A GENERAL INTERCONNECTION NETWORK WITH THE CONSIDERATION OF LOCALITY IN TRAFFIC

Shun-Shii Lin and Ferng-Ching Lin

Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan, R.O.C.

ABSTRACT

We propose a new general interconnection network with the consideration of locality in traffic. This network has $\log^2$N-logN maximum intercell delay but when high locality occurs in the communications, the mean intercell delay drastically decreases to O(1). The problem of how to map processors with a known traffic distribution onto the terminals of the network in order to minimize the mean intercell delay is analyzed and formulated to be a quadratic assignment problem. The usages of this network as a partitioner, a permuter, a full switch and a generalized connection network are explained.

1. Introduction

Interconnection networks are systems which provide a means for simultaneous connections among a set of terminals. A *permutation network* performs one-to-one mappings between two disjoint set of terminals. The Benes network [1] can perform all the permutations between its inputs and outputs with 2logN-1 intercell delay. A *generalized connection network* (GCN) maps any subset of its N input terminals to any subset of its N output terminals. An input may be connected to many outputs, but not the reverse. Thompson [7] showed a GCN with 7.6NlogN crosspoints and 3.8logN delay. Nassimi and Sahni [5] suggested a class of GCNs having O(klogN) intercell delay and $O(kN^{1+1/k}logN)$ switches for $1 \le k \le logN$.

A *full switch* performs one-to-one connections between any subset of N/2 terminals and its complement set, no distinction being made between inputs and outputs. In [2], Chung and Wong presented a full switch with (NlogN-N)/2 cells and 2logN-1 delay. A *partitioning network* partitions the set of terminals into arbitrary disjoint subsets such that the terminals within each subset can communicate with each other. Chung and Wong [2] presented a partitioning network which has NlogN + N - logN - 1 cells and 6logN - 4 intercell delay. In [7], Thompson suggested the use of GCNs as partitioning networks. Both of these two designs are not modular in the sense that the larger networks can not be constructed recursively from the smaller ones, therefore their expandability is limited.

All the previous interconnection networks only focused on the total count of cells and the maximum intercell delay. They ignored the principle of communication locality which states that both the cost and performance metrics of VLSI favor architectures in which communication is localized. Recently, in [6], Tan proposed an MIMD packet switching network taking advantage of communication locality. It is reasonable to assume in many cases that efficient and practical use of interconnection network should exhibit much greater traffic over short distances than over long communication paths.

With such consideration of locality in traffic, we present in this paper a new general-purpose network, called localizable interconnection network (LIN). Its structure is essentially a complete binary tree. However, unlike a normal tree, LIN gets thicker further from the leaves. Terminals are located at the leaves. Communication between any pair of processors is basically easy because it has a unique path going up the tree to their least common ancestor and then back down. Thus

communications can be routed locally without going higher up in the tree. The network's maximun intercell delay is $\log^2 N - N$ and its mean intercell delay decreases with higher locality of communication. For a uniform traffic distribution, its mean intercell delay is $(N\log^2 N - 3N\log N + 4N - 3)/(N - 1)$. For a half-geometric traffic distribution, its mean intercell delay is about one third of the uniform case. Most important, for a quarter-geometric traffic distribution, the mean intercell delay drastically decreases to $4.5 + (0.5 - \log^2 N - 3\log N) / (N - 1)$, a constant order. Other types of traffic distributions, harmonic and bounded, are also analyzed later in this paper.

2. The New Interconnection Network

Fig.1 shows the recursive construction of an N-LIN, where $N=2^n$ is the number of terminals (termed $T_0, ..., T_{N-1}$), each can be attached to a processor or a resource module. An N-LIN consists of two (N/2)-LINs, one (N/2)-permuter having N/2 inputs (termed $I_0, ..., I_{N/2-1}$), N/2 outputs (termed $O_0, ..., O_{N/2-1}$) and N/2 middle lines (termed $M_0, ..., M_{N/2-1}$), and N/2 switches $S_{N/2}, ..., S_{N-1}$ to connect or disconnect the links between $O_0, ..., O_{N/2-1}$ and $B_{N/2}, ..., B_{N-1}$. $B_0, ..., B_{N-1}$ are used as internal buses for connecting terminals and for constructing larger LINs.

