Spectrum Analysis and Min-Cut Transformation of Communication Networks in Parallel Computers

Z. George Mou
Department of Computer Science and
National Center for Complex Systems
Brandeis University, MA, USA
mou@cs.brandeis.edu

Abstract

We present a formal model for the analysis of communication networks in parallel computers. Unlike most others, our model focus on the transmission delays as opposed to the propagation delays of communication patterns. The model allows all symmetric communication networks to be examined by their spectrums and characterized by their transmission dimensions. A min-cut transformation is introduced as a tool for the spectrum analysis, which reduces any symmetric network to the same canonical form. Parallel architectures with different topologies can then be easily compared and evaluated.

1 Introduction

A communication network in a parallel computer supports inter-processor communications. The characteristics of the communication network are key factors to the overall performance of the parallel machine. The question is then how the network should be designed. Alternatively, we may ask how a given communication network can be evaluated and analyzed since a systematic analysis of given networks can be used to guide the design of the networks and help in making the right choices.

Communication networks have been previously characterized by their diameter, connectivity, blockingness, etc. In the case of meshes, including hypercube as a special case, they are also characterized by their dimensions. All these characterizations depend on the abstract topology of the network, and are independent of the bandwidth of the links. These characterizations can be used to answer questions related to *propagation delays* (or *latency*), which is a function of distance, but contains little information about *transmission delays*, which is a function of bandwidth. In fact, much of the previous work on the mappings and analysis of communication patterns was aimed at minimizing the propagation delay, which may or may not result in overall good performance.

There has been a trend of moving from the design of a large number of small processors (LS) to the design of a small number of large processors (SL) in recent years. Broadly speaking, an LS architecture has thousands or more processors where each processor has less than one Mbytes memory and less than 10 MFLOPS processing power. In contract, an SL architecture usually has less than one thousand processors, but each processors has more than one Mbytes of memory and more than 10 MFLOPS of processing power. Examples of LS machines include the early parallel machine models such as TMC's CM-2, MasPar's MP, and WaveTracer's DTC. Examples of SL machines include most of the more recent parallel machines such as TMC CM-5, Intel's Paragon, Meiko's CS-2, KSR's KSR-1, IBM's SP, SGI's Onyx.

It can be argued that the trend from LS to SL implies that the propagation based analysis is increasingly losing its relevance to the performance. On one hand, the smaller number of processors means the largest distance for a message to travel cannot be that large. On the other hand, the larger size of processors means from each processor a much larger amount of messages is likely to be send in and out in each unit of time. And therefore, it is the transmission delay, and not the propagation delay, that is much more likely to be the major cost of communications on SL machines which are currently dominating the MPP market *.

The objective of this paper is to develop a formal framework for transmission delay analysis of communication networks. This framework allows a network to be characterized by its *spectrum* and its *transmission dimension*. The spectrum of the network tells the effective bandwidths of the network for communication patterns of different *frequencies*, where frequency reflects the locality of a given communication pattern. The transmission dimension of a network is derived from the spectrum, and captures the overall performance of the network for a range of communications with different degrees of localities.

The transmission dimension we introduce and the conventional notion of network dimension (which can be referred to as *propagation dimension*) are related but different. First of all, transmission dimension is defined over all symmetric networks (Sec. 3) as opposed to over only networks in the mesh family. Secondly, transmission dimension generally takes real as opposed to integral values. Finally, transmission dimension is affected by changing the bandwidth of each communication link even if the abstract topology stays the same. The two dimensions are related by the fact that a k dimensional regular mesh (mesh with the same size along all dimensions) with unit bandwidth has a transmission dimensionality of k as well.

As a tool for the spectrum analysis, we introduce the *min-cut* transformation of symmetric networks. The min-cut transformation transforms any symmetric network into a logarithmic sequence of numbers. The min-cut transformation of a network can be interpreted as a *canonical network*. Since the canonical network for all symmetric networks have the same abstract topology, the min-cut transformation offers a way of comparing topologically distinct networks. The spectrum and the dimensionality of the network can then be calculated from the min-cut transformation.

This work is in part motivated by the fact that previous work on communication networks have mostly focused on the study of propagation delay of communications [8, 7, 6, 4, 5, 12, 11, 10, 9]. The problem has been recognized by Culler *et. al.*, and the bandwidth has entered in their LogP model as a parameter of the machine (in reciprocal form) [2]. The bandwidth in the LogP model however is assumed to be constant and independent of the communication pattern. This assumption cannot be justified in reality. In contrast, the dependency of effective bandwidth over the frequency of communication patterns is made explicit in our model.

In Section 2, we review and introduce some basic concepts needed in the discussion. Section 3 introduces the min-cut transformation. In Section 4, we study the number sequence generated by the min-cut transformation. The spectrum based on min-cut transformation is presented in Section 5. The applications to mesh networks are given in Section 6. Applications to other networks and some discussions are made in Section 7. The main results of this work are summarized in the last section.

*For communications with large messages relative to the network's capacity, as is the case with almost all the commercial parallel computers, the propagation time is fairly insignificant compared to transmission time. To see this let w be the weight of the message, c the capacity of the links, ℓ the longest path (diameter) in the network. The propagation time T_p and transmission time T_t are respectively

$$T_p = \ell * t_0, \qquad T_t = w/c$$

where t_0 is the clock period of the network. For a typical FFT algorithm, the message weight w is likely to be near the size of memory per node, which is in the order of 32 mbyes or more on machines such as SP-1, KSR, Meiko, CM5. The capacity of the links on those machines are typically in the order of 10 Mbytes/s. The transmission time T_t is thus in the order of seconds. The diameter r on the other hand is in the order of 10-100, whereas the network clock cycle is in the order of microseconds or even shorter. We thus conclude that the transmission time is at least two or three orders of magnitude higher than the propagation time in this case.

