Proceedings
of the
1982 Conference
on
INFORMATION SCIENCES AND SYSTEMS

Department of Electrical Engineering and Computer Science
Princeton University
Princeton, New Jersey 08544
SOME RESULTS ON PACKET-SWITCHING NETWORKS FOR MULTIPROCESSING
(Extended Abstract)
by
Clyde P. Kruskal
Department of Computer Science, University of Illinois, Urbana, IL 61801
and
Marc Snir
Courant Institute, New York University, New York, NY 10012

ABSTRACT

We present several results characterizing the structure of A-networks and SW banyan networks and their relationship to each other. New analyses of the performance of unbuffered and buffered A-networks are given.

INTRODUCTION

We consider packet switching networks built of switches connected by unidirectional lines. A p input, q output (p,q) switch can receive packets on each of its p incoming lines and forward them through each of its q outgoing lines. Formally, a network is a labelled digraph where nodes are of the following three types:

1. Source nodes which have indegree 0 and outdegree 1;
2. Sink nodes which have indegree 1 and outdegree 0;
3. Switches which have positive indegree and outdegree.

Each p,q switch has its p inputs labelled 0,...,p-1 and its q outputs labelled 0,...,q-1. Thus, labels are associated with both endpoints of each edge.

Two networks are topologically equivalent if their underlying graphs are isomorphic; two networks are isomorphic if there exists a label preserving graph isomorphism between them.

The reversal G^R of the network G is the network obtained from G by reversing the direction of each edge, thus transforming inputs into outputs and outputs into inputs.

Let G be a network, and let S be a subset of nodes in G. The subnetwork induced by S is obtained from the subgraph of G spanned by S by deleting isolated nodes, replacing each node with indegree zero and outdegree k > 1 with k distinct nodes, each connected to one edge, and performing the equivalent transformation on nodes with outdegree zero.

We shall restrict our attention to nonadaptive (or oblivious) routing algorithms: The path of a packet through the network is fixed at the source node issuing it. The path to be used can be simply encoded as the sequence of labels of the successive switch outputs on the path. We shall call this sequence the path descriptor. If a header consisting of a path descriptor is attached to each packet, the routing logic at each switch is very simple: An output is selected according to the first label in the header, and this label is then dropped (Figure 1; see also [W]). Where messages have to be returned through the reverse network (i.e., using the same network in the reverse direction), the description of the path to be followed can be progressively replaced by the description of the path traversed so far. Each switch uses the first label in the header to select an output and then drops it and appends to the end of the header the label of the input at which the packet was received (Figure 1). At each step the two segments of the header uniquely identify the path so far followed and the path yet to be followed by the packet. When the packet reaches its destination at a sink node, the header contains a description of the path followed, i.e., a valid (reversed) path descriptor for that path in the reverse network.

A-NETWORKS

In a general network the paths leading from different source nodes to the same sink node may have different path descriptors. Thus, a routing table is needed at each source node, containing a path descriptor for each sink node. It is convenient to have all these tables identical. Extending the original definition of Patel [Pa1,Pa2], we define a digit controlled or 2-state network (A-network) to be a network with the following two properties:

(i) There is a unique path from each source node to each sink node.
(ii) The path descriptors associated with paths leading to the same sink node are identical.

The first condition defines an A-network to be a banyan network [GL]. However, not every banyan network is also an A-network. The second condition implies that the nodes of an A-network can be arranged by levels, so that the source nodes are all at level zero, and edges connect nodes at level i to nodes at level i+1 only. Note, however, that the sink nodes of an A-network need not be at the same level.

We can take the path descriptor associated with paths leading to the sink node s to be the address of s. Thus, the unique information needed to route a packet to a sink node is the address of that node.

305
The following theorem provides an alternative recursive definition for \( \Delta \)-networks.

**Theorem 1.** A network is a \( \Delta \)-network if and only if

(i) It consists of a single switch, or
(ii) It consists of one level of \( k \)-output switches followed by disjoint \( \Delta \)-networks \( A_1, \ldots, A_j \). The source nodes of the \( k \)-output switches are connected to the inputs of the switches in the first level, and output \( j \) of each switch in that level is identified with a source node of \( A_j \).

We use the following lemma in the proof of Theorem 1.

**Lemma 2.** Let \( G \) be a \( \Delta \)-network, \( S \) be a set of nodes in the \( i \)-th level of \( G \), and \( S' \) be the set of nodes consisting of \( S \) and all the descendents of nodes from \( S \). Then the subnetwork \( G' \) induced by \( S' \) is a \( \Delta \)-network.