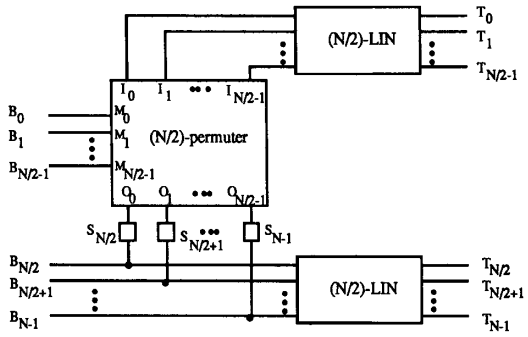

Fig.1 The construction of N-LIN

The (N/2)-permuter can be realized by the famous Benes network with 2logN - 3 stages and (N/2)log(N/2) - N/2 + 1 elementary two-state cells, as depicted in Fig.2. The N/2 lines $M_0, ..., M_{N/2-1}$ right below the stage logN-2 are linked outside to $B_0, ..., B_{N/2-1}$ in a sequence. As the construction base, a 2-LIN is formed by only one switch. The total number of cells for our N-LIN is $O(N\log^2 N)$.

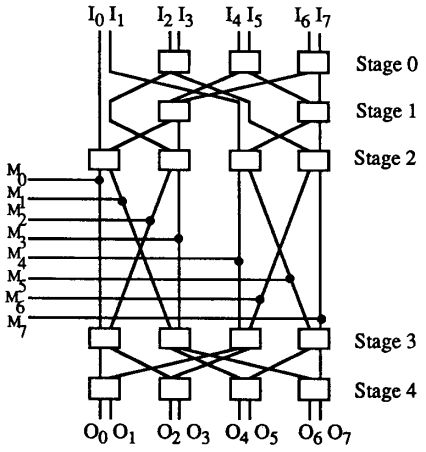

Fig.2 An 8-permuter

3. Consideration of Locality in Traffic

The *maximum intercell delay* D(N) of the N-LIN is the maximum number of cells that must be traversed to transmit a message between any two connected terminals. $D(1) = 0$, $D(2) = 1$, and $D(N) = \log^2 N - \log N$ for $N \geq 4$. The *mean intercell delay* M(N) of the N-LIN is the expected number of cells that must be traversed to transmit a message between any two connected terminals. Mean intercell delay is a better measure of message delay than the maximum intercell delay. It depends on the message traffic distribution of communications among the different terminals in the network. Because our N-LIN is a binary tree structure, any two terminals can communicate with each other through a path inside a subtree with 2^t terminals for some t, $1 \leq t \leq \log N$. If $\Phi(t)$ is the probability of interprocessor communications crossing $D(2^t)$ cells, then $M(N) = \sum_{1 \leq t \leq \log N} \Phi(t)\, D(2^t)$, where $\sum_{1 \leq t \leq \log N} \Phi(t) = 1$.

Clearly, different degrees of communication locality lead

to different $\Phi(t)$ and, in turn, different mean intercell delays. To get the quantitative insight of how communication locality affects the mean intercell delay of the network, we will look at some typical traffic distributions in the following subsections.

A. Uniform traffic distribution

For any terminal in the tree-structured N-LIN, the number of terminals it can communicate with by using $D(2^t)$ cells is 2^{t-1}. A message traffic distribution is *uniform* if the probabilities that a terminal communicates with other terminals are all equal. For this distribution, we have $\Phi(t) = 2^{t-1}/(N - 1)$, $1 \le t \le \log N$, and $M(N) = \sum_{1\le t\le \log N} \Phi(t)\, D(2^t) = 1/(N - 1) + \sum_{2\le t\le \log N} (t^2 - t)\, 2^{t-1}/(N - 1) = (N\log^2 N - 3N\log N + 4N - 3) / (N - 1) = O(\log^2 N)$.