0-8186-6965-9/95 $04.00 © 1995 IEEE

2 Preliminaries

2.1 Networks and Communications

A communication *network* is a tuple $H = (G, P, \psi)$ where

- $G = (U, E)$ is a connected directed graph, called its *topology*
- P: a subset of U called *terminals*, indexed by integers from 0 to $|P| - 1$.
- $\psi : E \rightarrow R^+$ is a function that maps each edge in E to a non-negative real number called its *capacity* or *bandwidth*.

The nodes in the set $(G - P)$ are referred to as *switching* node. G

A *message* over a network is a triple (p, q, w) where p and q are terminals of the network called *source* and *destination* of the message, w is called its weight. A *message pattern* is a set of messages. A message pattern is *uniform* if all the messages in it have the same weight.

A *bisectional message pattern* of *frequency* i and (uniform) *weight* w is the set of messages

$$\phi(i, w) = \{(p, q, w) \mid p \text{ and } q \text{ differ in } i\text{th least significant bit}\}$$

A *full spectrum of communication* of weight w is the sequence of bisectional communications with all possible frequencies:

$$\Phi(w) = (\phi(0, w), \ldots, \phi(n - 1, w))$$

where $n = \log(|P|)$. An example of full spectrum communication is given in Figure 1.

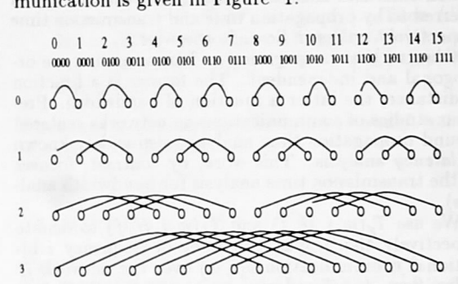

Figure 1: A full spectrum communication over 16 nodes and a mapping onto a two dimensional mesh. The mapping is explained in Section 6, and can be shown to be optimal in terms of both propagation time and transmission time.

2.2 Message Flow

A *flow* on a network H for a communication pattern is a mapping f which assigns each message (p, q, w) in the pattern a path from p to q, and a real number c called the *strength* or *weight* of the flow for the message. The flow is said to have *uniform weight* if the strength assigned to all messages is equal, *uniform path* if the lengths of all the paths are equal. The flow is *uniform* if it has both uniform weight and uniform path. In this paper, we are primarily concerned with uniform flow for uniform bisectional communication patterns. A flow for a sequence of communication

patterns is simply a sequence of flows, one corresponds to one pattern in the sequence.

We assume the network is clocked. Let f be a flow over H for a pattern $s, (p, q, w) \in s$, and $f(p, q, w) = (z, c)$. Then the source p injects a *parcel* of weight c into the network at each clock cycle. A flow is not *valid* unless

- the sum of weights of all parcels that travel through a link at any time step is smaller or equal to the bandwidth of the link.
- for any node, the total weight of outgoing parcels at clock cycle t equals the total weight of incoming parcels not destined to the node at cycle $t-1$.

In other words, a valid flow never overflows a link, and never causes accumulations of messages at any intermediate node.

2.3 Performance Measurement

Let f be a uniform flow over H for a frequency i bisectional communication $\phi(i, w)$ with strength c and path length ℓ. We define ℓ and w/c to be respectively the *abstract propagation time* and *abstract transmission time* of the communication under flow f. Let the network be clocked at frequency ω, $\tau = 1/\omega$, then the *concrete propagation time* is the product of abstract propagation time and τ, the *concrete transmission time* is the product of the transmission time and τ. In the following discussions, abstract propagation time and abstract transmission time are often referred to by propagation time and transmission time respectively unless indicated otherwise.

Observe that propagation and transmission are orthogonal and independent. The former is a function of distance, the latter a function of bandwidth. Previous studies of communications on networks centered around propagation time analysis, also widely known as *latency* analysis. This work, by contrast, focuses on the transmission time analysis (or bandwidth analysis).

We use $T_p(w, i, H, f)$ and $T_f(w, i, H, f)$ to denote respectively the propagation time of frequency i bisectional communication $\phi(i, w)$ over the network H under flow f. Similarly, we use $T_p(w, H, f)$ and $T_f(w, H, f)$ to denote respectively the propagation and transmission time for the full spectrum of bisectional communications. For notational convenience, we will also sometimes omit the network H and/or flow f when one or both are clear from the context. For instance, $T_t(64k)$ refers to the transmission time for a full spectrum communication of weight 64k over a certain network under a certain flows.

2.4 Characterizations

We define a set of network parameters.

- propagation diameter $R_p(H) = T_p(x, H)$, where x is any weight
- transmission diameter $R_t(H) = T_t(1, H)$
- bandwidth for frequency i, $B(H, i) = 1/T_t(1, H, i)$
- ensemble bandwidth $B(R_t(H)) = 1/R_t(H))$,
- propagation dimension $K_p(H) = k_p$, where k_p satisfies
$$R_p(H) = k_p(N^{1/k_p} - 1)$$
and N is the number of terminal nodes in H.
- transmission dimension $K_t(H) = k_t$, where k_t satisfies
$$R_t(H) = k_t(N^{1/k_t} - 1)$$

Note that since propagation and transmission are independent, it is quite possible for a network to have different transmission and propagation dimensions.

The *response spectrum* of a network is defined to be the sequence

$$(B(H, 0), B(H, 1), \ldots, B(H, \log(n) - 1))$$

The response spectrum of a network thus gives the effective bandwidth of the network for communications of all different frequencies. The response spectrums of a 1-d mesh, 2-d mesh, 3-d mesh, and a 12 dimensional hypercube are plotted in Figure 5.

Finally, we say two networks are *equivalent* denoted by $\cong$ if and only if they have the same response spectrum.

2.5 Notation of sequences

A sequence is usually given in the form enumeration $S = (s_0, s_1, \ldots, s_{n-1})$. The length of the sequence is denoted by $|S|$. We sometimes use the expression $< s_i >$ to denote a sequence, which consists of $(s_0, s_1, \ldots, s_{n-1})$ when the values of the terms and the length of the sequence are given by the context.

