilde{P}_{l}},$$

where $\tilde{P}_l$ is the probability that the lower input is active, and $\hat{P}_{u,\ell}$ is the probability that the upper output will be active due to a connection from the lower input. The former probability is just.

$$\tilde{P}_{\ell} = \sum_{k} \tilde{P}_{\ell_k}$$

where the summation is run over all the sources having access to the lower input. The latter probability is just.

$$\hat{P}_{u,\ell} = p_{u\_u}(1 - p_{\ell\_u}) + 0.5\, p_{u\_u}\, p_{u\_\ell}$$

where P_{u_ℓ} is the probability that the upper input attempts a connection to the lower output.

This probability is obtained as

$$P_{u\_\ell} = \sum_{i \in \tilde{a}_u} \tilde{P}_{\ell} \frac{\sum_{j \in \hat{b}_l} r_{ij}}{\sum_{j \in B} r_{ij}},$$

where $\tilde{a}_u$ is the set of all sources having access to the upper input, and P_{u_i} is the probability that that upper input becomes active due to source i, and $\tilde{b}_{\ell}$ is the set of all outlets accessible from the lower output, r_{ij} is the probability that surce i requires connection to outlet j, and $\hat{B}$ is the set of all outlets accessible form the two switch outputs.

Now, we state the above procedure algorithmically, Let ℓ denotes the stage number, with $\ell = 0$ for the stage to which the inlets are connected and $\ell = L - 1$, for the stage to which the outlets are connected, where $L = \log n$ is the number of network stages. Additionally, let.

- $\tilde{a}^{\ell}_{i}$ = set of all inlets having access to input i of stage ℓ.
- $\hat{a}^{\ell}_{i}$ = set of all inlets having access to ouput i of stage ℓ. Clearly, $\hat{a}^{\ell}_{i} = \tilde{a}^{\ell}_{2\lfloor i/2 \rfloor} \cup \tilde{a}^{\ell}_{2\lfloor i/2 \rfloor + 1}$

- $\hat{b}_i^{\ell}$ = set of all outlets accessible from output i of stage ℓ.
- $\hat{B}_i^{\ell}$ = set of all outlets accessible from the two outputs of the switch of which i is an ouput. Clearly, $\hat{B}_i^{\ell} = \hat{b}_{2\lfloor i/2 \rfloor}^{\ell} \cup \hat{b}_{2\lfloor i/2 \rfloor + 1}^{\ell}$.
- $\hat{P}_i^{\ell}$ = probability that output i of stage ℓ is active. Clearly $\hat{p}_i^{L-1} = \hat{p}_i$, for all i.
- $\hat{P}_{i-j}^{\ell}$ = probability that input i, of stage ℓ tries to connect to output j of the same stage. Note that $P_{i-j}^{\ell} = 0$ for i and j such that $2\lfloor i/2 \rfloor \neq 2 \lfloor j/2 \rfloor$.
- $\tilde{P}_{ij}^{\ell}$ = probability that input i, of stage ℓ, becomes busy active due to a connection to inlet j. Clearly, $\tilde{p}_i^{\ell} = \sum_{j \in \tilde{a}_i^{\ell}} \tilde{p}_{ij}^{\ell}$, for all i.
- $\hat{P}_{ij}^{\ell}$ = probability that output i, of stage ℓ, becomes active due to a connection to inlet j. Note that $\hat{p}_{ij} = 0$ for $j \notin \hat{a}_i^{\ell}$. Clearly, $\hat{p}_i^{\ell} = \sum_{j \in \hat{a}_i^{\ell}} P_{ij}^{\ell}$, for all i.
- $\hat{P}_{i,j}^{\ell}$ = probability that output i, of stage ℓ, is connected to input j of that stage. Clearly, $\hat{p}_i^{\ell} = \sum_{j \in \hat{a}_i} p_{i,j}^{\ell}$, for all ℓ and i.