B. Geometric traffic distribution

We call a message traffic distribution *half-geometric* if the probability that a terminal communicates with others by using D(2) cells is 1/logN, and communicates with others by using $D(2^t)$ cells is $1/(2^{t-1}\log N)$, $2 \le t \le \log N$. This results a half decreasing effect and $\Phi(t) = 1/\log N$ for $1 \le t \le \log N$. Thus $M(N) = \sum_{1\le t\le \log N} \Phi(t)\, D(2^t) = 1/\log N + \sum_{2\le t\le \log N} (t^2 - t) / \log N = (\log^2 N)/3 + 1/\log N - 2/3$.

A traffic distribution is *quarter-geometric* if the probability that a terminal communicates with others by using D(2) cells is $c_q = N/(2N-2)$, and communicates with others by using $D(2^t)$ cells is $c_q/2^{2(t-1)}$ for $2 \le t \le \log N$. It can be calculated that $\Phi(t) = c_q/2^{t-1}$ for $1 \le t \le \log N$. In this case, $M(N) = \sum_{1\le t\le \log N} \Phi(t)\, D(2^t) = 4.5 + (0.5 - \log^2 N - 3\log N) / (N - 1) = O(1)$.

C. Harmonic distributions

The following sum is the well-known *harmonic number*: $H_n = 1 + 1/2 + 1/3 + \ldots + 1/n = O(\log n)$. We call a traffic distribution *harmonic* if the probability that a terminal communicates with others by using $D(2^t)$ cells is $1/(t \cdot H_n)$, $1 \le t \le n$. This results a harmonic decreasing effect and $\Phi(t) = 1/(t \cdot H_n)$ for $1 \le t \le n$. Therefore, $M(N) = \sum_{1\le t\le n} \Phi(t)\, D(2^t) = O(\log^2 N/\log\log N)$.

D. Bounded traffic distributions

A traffic distribution is *b-bounded*, $1 \le b \le \log N$, if $\Phi(t) = 0$ for $b+1 \le t \le \log N$. Under such a distribution, each terminal only communicates with terminals within the "distance" of $D(2^b)$ cells. As for the extreme 1-bounded case, each terminal only communicates with one other terminal and M(N)=D(2)=1. When b is greater than 1, M(N) depends on the traffic distribution $\Phi(t)$, $1 \le t \le b$. For b-bounded uniform distribution, $M(N) = O(b^2)$. For b-bounded quarter-geometric distribution, $M(N) = O(1)$.

4. The Processor Mapping Problem

In real applications, the traffic distributions may not be exactly what we have described. Usually, $\Phi(t)$ can not be expressed by a simple formula, it must be derived from some experiment or simulation results. In this situation, what we can do is to map the processors with prior knowledge of their communication probabilities onto the terminals of the N-LIN such that M(N) is minimized.

If two terminals have a common parent in the tree, we say that they have genealogical distance 1. If they have a common grandparent but distinct parents, we say their genealogical distance is 2, and so on. Suppose the probabilities q_{ij} of interprocessor communication from P_i to P_j, $0 \le i \le N-1$, $0 \le j \le N-1$, have been known through prior knowledge, where $\sum_{0\le i\le N-1} q_{ij} = 1$ and $\sum_{0\le j\le N-1} q_{ij} = 1$. Processor pair P_i, P_j will be attached to the terminals $T_{\pi(i)}$ and $T_{\pi(j)}$ with genealogical distance $gen(\pi(i), \pi(j))$. The intercell delay between $T_{\pi(i)}$ and $T_{\pi(j)}$ is $D(2^{gen(\pi(i), \pi(j))})$ since they must communicate with each other through a path inside a subtree with $2^{gen(\pi(i), \pi(j))}$ terminals. The *processor mapping problem* is to find a permutation π such that

$$M(N) = (1/N^2) \sum_{0\le i\le N-1} \sum_{0\le j\le N-1} q_{ij} \cdot D(2^{gen(\pi(i), \pi(j))})$$

is minimized. This problem is indeed the classical *quadratic assignment problem* that was shown to be NP-complete [3] and many methods [4][8] are applicable to obtain approximation

solutions.