Given sequences $s =< s_i >, s' =< s'_i >$. We define

- Constant multiply: $cs =< cs_i >$.
- Normalization: $!s =< s_i/s_0 >$

3 Min-Cut Transformation

In this section, we introduce the *min-cut* transformation as a tool in the bandwidth analysis of networks. This transformation maps a wide range of networks to sequences of real numbers with lengths logarithmic to the numbers of terminals in the networks. The bandwidth analysis can then be performed over the sequences.

3.1 Min-cut transformation

By graph theory, a *cut* of a connected graph is a set of edges whose removal results in two disconnected subgraphs. Suppose the edges are weighted, the *weight* or *bandwidth* of a cut is the sum of the weights of all the edges in the cut. A *min-cut* is a cut that has minimum weight. The min-cut is *symmetric* if the two subgraphs are isomorphic. A graph is a *symmetric graph* if symmetric min-cuts exist recursively.

We define the *min-cut transformation* of a symmetric network H to be the sequence of

$$\mu(H) = (c_0, c_1, \ldots, c_{m-1})$$

where c_{m-i} is the bandwidth of the ith cut when the network is recursively cut. Note the weight of the first cut is the last entry and the weight of the last cut is the first entry in the sequence. We refer to the cut corresponding to the weight c_i in the sequence the cut of *level* i, which is counted from left to right.

As an example, a 16-processor network M with uniform link-bandwidth of one, connected by a 4 by 4 two dimensional mesh, has the min-cut transformation of

$$\mu(M) = (1, 2, 2, 4)$$

as shown in Figure 2.

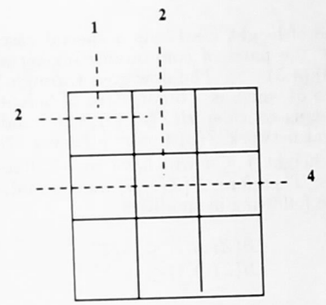

Figure 2: The min-cut transformation of two dimensional mesh

It should be pointed out that while symmetry seems to be a severe restriction over the domain of networks, it is a property that all networks found in real parallel machines seem to possess. Examples include those in IBM SP-1, KSR KSR-1, MasPar MP-2, Intel Paragon, Meiko CS-2, TMC CM-5.

We define a sequence of numbers $C = (c_0, c_1, \ldots, c_{m-1})$ as *proper* if $2c_i \geq c_{i+1}$ for all i's. An important property of the min-cut transformation is the *properness* of the produced sequence as stated in the following theorem.

Theorem 1 *A sequence is a min-cut sequence if and only if it is proper.*

Proof: The *if* part of the lemma should be obvious. We will only prove the *only if* part by contradiction. Let $c1$ be the weight of a cut at step i, and $c2$ be the weight of the next cut. Suppose $c1 > 2c2$, then there must exist another cut at the ith step, whose weight is $2c^{i+1}$, which is smaller than c^i (see Figure 3.1). This means the $c1$ was not a min-cut.

3.2 Linearity of Min-Cut Transformation

In this section, we show that min-cut transformation is a linear operation with respect to well-defined operations. The linearity of this transformation is important since it means that the transformation of a network can be derived from the transformations of the network's component networks.

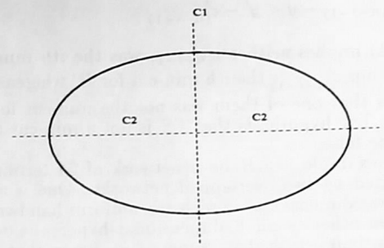

Figure 3: The schematic illustration for the proof of the properness of min-cut.

Let us define two operations over networks: number multiplication, and addition. Given a network H, its product with a number a is the network where each link's capacity is multiplied by a. Given two networks with the same terminal nodes, their sum is the network in which the terminals are inter-connected by both the two networks. More formal definitions are given as follows:

multiplication Let $H = (G, P, \psi)$, then $aH = (G, P, \psi')$ where $\psi'(\epsilon) = a\psi$ if and only $\psi(\epsilon) = \psi$.

addition Let $H = (G, P, \phi), H' = (G', P, \psi')$. Observe that they share the same terminal set P, where $G = (V, E), G' = (V', E')$. Then $(H + H') = (G'', P, \phi'')$, where $G'' = G \cup G'$ with nodes in P overlapped; $\psi''(\epsilon) =$ if $(\epsilon \in E)$ then $\psi(\epsilon)$ else $\psi'(\epsilon)$.

Theorem 2 (*Linearity*) *The min-cut transformation is a linear transformation:*

1. $\mu(aH) = a\mu(H)$
2. $\mu(H + H') = \mu(H) + \mu(H')$

Proof:
We only give the proof for (2) because (1) is obvious. Let $\mu(H) = (c_0, \ldots, c_{n-i-1}, \ldots, c_{n-1}), \mu(H') = (c'_0, \ldots, c'_{n-i-1}, \ldots, c'_{n-1}), \mu(H + H') = (s_0, \ldots, s_{n-i-1}, \ldots, s_{n-1})$. Thus the ith cut E_i for H has capacity $c_{(n-i-1)}$, and the ith cut E'_i for H' has capacity c'_{n-i-1}. Clearly, $EE = E_i \cup E'_i$ is a cut for $HH = H + H'$ at the ith stage with the capacity of $c_{(n-i-1)} + c'_{(n-i-1)}$. All we need to show is that the cut EE is a min-cut for HH.