**Proof:** Clearly, there is a unique path from each node of \( G' \) to each sink node of \( G' \). Each source node of \( G' \) corresponds to a node in \( S \) and each sink node of \( G' \) is also a sink node of \( G \). Let \( p \) and \( p' \) be descriptors for paths from two distinct source nodes \( i \) and \( i' \) of \( G' \) to the same sink node. Let \( q \) and \( q' \) be descriptors for paths leading from source nodes of \( G \) to the nodes corresponding to \( i \) and \( i' \). Then \( dp = q'p' \) and \( q = |q'| \), so that \( p = p' \).

**Proof of Theorem 1:** Let \( G \) be a \( \Delta \)-network. If \( G \) contains a single level of switches then \( G \) will contain exactly one switch. Otherwise let \( S \) be the set of switches in the first level of \( G \). The uniqueness of paths descriptors implies that all the switches in \( S \) have the same number of outputs. Consider the set of descendents of the outputs of nodes in \( S \) labelled with \( j \). Let \( G_j \) be the subnetwork induced by the union of \( S \) with this set. The uniqueness of paths implies that the networks \( G_j \) are pairwise disjoint. By lemma 2, each of these subnetworks is a \( \Delta \)-network.

The converse implication is immediate.

A rectangular \( \Delta \)-network of degree \( k \) and order \( n \) is a \( \Delta \)-network with \( N = k^n \) source nodes and \( N' = k^n \) sink nodes built of \( k \times k \) switches. In such networks each output-node address is a sequence of \( n \) \( k \)-valued labels, which we may take to be an integer in the range \( 0, \ldots, k^{n-1} \), written in radix \( k \). We shall assume henceforth that each sink node is labelled with its address. Alternatively, rectangular \( \Delta \)-networks can be recursively characterized in the following way.

(i) A rectangular \( \Delta \)-network of degree \( k \) and order \( n \) has one \( k \times k \) switch connected to each source node and \( k \) sink nodes.

(ii) A rectangular \( \Delta \)-network of degree \( k \) and order \( n \) (of level \( n \)) consists of one level of \( k^{n-1} \) \( k \times k \) switches followed by \( k \) disjoint rectangular \( \Delta \)-networks \( A_1, \ldots, A_n \) of degree \( k \) and order \( n-1 \). The source nodes of the \( k \)-output switches are connected to the inputs of the switches in the first level, and output \( j \) of each switch in that stage is identified with a source node of \( A_j \) (see Figure 2).

The equivalence of these two definitions follows from Theorem 1.

Not all the rectangular \( \Delta \)-networks of degree \( k \) and order \( n \) are isomorphic, or even topologically equivalent. The switches in the first stage can be attached to the subnetworks in the remaining stages in any arbitrary order, and furthermore these subnetworks need not be equivalent. An example of two topologically nonequivalent rectangular \( \Delta \)-networks of degree 2 and order 3 is given in Figure 3.

The recursive characterization of \( \Delta \)-networks also illustrates the close relationship between \( \Delta \)-networks and Benes networks [Be,Jo]:

**Corollary 3.**

(i) A network is a Benes network if and only if it consists of a rectangular \( \Delta \)-network \( G \) followed by \( G' \), where each switch in the last stage of \( G \) is identified with the corresponding switch in the first stage of \( G' \).

(ii) If \( G \) is a \( \Delta \)-network where the number of inputs at each switch does not exceed the number of outputs of that switch, then \( G \) followed by \( G' \) as above is a permutation network.

**Proof:** (i) is an immediate consequence of the definition of Benes networks. The proof of (ii) is a simple generalization of the proof Waksman gives in [Wa] for Benes networks.

To summarize, a rectangular \( \Delta \)-network of degree \( k \) and order \( n \) has \( k^n \) source nodes, \( k^n \) sink nodes, and \( nk^{n-1} \) switches organized in \( n \) levels containing each \( k^{n-1} \) switches, with connections between adjacent levels only. A switch at stage \( j \) has \( k^j \) predecessors at stage \( j-r \) and \( k^r \) successors at stage \( j+r \), and is connected to each of them through a unique path. In particular, it is connected through unique paths to \( k^j \) source nodes and \( k^{n-j} \) sink nodes. While these properties are sufficient to justify the performance analyses given below, they do not uniquely characterize the topology of rectangular \( \Delta \)-networks. The reverse of a \( \Delta \)-network fulfills the above conditions, but the reverse of a \( \Delta \)-network drawn in Figure 3.2 is not topologically equivalent to \( \Delta \)-network. Moreover, it is possible to define networks fulfilling above conditions which are neither equivalent to a \( \Delta \)-network nor to a reversed \( \Delta \)-network.