5. LIN Used as Partitioner

A partitioning network can connect together all terminals belonging to the same subset in a terminal partition. Typical applications exist in computer systems where N resources are connected to the network and sometimes have to be regrouped into disjoint subsystems to provide separate and private communications.

Suppose the N resources are connected to the terminals $T_0, T_1, ..., T_{N-1}$ of N-LIN through the processor mapping discussed earlier and the desired partition is represented by:

$$\{ U_{1,1}, U_{1,2}, ..., U_{1,i_1}, L_{1,1}, L_{1,2}, ..., L_{1,j_1} \},$$
$$\vdots$$
$$\{ U_{g,1}, U_{g,2}, ..., U_{g,i_g}, L_{g,1}, L_{g,2}, ..., L_{g,j_g} \},$$
$$\{ U_{g+1,1}, U_{g+1,2}, ..., U_{g+1,i_{g+1}} \},$$
$$\vdots$$
$$\{ U_{g+r,1}, U_{g+r,2}, ..., U_{g+r,i_{g+r}} \},$$
$$\{ L_{g+1,1}, L_{g+1,2}, ..., L_{g+1,j_{g+1}} \},$$
$$\vdots$$
$$\{ L_{g+s,1}, L_{g+s,2}, ..., L_{g+s,j_{g+s}} \},$$

where $U_{r,i}$($L_{r,i}$) denotes the i-th terminal in the upper (lower) (N/2)-LIN of subset r in the partition, i.e. $U_{r,i} \in \{T_0, T_1, ..., T_{N/2-1}\}$ and $L_{r,i} \in \{T_{N/2}, T_{N/2+1}, ..., T_{N-1}\}$. The setup algorithm, which takes $O(N\log^2 N)$ time, consists of the following steps:

Step 1: We recursively realize the following sub-partition

$$\{ U_{1,1}, U_{1,2}, ..., U_{1,i_1} \},$$
$$\vdots$$
$$\{ U_{g,1}, U_{g,2}, ..., U_{g,i_g} \},$$
$$\{ U_{g+1,1}, U_{g+1,2}, ..., U_{g+1,i_{g+1}} \},$$
$$\vdots$$
$$\{ U_{g+r,1}, U_{g+r,2}, ..., U_{g+r,i_{g+r}} \}$$

in the upper (N/2)-LIN. It has g+r subsets of terminals being respectively connected to the inputs $I_{h_1}, ..., I_{h_{g+r}}$ of the (N/2)-permuter, where $0 \le h_i < N/2$ for $1 \le i \le g+r$.

Step 2: Similarly, the following sub-partition

$$\{ L_{1,1}, L_{1,2}, ..., L_{1,j_1} \},$$
$$\vdots$$
$$\{ L_{g,1}, L_{g,2}, ..., L_{g,j_g} \},$$
$$\{ L_{g+1,1}, L_{g+1,2}, ..., L_{g+1,j_{g+1}} \},$$
$$\vdots$$
$$\{ L_{g+s,1}, L_{g+s,2}, ..., L_{g+s,j_{g+s}} \}$$

is realized in the lower (N/2)-LIN. It has g+s subsets of terminals being respectively connected to the internal buses $B_{f_1}, ..., B_{f_{g+s}}$, where $N/2 \le f_i < N-1$ for $1 \le i \le g+s$.

Step 3: Set the switches $S_{f_1}, ..., S_{f_g}$ in "connect" state and the rest in "disconnect" state.