Now the procedure is as follows

procedure Bandwidth - Generalized (P, R, n, L)

begin
for $i \leftarrow 0$ to n – 1 do { initialize vectors }
if i = j then $\tilde{P}_{ij}^{0} \leftarrow P_i$

else $\tilde{p}_{ij}^{0} \leftarrow 0$

$\tilde{a}_i^{0} \leftarrow i$.

for $l \leftarrow 0$ to L –1 do {for each stage of the network}

for i $\leftarrow$ 0 to n – 1 do {for each input or output, as appropriate, of current stage } { build the sets a_i^{ℓ}, b_i^{ℓ}, and $\hat{B}_i^{\ell}$ }

$\hat{a}_i^{\ell} \leftarrow \tilde{a}_{2\lfloor i/2 \rfloor}^{\ell} \cup \tilde{a}_{2\lfloor i/2 \rfloor + 1}^{\ell}$

if i mod 2 = 0 then $\hat{b}_i^{\ell} \leftarrow \{\frac{\lfloor i/2 \rfloor n}{2^{\ell}} \bmod n, \frac{\lfloor i/2 \rfloor n}{2^{\ell}} \bmod n + 1, \ldots, \frac{\lfloor i/2 \rfloor n}{2^{\ell}} \bmod n + \frac{n}{2^{l+1}} - 1 \}$

else $\hat{b}_i^{\ell} \leftarrow \{\frac{\lfloor i/2 \rfloor n}{2^{\ell}} \bmod n + \frac{n}{2^{\ell+1}}, \frac{\lfloor i/2 \rfloor n}{2^{\ell}} \bmod n + \frac{n}{2^{\ell+1}} + 1, \ldots, \frac{\lfloor i/2 \rfloor n}{2^{\ell}} \bmod n + \frac{n}{2^{\ell}} - 1 \}$

$\hat{B}_i^{\ell} \leftarrow \tilde{b}_{2\lfloor i/2 \rfloor}^{\ell} \cup \tilde{b}_{2\lfloor i/2 \rfloor + 1}^{\ell}$

$j \leftarrow 2\lfloor i/2 \rfloor$ {point at upper input or output as appropriate, of current switch} {now calculate probabilities}

$P_{i-j}^{\ell} \leftarrow \sum_{z \in a_i^{\ell}} \hat{P}_{iz}^{\ell} \frac{\sum_{y \in \hat{b}_i^{\ell}} r_{xy}}{\sum_{j \in \hat{B}_j^{1}} p_{az}}$,

$P_{i-j+1}^{\ell} \leftarrow 1 - p_{i-j}^{\ell}$

$\hat{p}_{ij}^{\ell} \leftarrow {}_{i-j}^{\ell} (1 - p_{j+1-i}^{\ell}) + 0.5\, p_{j-i}^{\ell}\, p_{j+1-i}^{\ell}$

$\hat{p}_{ij+1}^{\ell} \leftarrow p_{j+1-i}^{\ell}\, q_{j-1}^{\ell} + .5\, p_{j-i}^{\ell}\, p_{j+1-i}^{\ell}$

for each $k \in \tilde{a}_i^{\ell}$ do {build half the activity vector of output i}

$\tilde{p}_{ik}^{\ell} \leftarrow \hat{p}_{i,j}^{\ell} \frac{\tilde{p}_{ik}^{\ell}}{\hat{p}_j^{\ell}}$

for each $k \in \tilde{a}_{j+1}^{\ell}$ do {build other half of the activity vector of output i}

$\hat{p}_{ik}^{\ell} \leftarrow \hat{p}_{i,j+1}^{\ell} \frac{\hat{p}_{j+1k}^{\ell}}{\hat{p}_{j+1}^{\ell}}$

while $l < L-1$ do {export vectors of current output to connected input of next stage} $k \leftarrow (2i + \lfloor \frac{2i}{n} \rfloor) \bmod n$ {first find that input using perfect shuffle function}

$\tilde{p}^{\ell+1} \leftarrow \hat{p}_{ij}^{\ell}$ {then export the activity vector to next stage}

$\tilde{a}_{kj}^{\ell+1} \leftarrow a_{ij}^{\ell}$ {and also export the set of accessible inlets}

$\hat{p}_i^{L-1} \leftarrow \sum_{j \in \hat{a}_i^{L-1}} p_{ij}^{L-1}$ {find probability that outlet i is active}

end

Now, using the probabilities $\hat{P}_i^{L-1}$ we can find the probability P (x) that x outlets, $0 \le x \le n$, is active. This is done by considering all the $\binom{n}{x}$ combinations in which x outlets are active. From this we can find the average of x, which when divided by the maximum number possible of active outlets gives the bandwidth BW . That, is,

$$BW = \frac{\sum_x x\, p(x)}{\min(x_{in}, x_{out})\, n}$$

where x_{in}, x_{out} are the fraction of terminals connected on the inlet and outlet sides respectively.