Suppose EE is not a min-cut for HH , then there should be another cut YY with capacity $yy < s_{n-i-1}$. The cut YY can be decomposed into two cuts Y and Y' where Y is a cut for H with capacity y, Y' a cut for H' with capacity y'. We now have $yy = y + y' < s_{n-i-1} = c_{n-i-1} + c'_{n-i-1}$. Equivalently, one of the following must hold

1. $y < c_{(n-i-1)}$ and $y' < c'_{(n-i-1)}$

2. $c_{(n-i-1)} - y < y' - c'_{(n-i-1)}$

But (1) implies neither $c_{(n-i-1)}$ was the ith min-cut for H, nor $c'_{(n-i-1)}$ the ith min-cut for H' whereas (2) implies that one of them was not the min-cut for H or H'. The hypothesis that EE is not a min-cut thus must be false.

For example, let H be a network of 64 terminals connected by two overlapped networks. One is an 8 by 8 two dimensional mesh with uniform bandwidth 8. The other is an 8 dimensional hypercube with uniform bandwidth 0.5. Suppose we know that the min-cut transformation of a 64 node 2-d mesh with unit bandwidth, denoted by $M^2(64)$, has the min-cut transformation (1,2,2,4,4,8), and the min-cut transformation of a 64 node hypercube with unit bandwidth, denoted by $M^8(64)$, is (1,2,4,8,16,32). Then, since $H = 8M^2(64) + 0.5M^8(64)$, we have

$$\begin{aligned}\mu(H) &= \mu(8M^2(64) + 0.5M^8(64)) \\ &= 8\mu(M^2(64) + 0.5\mu(M^8(64)) \\ &= 8(1,2,2,4,4,8) + 0.5(1,2,4,8,16,32) \\ &= (8,16,16,32,32,64) + (0.5,1,2,4,8,16) \\ &= (8.5,17,18,36,40,80)\end{aligned}$$

Note that the min-cut transformation of this network is derived from the linear property of the transformation without actually "cutting" the network.

4 Sequence Analysis

In the previous section, we showed how the min-cut transformation maps symmetric networks into sequences of real numbers. This section is devoted to the analysis of sequences. In the next section, we will put the two things together, and show how a network can be analyzed by a min-cut transformation followed by a sequence analysis. A tool we use in the sequence analysis is a simple model called *canonical networks* which can be thought as the physical interpretations of the sequences by min-cut transformation.

4.1 Canonical networks

A canonical network is a complete binary tree where the capacity of any edge at height i is s_i, for $i = 0$ to $n-1$. The $N = 2^n$ terminals are leaves, indexed from left to right by the numbers from 0 to $N-1$. Clearly, a canonical network defines a sequence, and vice versa. We use $Z(s)$ to denote a canonical network corresponding to a sequence s. An example of a canonical network $Z(c_0, c_1, c_2, c_3)$ with 16 nodes is given in Figure 4.

It is fairly straightforward to see that a bisectional communication of frequency i uses all links with height smaller than or equal to i on a canonical network. More concretely, given $\phi(i,w)$, the frequency i bisectional communication (of weight w) and a canonical network $Z(s)$, a message in $\phi(i,w)$ will go through some node(s) and link(s) of height $(0,1,\ldots,i)$. Moreover, links of height greater than i are not used, the flow for $\phi(i,w)$ therefore is a conjunction of 2^{n-i-1} disjoint and identical sub-flows, each of which is over

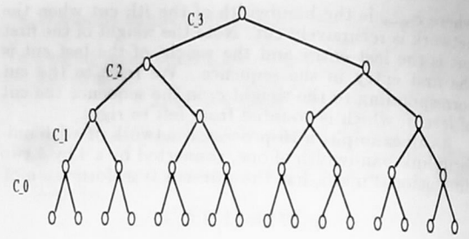

Figure 4: An example of a canonical network $Z(c_0, c_1, c_2, c_3)$

a sub-tree of height $i+1$. As a special case, for frequency 0, the pairs of communicating terminals are (0,1),(2,3),(4,5), ... The flow goes through N/2 subtrees, each of which is a binary tree of height one.

Now let us consider $B(Z(c),\ i)$, the bandwidth of a canonical network $Z(c)$ for frequency i. Note that a link at height $j \le i$ will have to be shared by 2^j processors. Therefore, $B(Z(c), i)$ must simultaneously satisfy the following inequalities.

$$\begin{aligned}&B(Z(c), i) \le c_0/2^0, \\ &B(Z(c), i) \le c_1/2^1, \\ &\ldots, \\ &B(Z(c), i) \le c_i/2^i.\end{aligned}$$

from which we conclude

Theorem 3 *The bandwidth for bisectional communication of frequency i on a canonical network Z(s) is*

$$B(Z(c), i) = min(c_j/2^j \mid j = 0 \text{ to } i-1)$$

Given a sequence $c = (c_0, \ldots, c_{n-1})$, we define a *power divided min sequence* denoted by $*c$ by

$$\begin{aligned}*c \ = (&c_0, min(c_0, c_1/2^1), min(c_0, c_1/2^1, c_2/2^2), \\ &\ldots, min(c_0, c_1/2^1, \ldots, c_{n-1}/2^{n-1}))\end{aligned}$$

This allows us to rewrite the statement in Theorem 3 as $B(Z(c), i) = *c_i$. Moreover, we have

Corollary 1 *The response spectrum of a canonical network Z(c), denoted by B(Z(c)) is the sequence*

$$(*c_0, *c_1, \ldots, *c_{n-1})$$

Example 1

1. B(Z(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1))
   =(1,1/2,1/4,1/8,1/16,1/32,1/64,1/128,
   1/256,1/512,1/1024,1/2048)
2. B(Z(1,2,2,4,4,8,8,16,16,32,32,64))
   =(1,1,1/2,1/2,1/4,1/4,1/8,1/8,1/16,1/16)
3. B(Z(1,2,4,4,8,16,16,32,64,64,128,256))
   =(1,1,1,1/2,1/2,1/2,1/4,1/4,1/4,1/8,1/8,1/8)

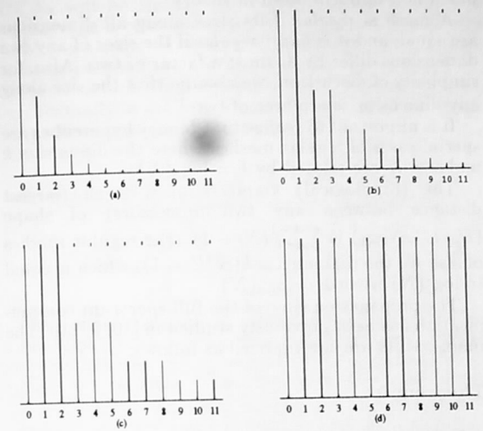

Figure 5: The response spectrum of the canonical networks in Example 1. As will be seen, they can also be interpreted as the response spectrums of a one dimensional mesh (a), two dimensional mesh (b), three dimensional mesh (c), and a twelve dimensional hypercube (d) respectively, all with 4096 processors. The effective bandwidth for high frequency communications increases with the dimension of the network.