Step 4: Now, in the (N/2)-permuter, let the input ports $I_{h_1}, ..., I_{h_g}$ be connected to the output ports $O_{f_1}, ..., O_{f_g}$ respectively. The rest ports can be connected in arbitrary order. This can be done by using the famous "looping algorithm". The result is that g+r subsets of terminals are respectively connected to some middle lines $M_{v_1}, ..., M_{v_{g+r}}$. At last, the N-LIN has g+r+s subsets of terminals being respectively connected to the internal buses $B_{v_1}, ..., B_{v_{g+r}}, B_{f_{g+1}}, ..., B_{f_{g+s}}$.

6. LIN Used as Full Switch, Permuter and GCN

A full switch is capable of performing one-to-one connections, no distinction being made between inputs and outputs. Thus a full switch is a special case of partitioning networks, but it is still more general in its interconnection capabilities than a permutation network. A practical application in real world is the switching system of telephone calls, where each telephone user can phone any other one. However, people would more often phone their relatives or friends close in space (local calls) than those far from them (long distance calls). In this kind of systems, very high locality occurs in the communications. Thus LIN can be used to reduce the mean intercell delay between users.

A permutation network performs one-to-one connections

between two disjoint equal-sized subsets of terminals. Typical applications are in parallel processing systems, such as the processor-to-memory organization. A processor is likely to access a particular favorite memory most of the time except when an interprocessor communication is needed. In such case, we can place those favorite processor-memory pairs close together in our LIN by using D(2) intercell delay.

A generalized connection network can connect any terminal of its input set to any subset of its output set. An application of LIN is to use it in a multiple SIMD machine which is a parallel processing system. A general multiple SIMD system consists of P processors, Q control units, and an interconnection network, where $Q \leq P$. If two processors are assigned to different control units, they are no longer following the same instruction stream and will work independently. We can attach P processors and Q control units to the terminals of LIN. Each of the control units can use the LIN to broadcast instructions to their PEs accordingly.

7. Conclusion

We have presented a tree-structured interconnection network with the consideration of traffic locality. Other salient features of LIN are the following:

1) Easy extension. The tree structure of LIN is both modular and extensible. The number of processors may easily be expanded in an incremental way. This property makes LIN particularly attractive for the implementation of multiprocessor networks of the future.

2) Simple routing. The regular structure of LIN allows simple routing algorithm, which runs in $O(N\log^2 N)$ time on a single processor computer. We have also developed a parallel setup algorithm, which is omitted here, to determine the switch settings for the N-LIN in $O(\log^3 N)$ time.

3) Universality. Our network is very general in the sense that it can be used to simulate many dynamic switching networks (such as partitioner, full, permuter and GCN) as well as many static networks (such as n-cube, mesh-connected network, and so on). Furthermore, it is also possible to modify LIN to act as a package switching network. This remains an interesting research issue.

References

[1] V. E. Benes, Mathematical Theory of Connecting Networks and Telephone Traffic, Academic Press, 1965.

[2] K. M. Chung and C. K. Wong, "Asymptotically optimal interconnection network from two-state cells," IEEE Trans. Comput., vol.C-28, No.7, pp.500-505, July 1979.

[3] M. Hanan and J. K. Kurtzberg, "Placement techniques," in Design Automation of Digital Systems, vol.1, Edited by M. A. Breuer, Prentice Hall, pp.211-282, 1972.

[4] A. Iosupovici, C. King and M. A. Breuer, "A module interchange placement machine," in Proc. 20th Design Automation Conference, pp.171-174, 1983.

[5] D. Nassimi and S. Sahni, "Parallel permutation and sorting algorithms and a new generalized connection network," J. ACM, v.29, pp.642-677, July 1982.

[6] Hsio-Nan Tan, "RSESS interconnection network," In Proc. 1985 Int'l Conf. Parallel Processing, pp.466-473, 1985.

[7] C. D. Thompson, "Generalized connection networks for parallel processor intercommunication," IEEE Trans. Comput., vol.C-27, No.12, pp.1119-1125, Dec. 1978.

[8] K. Ueda, T. Komatsubara and T. Hosaka, "A parallel processing approach for logic module placement," IEEE Trans. Computer-Aided-Design., vol.CAD-2, No.1, pp.39-47, Jan. 1983.