The Equiprobable Model

In the special case when all the sources are

statistically identical, and all the destinations are equally likely to be referenced by an active source, the calculation of the bandwidth is greatly simplified [10]. In this mode, called the equiprobable model, the elements of the load vector are identical, and so are the elements of the reference matrix. That is, $p_0 = p_1 = \ldots = p_{n-1} = p$ and $r_{00} = r_{01} = \ldots = r_{n-1,n-1} = 1/n$.

Under this model the bandwidth of an interconnection network is just the probability that an outlet is active. For an n x n crossbar, the bandwidth is obtained by the expression

$$BW = 1 - (1 - \frac{p}{n})^n$$

The reasoning for this expression is as follows. The probability that an outlet is referenced by a given inlet is $\frac{p}{n}$. Thus the probability that the outlet is not referenced by any inlet is $(1 - \frac{p}{n})^n$. The complement of this probability represents the probability that the outlet is referenced by at least one inlet, which is the probability that the outlet will be active, i.e. the bandwidth.

For an n x n BB network, we find the bandwidth as follows. First, we find the probability that the output of a typical SE in the first stage is active, using the crossbar expression. This probability is then considered as the probability that an inlet to the second stage is active. By applying the crossbar expression again, we can find the probability that an outlet of the second stage is active, hence an inlet to the third stage is active. Repeating this procedure recursively, we can find the probability that a network outlet is active, i.e. the bandwidth.

To state the above procedure algorithmically, let $\tilde{p}^{\ell}$ denote the probability that an inlet to stage ℓ is active, and $\hat{p}^{\ell}$ the probability that an outlet of stage ℓ is active.

procedure Bandwidth - Equiprobablel (p, L)

begin
{ initialization }

$p^0 \leftarrow p$

for $\ell \leftarrow 0$ to L – 2 do { for each stage, except the last } .

$\hat{p}^{\ell} \leftarrow 1 - (1 - \frac{\tilde{p}^{\ell}}{n})^2$

$\tilde{p}^{\ell+1} \leftarrow \hat{p}^{\ell}$ {export output probabilities to be input to next stage} .

end

Now,

$$BW = 1 - (1 - \frac{\hat{p}^{L-1}}{2})^2.$$

REFERENCES

[1] **Nassar, H. and J. Carpinelli,** 1993. "A Simple Fault-Tolerance Technique for Interconnection Networks," Proc. of the sixth Int'l Conf. on Parallel and Distributed Computing Systems, ISCA, 25 - 30.

[2] **Bhuyan, L., Q. Yang,** and **D. Agrawal,** 1989. "Performance of Multiprocessor Interconnection Networks, IEEE Computer, 25 - 37 .

[3] **Wu, C.** and **T. Feng,** 1980. "On a Class of Multistage Interconnection Networks," IEEE Trans. Computers, 694 - 702.

[4] **Goke, L. and G Lipovski,** 1973. "Banyan networks for partitioning multiprocessor systems," Proc. 1st Annu. Symp. Computer Architecture, 21 - 28.

[5] **Ahmadi, H.** and **W. Denzel,** 1989. "A Survey of Modern High-Performance Switching Techniques," IEEE J. Selected Areas on Communications., 1091 - 1103.

[6] **Batcher, K.,** 1968. "Sorting Networks and Their Application," Proc. Spring Joint Computer Conf., AFIPS, 307 - 314 .

[7] **Kruskal, C.** and **M. Snir,** 1983. "The Performance of Multistage Interconnection Networks for Multiprocessors," IEEE Trans. Computers, 1091 - 1098.

[8] **Mun, Y.** and **H. Youn,** 1994. "Performance Analysis of Finite Buffered Multistage Interconnection Networks," IEEE Trans. Computers, 153 - 162.

[9] **Nassar, H.** and **J. Carpinelli,** "Performance of Multiprocessor Interconnection Networks with Arabitrary Access Conditions," Submitted for publication.

[10] **Patel, J.,** 1981."Performance of processor - memory interconnections for multiprocessors," IEEE Trans. Computers, 771 - 780 .

[11] **Gyungho, L, C. Kruskal,** and **D. Kuck,** 1994. "On the Effectiveness of Combining in Resolving 'Hot Spot' Contention," J. of Paralled and Distributed Computing 136 - 144 .