4. B(Z(1,2,4,8,16,32,64,128,256,512,1024,2048,4096)
   = (1,1,1,1,1,1,1,1,1,1,1,1,1)

which are plotted in Figure 5.

The transmission time for bisectional communication depends on the dimension and the weight of the communication, as well as the sequence of the network. More precisely,

Theorem 4

$$\begin{aligned}&T_t(w, i, Z(c)) \\ =\ &w/min(s_0, s_1/2^1, s_2/2^2, \ldots, s_{i-1}/2^{i-1}) \\ =\ &w/*c_i\end{aligned}$$

Proof: It follows directly from Theorem 3 and Corollary 1.

The transmission diameter and dimension of a canonical network can also be computed easily

Theorem 5 *Let* $c = (c_0, c_1, \ldots, c_{n-1})$.

1. *transmission diameter:*

$$R_t(Z(c)) = (1/*c_0 + 1/*c_1 + \cdots + 1/*c_{n-1})$$

2. *transmission dimension:*

$$K_t(Z(c)) = k_t$$

where k_t *is the root for* $R_t(Z(c)) = k_t(N^{1/k} - 1), N = 2^n$.

4.2 Normalization

Given a sequence $s = (s_0, s_1, \ldots, s_{n-1})$, we can convert it to its normalized form $!s = (1, s_1/s_0, \ldots, s_{n-1}/s_0)$. For a network $H(s)$, we refer to $H(!s)$ as its *normalization*, $!s$ as its *structure*, and s_0, the first term of s, as its *base*. Furthermore, we say two networks H(s) and H(t) are *similar*, denoted by $H(s) \sim H(t)$ if and only if $!s = !t$. In other words, the two networks share the same structure.

The bandwidth and transmission time on a canonical network for all frequencies i can be shown to be related to those of network's structure by a the constant factor equal to the network's base.

Theorem 6

1. $B(w, i, Z(ac)) = a(B(w, i, Z(c))$,
2. $T_t(w, i, Z(ac)) = T_t(w, i, Z(c))/a$.

This allows us to focus our attention on the structure of canonical networks. Since the similar relation "$\sim$" is obviously an equivalence relation, the domain of all possible networks are partitioned into many (infinite) equivalent classes each of which contains an infinite number of networks. Theorem 6 allows us to study each equivalent class of infinite members by studying one representative, that is the normalized network in that class.

5 Spectrum Analysis

The min-cut transformation is a powerful tool for network spectrum analysis because a network is equivalent to the canonical network defined by the min-cut sequence. Formally,

Theorem 7 *Let H be a symmetric network,* $\mu(H)$ *its min-cut transformation, then* $H \cong Z(\mu(H))$.

Proof: Let $\mu(H) = (c_0, c_1, \ldots, c_{n-1}) = c$. We index the terminals of H in such a way that two terminals are respectively in the two subgraphs generated by the cut with capacity c_i if and only if they differ in their ith least significant bit. We omit the proof that this can always be done (see Figure 6 for an example.) Note that there are always 2^i terminals on each side of each cut at level i, and the cut has the capacity of c_i. Although the messages may go across links in the cuts of other level(s), the properness property of the min-cut transformation (Theorem 1) ensures that the min-cut of the flow (the bottleneck) occurs at cut of level i. The strength of the flow therefore is $c_i/2^i$. This holds for any frequency i, which means the network H has the same response spectrum as the canonical network $Z(\mu(H))$.

The above theorem allows us to perform the spectrum analysis of communication networks in terms

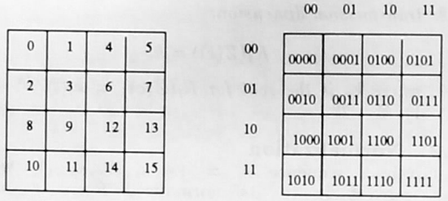

Figure 6: An example of mapping so that messages of frequency i go across cuts at level i. This mapping is obtained by a bit-permutation of the terminal indices as illustrated in Figure 7.

of their min-cut transformations. Since the min-cut transformations of networks are sequences with length logarithmic to the size of the network, it can be analyzed much easily than the original network.

It should also be pointed out that while min-cut transformation preserves information about spectrum analysis and transmission time related parameters, it does not preserve the information required for propagation time related analysis. As a matter of fact, all symmetric networks are mapped by the min-cut transformation to canonical networks with logarithmic propagation diameter.

The following proposition follows Theorem 7

Theorem 8 *Let H be a symmetric network, H' the canonical network defined by H's min-cut transformation, then we have*

1. *The transmission diameters of H and H' are equal;*
2. *The transmission dimension of H and H' are equal;*
3. *The ensemble bandwidth of H and H' are equal.*

6 Mesh Networks

Meshes are simple and among the most widely used networks in parallel architectures. This section is devoted to the analysis of mesh networks. Although the discussion focuses on parameters based on transmission, some discussion about propagation analysis is included for the purpose of comparison.