**\( \Delta^2 \)-Networks**

Where the reverse network is used to transmit replies, it is worthwhile to have a \( \Delta^2 \)-network in both directions. A network \( G \) is a \( \Delta^2 \)-network if both \( G \) and \( G' \) are \( \Delta \)-networks. In a \( \Delta \)-network each source node is associated with the unique address corresponding to the path descriptors of the paths leading to that node in the reverse network. It is easy to see that all the paths connecting source nodes to sink nodes in a \( \Delta \)-network have the same length. In particular, a \( \Delta \)-network has the topology of an \( n \)-level banyan [GL].

306
G is a rectangular $\Delta^2$-network of degree k and order $n$ if both G and $G'$ are rectangular $\Delta^2$-networks of degree k and order $n$. It is easy to check that $\Delta$-networks [La], baseline networks [WF], flip networks [Ba], and indirect binary cube networks [Fe] all satisfy the above definition. Using special case arguments, Wu and Feng showed that these networks to be topologically equivalent to an SW banyan network. The topological equivalence of these networks turns out to be a corollary of the following general result.

**Theorem 4.** Any two rectangular $\Delta^2$-networks of degree k and order n are isomorphic.

**Proof:** The claim is trivial for $n=1$. Assume it holds for $n-1$, and let G and $G'$ be two $\Delta^2$-networks of order $n$ and degree k. As $G$ and $G'$ are rectangular $\Delta$-networks, they both admit a decomposition of the form illustrated in Figure 4, where $G_1$, $G_2$, ..., $G_k$ are rectangular $\Delta$-networks of order $n-1$ and degree k, and each output of $G_i$ ($G'_i$) is connected to an input of a switch in the last stage labeled i. Moreover, it outputs connected to the same switch in the last stage have the same label. By Lemma 2, each network $G_i$, $G'_i$ is a $\Delta$-network. It follows that each of them is a rectangular $\Delta^2$-network of order $n-1$ and degree k, and therefore all of them are isomorphic. The claim now follows.

The last theorem is valid in general for $\Delta^2$-networks where each level consists of identical switches.

**Generalizations**

While $\Delta$-networks are very attractive in their simplicity, performance or reliability considerations, sometimes dictate the use of more complex networks. Two strategies can be used to augment a network G, without sacrificing much of its structure:

1. The $d$-replication of G is defined to be the network consisting of $d$ identical distinct copies of G, with the $d$ corresponding source (sink) nodes in each copy identical (see Figure 5).

2. The $d$-dilation of G is defined to be the network obtained from G by replacing each edge by $d$ distinct edges (see Figure 6).

**Performance Analyses**

We shall analyze the performance of rectangular $\Delta$-networks under the assumptions usually used in the literature [Pa2]. We assume that the network is synchronous, so that packets can be sent only at times $t_2$, $2t_2$, $3t_2$, ..., where $t_2$ is the network cycle time. The analysis is based on the following model: Packets are generated at each source node by independent, identically distributed random processes. Each process generates with probability p at cycle t a packet, and sends a generated packet with equal probability to any sink node.

The symmetric structure of $\Delta$-networks implies the following result which is implicitly used in all the performance analyses of these networks.

**Lemma 5.** Let packets be generated at the source nodes of a $\Delta$-network by independent, identically distributed random processes, that uniformly distribute the packets over all the sink nodes. Assume that the routing logic at each switch is "fair", i.e., conflicts are randomly resolved. Then

1. The pattern of packet arrivals at the inputs of the same switch are independent.
2. Packets arriving at an input of a switch are uniformly distributed over the outputs of that switch.

Moreover, if the switches at each level are identical then for each level in the network, the pattern of packet arrivals at the inputs of that level have the same distribution.

**Unbuffered $\Delta$-Networks**

We first consider rectangular $\Delta$-networks built of k x k unbuffered switches. When several packets at the same switch require the same output, a randomly chosen one is forwarded and the remaining packets are deleted. The relevant figure of merit for such networks is the probability $p_t$ of having a packet on an input at the i-th stage of the network. By Lemma 5, it is easily seen that $p_t$ is well defined and satisfies the recurrence relation $p_{t+1} = 1 - (1 - p_t)k^k$ with boundary condition $p_0 = p$, the probability of packet creation at a source node. Patel [Pa1] leaves open the question of how $p_t$ behaves asymptotically. It turns out that for any fixed initial value $p_0$ $p_m = \frac{2k}{(k-1)m} \left(1 - \frac{(k+1)\ln(m)}{m} + O\left(\frac{\ln(m)}{m}\right)\right)$. It is interesting to note that the first and second order terms in this expansion are independent of $p_0$. In particular, in a rectangular $\Delta$-network with $N$ source nodes built of 2 x 2 switches the bandwidth of the network, that is the average number of packets arriving at the other end of the network per cycle is