6.1 Basics

A *k dimensional mesh* of *shape* $(N_0, N_1, \ldots, N_{k-1})$ consists of the following terminals

$$P = \{(x_0, x_1, \ldots, x_{k-1} \mid 0 \le x_i < N_i\}$$

and no switching nodes. N_i is the *size* of the ith dimension for $i = 0$ to $k-1$ whereas $N = \prod_{i=0}^{k-1} N_i$ is the size of the mesh. Two terminals are connected by a link if and only if their coordinates differ only along one dimension by one. For instance, (1,2,3) is connected to (1,3,3) but not to (3,2,3) on a three dimensional mesh. All the links on a mesh network in practice have uniform bandwidth although one can define a non-uniform mesh in theory.

A mesh is *regular* if its sizes along all dimensions are equal, and it is *quasi-regular* if the sizes of any two dimensions differ by at most a factor of two. Also, for simplicity of discussion, we assume that the size along any dimension is a power of two †

It is important to realize that binary hypercubes are special cases of regular meshes where the dimension k and size N are related by $k = \log_2(N)$.

The (topological) *diameter* of a mesh (largest distance between any two processors) of shape $(N_0, \ldots, N_{k-1})$ is $\sum_{i=0}^{k-1}(N_i - 1)$. For regular meshes of size N, the diameter is $k(N^{1/k} - 1)$, which is equal to $\log_2(N)$ when $k = \log_2(N)$.

The propagation time of the full spectrum communication has been previously studied in [10, 9, 11]. The main results are summarized as follows:

Theorem 9

- *A full spectrum communication on any dimensional mesh takes propagation steps that is at least equal to the diameter of the mesh.*
- *Optimal mappings of full spectrum communication exist such that the propagation steps equal the diameter of the mesh.*

We showed that the propagation optimal mappings are not unique, and in fact any *bit-permuting mapping* [11, 3] is optimal. There is, however, only one mapping that has the property that the communication distances are monotonically increasing with the frequency. This mapping is illustrated in Figure 6.1 for the case of two dimensional meshes.

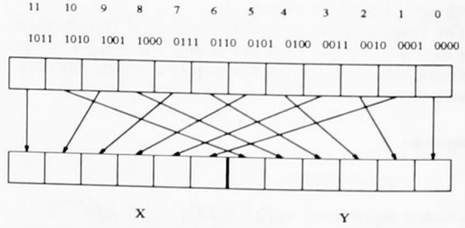

Figure 7: Mapping of full spectrum communication by bit-permutation for a two dimension mesh. The binary number $b_{n-1}, \ldots, b_0$ of index i is permuted to obtain two new binary numbers, one consists of all the bits in odd positions, the other consists of all the bits in even positions. These two binary numbers are then interpreted as integers which define the mesh coordinates of the processor for index i.

†This is also justified by the fact that most commercial mesh based machines such as MasPar MP-1, WaveTracer DTC, TMC CM-2 have power-of-two sizes along all dimensions.

6.2 The s(k,n) sequences

As will be shown, the min-cut transformation of quasi-regular meshes produces a family of sequences which we refer to as $s(k,n)$ sequences, with certain strong properties. In this section, we study $s(k,n)$ sequences and their corresponding canonical network.

Formally, a $s(k,n)$*sequence* is a sequence of length n consisting of $\lceil n/k \rceil$ subsequences as in

$$s(k,n) = (\underbrace{1, 2, \ldots, 2^{k-1}}_{k}, \underbrace{2^{k-1}, 2^k, \ldots, 2^{2k-2}}_{k}, \ldots, \underbrace{2^{n/k-(k-1)}, 2^{n/k-(k-2)}, \ldots, 2^{n/k-(k-m)}}_{m})$$

where m is n modular k. Although the definition looks complicated, there is a very simple way to construct the sequences. We will construct k numbers at a time from left to right. The first k numbers are always $1, 2, \ldots, 2^{k-1}$. To construct any other group of k numbers, repeat the last number in the previous group as the first in the current group, and double it each time to get an additional number until all the numbers are generated for the group. The last group may and may not contain k numbers depending on whether n is a multiple of k. But the rule for constructing the numbers is the same, which is to repeat the last number in the previous group, and double it each time until all the numbers in the group are generated. For example,

Example 2

1. s(1,8) = (1,1,1,1,1,1,1,1)
2. s(2,8) = (1,2, 2,4, 4,8,8,16,16,32)
3. s(3,11) = (1,2,4,4,8,16,16,32,64,64,128)
4. s(10,10)= (1,2,4,8,16,32,64,128,256,512,1024)

Note that the definition does not require that n be a multiple of k. The canonical network defined by a $s(k,n)$ sequence has the following property

Theorem 10 *Let $M = Z(s(k,n))$ be the canonical network defined by the $s(k,n)$ sequence, n is a multiple of k, $N = 2^n$. Then*

1. *the transmission dimensionality of M is k,*
2. *the transmission diameter over M is exactly $k(N^{1/k} - 1)$.*

We omit the proof since it follows directly from Theorems 4 and 5. It should also be noted that when n is not a multiple of k, the transmission dimensionality of the corresponding network is a non-integer that is smaller than k.

As some examples, we have

Example 3

1. $R_t(Z(d(1,n))) = n, D_t(Z(d(1,n))) = 1$
2. $R_t(Z(d(2,10))) = 62, D_t(Z(d(2,10))) = 2$
3. $R_t(Z(d(3,12))) = 45, D_t(Z(d(3,12))) = 3$
4. $R_t(Z(d(n,n))) = n, D_t(Z(d(n,n))) = n$
5. $R_t(Z(d(2,5)) = 10, D_t(Z(d(2,5))) = 1.879798$

It should be obvious that $Z(d(1,n))$ corresponds to a conventional tree, and $Z(d(n,n))$ a *fat-tree* [1].

6.3 Mesh analysis

The following theorem reveals the relations between mesh and $s(k,n)$ sequences under the min-cut transformation.

Theorem 11 *Let M(k,N,c) be a quasi-regular mesh, where k is the dimensionality, N is the total number of terminals, c is the uniform bandwidth of the links. Then $\mu(M(k,u,c)) = c \cdot s(k, \log_2(N))$.*

The following are some examples for the above theorem. Keep in mind that $M(k,N,c)$ denotes a k dimensional quasi-regular mesh of size N with uniform link bandwidth c.

Example 4

1. 1-d mesh: $\mu M(1,64,8) = 8(1,1,1,1,1,1,1,1)$
2. 2-d mesh: $\mu M(2,1024,4) = 4(1,2,2,4,4,8,8,16,16,32)$
3. 3-d mesh: $\mu M(3,256,1) = (1,2,4,4,8,16,16,32,64)$
4. 10-d hypercube: $\mu(M(10,1024,2) = 2(1,2,4,8,16,32,64,128,256,512)$

By Theorem 11, the examples in Example 1 are the min-cut transformation of one, two, three, and ten dimensional regular meshes of size 4096 and unit bandwidth. The spectrums plotted in Figure 5 can now be interpreted as the response spectrums of the four meshes of dimensions one, two, three, and twelve respectively. Observe that higher dimensional meshes have higher responses to higher frequency communications.

The correspondence between meshes and the $s(k,n)$ sequences allows us to calculate the response spectrum and other transmission parameters easily.

Theorem 12

1. *Let $M = M(k,N,c)$ be a k dimensional quasi-regular mesh,$n = \log_2(N)$. Then the spectrum of M is given by the length n sequence*

$$u(k,n) = c(\underbrace{1, 1, \ldots, 1}_{k}, \underbrace{1/2, 1/2, \ldots, 1/2}_{k}, \ldots, \underbrace{1/2^{n/k-1}, 1/2^{n/k-1}, \ldots, 1/2^{n/k-1}}_{m})$$

where $m = k$ if n is a multiple of k, $m = n$ modular k otherwise.

2. *The transmission time for frequency i of unit weight is $c/(u(k,n)_i)$, and has the form of a product between the base bandwidth c and a power of two. The transmission diameter is given by*

$$\begin{aligned} R_t(M) &= 1/c\ (p\sum_{i=0}^{p-1} 2^i) + 1/c\ (m2^{n/k-1}) \\ &= 1/c\ (N_0 + N_1 + \cdots + N_{k-1} - k) \end{aligned}$$

where $p = \lfloor n/k \rfloor$ -1, m the same as in (1), N_i is the size for the ith dimension for $i = 0$ to $k-1$.

Proof: By definition of the response spectrum and Theorem 11.

Now we are in the position to present the relations between the characterizations based on transmission and those based on propagation.

Theorem 13 *Let M be a quasi-regular mesh of unit base bandwidth, R and D respectively the transmission diameter and dimension, R' and D' respectively the propagation diameter and dimension. Then R = R', D=D'.*

Proof: By Theorem 12 (2), the transmission diameter is equal to the physical diameter of the mesh topology, which in turn equals to the propagation diameter for the spectrum communication by [11, 10]. That the two diameters are equal in turn implies that the transmission dimension and propagation dimension are equal.

7 Other Applications

The previous section shows that the spectrum analysis is effective in the analysis of mesh networks. In this section, we demonstrate that it can also be effectively applied to other communication networks.

As an example, let us consider a butterfly network of N terminals and $\log(n)$ switching nodes. Let c be the uniform link bandwidth of the network. It is easy to see that the min-cut transformation is

$$c(1, 2, 4, 8, \ldots, N/2)$$

which is the same as a binary hypercube with the same number of nodes. The butterfly network thus has uniform bandwidth for communication of all frequencies and is equivalent to hypercube of the same size.

Tree networks, strictly speaking, are not symmetric networks. However, if we (recursively) remove the root and connect the links adjacent to the root, the tree network becomes a symmetric network provided the tree-arity is even. The min-cut transformation for trees thus exists as long as the trees are symmetric (e.g. completed binary trees). The min-cut transformation of a symmetric tree networks is in fact the tree itself on which the spectrum analysis can easily be performed in the first place. When the tree is conventional, i.e. the links are of unit bandwidth at all levels, it is equivalent to a one dimensional mesh in terms of its response spectrum despite the fact that it has much smaller physical diameter. The dimensionality of conventional trees is thus one.

Recently, the so-called fat-trees have gained some popularity in the parallel computing community. Fat trees are essentially canonical networks with a sequence of link bandwidths that can be assigned to different values. If the sequence consists of identical numbers, the fat-tree degenerates to a non-fat conventional tree. At the other extreme, if the sequence is such that each number is twice as large as the one before, the network will have the same response spectrum as a binary hypercube and is referred to as a *complete fat-tree*[‡]. Since complete fat-trees are expensive to build, *incomplete fat-trees* are often adopted. The CM-5 by Thinking Machines is an example of a network based on a non-complete fat-tree network.

Let us consider an incomplete fat tree with the bandwidth sequence of

$$(1, 2, 4, 4, 8, 8, 16, 16)$$

of 256 terminals. Its response spectrum is thus

$$(1, 1, 1, 1/2, 1/2, 1/4, 1/4, 1/8)$$

which has an equal or narrower bandwidth for all frequencies than a three dimensional mesh network with unit bandwidth. The transmission diameter is the sum of all the reciprocals in the response spectrum which is equal to 23. For comparison, a unit bandwidth two dimensional mesh of the same number of terminals has a not much larger diameter of 30. Not surprisingly, the transmission dimensionality of this fat tree, which is the root of the equation

$$k(256^{1/k} - 1) = 1 + 1 + 1 + 2 + 2 + 4 + 4 + 8 = 23$$

is only 2.320. In other words, we expect this fat-tree to behave very much like a two dimensional mesh in terms of its transmission performance, and not respond to high frequencies very well. This example shows that neither the small topological diameter nor the fact that there is a topological isomorphism between fat-trees and hypercubes can automatically contribute to the transmission performance of the network.

Since it is obvious that low-dimensional meshes have low bandwidth response to high frequency communications, it is frequently proposed that a high-dimensional network be added to enhance the overall performance. The additional high dimensional mesh often has much smaller base bandwidth to reduce the overall cost. [§].

Let us consider the case where 1024 processors are connected by a 2-d mesh with the min-cut B1 = (1,2,2,4,4,8,8,16,16,32) as well as a "high-dimensional" network with reduced bandwidth of

$$B2 = 1/16(1, 1, 1, 1, 2, 4, 8, 16, 32, 64)$$

It would be interesting to consider how much this additional high dimensional network has enhanced the network transmission performance. The question can be easily answered by exploiting the linearity of min-cut transformation (Section 3). The response spectrum of the 2-d mesh is (1, 1, 0.5, 0.5, 0.25, 0.25, 0.125, 0.125, 0.063, 0.063), while the response spectrum of the additional network is 0.0625(1, 0.5, 0.25,0.125,0.125, 0.125,0.125, 0.125, 0.125, 0.125). The "high dimensionality" of this network is evidenced by the fact that the bandwidth for frequency 4 or higher communications remains constant. The combined bandwidth spectrum is then (1.0625, 1.0313, 0.5016,0.5008, 0.2508,0.2508,0.1258, 0.1258, 0.0638, 0.0638). The transmission diameter is $T_t = 0.9412 + 0.97 + 1.994 + 1.997 + 3.987 + 3.987 + 7.9491 + 7.9491 + 15.674 + 15.674 = 61.1224$, compared to the transmission diameter of the 2d mesh of 62. The transmission dimension of the composed network is then 2.0111. The contribution of the "high dimensional" global router to the overall dimension is thus only 0.0111.

8 Conclusion

In this paper, we started with an observation that the recent trend of moving from a large number of small processors (LS) to a small number of large processors (SL) in the design of parallel machines calls for new models for network analysis based on bandwidth and transmission time as opposed to those based on distance and propagation time.

We introduced the notion of frequency. We name the model "spectrum analysis" since it attempts to capture the behavior of the network by observing its transmission response to communications with all different frequencies. By an analogy with propagation based models, we redefine the notion of diameter and dimension in the context of transmission time analysis. The transmission diameter and dimension agree in value with their propagation counterpart in the case – and only in the case – of regular meshes with unit link bandwidth. Just as propagation diameter and dimension characterize a network's propagation behavior, the transmission diameter and dimension characterize a network's transmission behavior.

The min-cut transformation is introduced which allows us to study complex networks in terms of short sequences of real numbers. These sequences can be naturally interpreted as tree-structured networks which preserve the response spectrum of networks. By this interpretation, min-cut transformation can be thought as a process that maps symmetric networks to their canonical forms with tree structures. The properties of min-cut transformation, sequences, and canonical forms are studied. The result of this work is a system by which the response spectrum, transmission diameter, and transmission dimension can be automatically derived for a wide range of communication networks. The applications of the model to mesh and several other networks are included in this paper to demonstrate the effectiveness and usefulness of the model.

References

[1] C.E.Leiserson. Fat-trees: Universal networks for hardware-efficient supercomputing. *IEEE Trans. Computers*, c-34(10):892–900, October 1985.

[2] D. Culler, R. Karp, and D. Patterson. Logp: Towards a realistic model of parallel computation. In *Proceedings of the 4th ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming*, pages 1–12, May 1993.

[3] Peter M. Flanders. A unified approach to a class of data movements on an array processor. *IEEE Transactions on Computers*, C-31(9):809–819, September 1982.

[4] Peter M. Flanders and Dennis Parkinson. Data mapping and routing for highly parallel processor arrays. *Future Computing Systems*, 2(2):184–224, 1987.

[5] Donald Fraser. Array permutation by index-digit permutation. *Journal of ACM*, 23(2):298–309, April 1976.

[6] S. Lennart Johnsson and Ching-Tien Ho. Matrix transposition on Boolean n-cube configured ensemble architectures. *SIAM J. Matrix Anal. Appl.*, 9(3):419–454, July 1988.

[7] S. Lennart Johnsson and Ching-Tien Ho. Spanning graphs for optimum broadcasting and personalized communication in hypercubes. *IEEE Trans. Computers*, 38(9):1249–1268, September 1989.

[8] S. Lennart Johosson. Communication in network architectures. In R. Suaya and G. Birtwistle, editors, *VLSI and Parallel Computation*. Morgan Kaufmann, 1990.

[9] Z. G. Mou, C. Constantinescu, and T. Hickey. Divide-and-conquer on a 3-dimensional mesh. In *Proceedings of the European Workshops on Parallel Computing*, pages 344–355, Barcelona, Spain, March 1992.

[10] Z. G. Mou, Cornel Costantinescu, and T. Hickey. Optimal mappings of divide-and-conquer algorithms to mesh connected parallel architectures. In *Proceedings of International Computer Symposium*, pages 273–284, Taiwan, December 1992.

[11] Z. G. Mou and Xiaojing Wang. Optimal mappings of m dimensional fft communication to k dimensional mesh for arbitrary m and k. In M. Reeve and G. Wolf, editors, *Lecture Notes in Computer Science No. 694 - Parallel Architecture and Language Europe (PARLE93)*, pages 104–119. Springer-Verlag, Munich, Germany, June 1993.

[12] Paul N. Swarztrauber. Multiprocessor ffts. *Parallel Computing*, (5):197–210, 1987.

[‡] A number of recent commercial machines, including Meiko CS-2 and TMC CM-5 are based on fat trees. For considerations of fault tolerance and others, the networks often have a higher arity than two and redundant paths between processors.

[§] MasPar's MP-1 and MP-2 machines are such examples, where 1k or more processors are connected by a two dimensional mesh as well as a "global router" which has a much smaller diameter and a much narrower bandwidth than those of the two dimensional mesh.