$$N^2p \ln N = \frac{4N^2(1 - \ln N)}{\ln N} + O\left(\frac{\ln N}{N}\right).$$

Thus, the ratio between the average number of requests generated and the average number of requests successfully transmitted is asymptotically inversely proportional to the number of stages in the network. The same result is valid for $\Delta$-networks built of k x k switches.

The analysis of $d$-replicated rectangular $\Delta$-networks does not present any added difficulties. However, $d$-dilated rectangular $\Delta$-networks are more difficult to analyze: We assume that when $m > d$ packets are competing at a switch for the $d$ lines leading to the same switch at the next stage then $d$ of them are chosen at random and forwarded, and the remaining ones are deleted. Assume that a message is issued at each
input line at each cycle. We give the following recurrence for \( p_m \), the probability that a message is not deleted in the first \( m \) stages of a \( d \)-dilated rectangular \( \Delta \)-network of degree 2.

Let \( R(4,j) \) be the probability that \( j \) packets are transmitted through \( d \) identical lines leaving a switch at stage 1. Then

\[
R(0,j) = 0, \quad \text{for } j \neq d,
\]
\[
R(0,j) = 1, \quad \text{for } j = d,
\]
\[
R(m+1,j) = \sum_{i=j}^{2d} \binom{M}{i} R(m,r) \cdot R(m,s) \binom{i}{j} 2^{-i},
\]
for \( j < d \), and
\[
R(m+1,d) = \sum_{i=1}^{2d} \binom{M}{i} R(m,r) \cdot R(m,s) \binom{i}{d} 2^{-i}.
\]
Finally, \( p_m = \frac{1}{d} \sum_{j=1}^{2d} R(m,j) \).

A \( d \)-dilation of a rectangular \( \Delta \)-network of degree 2 with \( N \) source buses has \( (N!N)/2 \) \( 2d \times 2d \) switches and \( \binom{\log_2 N}{d} \) \( d \)-dilated lines. A \( d \)-fold replication of a rectangular \( \Delta \)-network of degree 2 also has \( (N!N)/2 \) \( 2d \times 2d \) switches and roughly the same number of lines. When the performance of two such networks is compared it turns out that a dilated \( \Delta \)-network has a higher bandwidth than a replicated network of comparable complexity only for impossibly large values of \( N \): For \( N < 2^{10} \), dilated networks built of \( 2 \times 2 \), \( 4 \times 4 \), \( 8 \times 8 \), or \( 16 \times 16 \) switches are worse than comparable replicated networks.

**BUFFERED DELTA-NETWORKS**

The bandwidth of \( \Delta \)-networks can be improved by using buffers to queue conflicting packets. An accurate analysis of the performance of buffered \( \Delta \)-networks does not seem tractable. Several authors have analyzed the performance with buffers of length one and performed simulations for larger buffers (see [DJ] and references therein). We present here a formula that seems to yield a good approximation for the performance with large buffers and also indicate some tradeoffs suggested by that formula.

Consider a \( k \times k \) switch with unbounded buffers. Let \( t_c \) be the cycle time of the switch, that is the interval between successive packet arrivals; let \( t_r \) be the transit time of a packet from one switch to the next one, when the buffers on its path are empty (\( t_r > t_c \)). In general the transit time of a packet \( P \) through a switch is \( t_r + N \cdot t_c \), where \( N \) is the number of packets with the same destination that arrived before \( P \), or arrived at the same time as \( P \), but were transmitted before \( P \) (we assume that the order of transmission of packets arriving at the same cycle is random). If at each cycle a packet arrives on each input with probability \( p \), then

\[ m_{av} = \frac{(1-1/k)p}{2(1-p)}. \]

Thus the average transit time of a packet through a \( k \times k \) switch is

\[ t_k = t_r + t_c \cdot \frac{(1-1/k)p}{2(1-p)}. \]

This suggests a formula of the form

\[ T(k,N) = \log_2 N \cdot t_k \]

for the average transit time through the \( \log_2 N \) stages of a rectangular \( \Delta \)-network of degree \( k \) with \( N \) source nodes. The following table compares the values predicted by this formula with values obtained through a simulation of six stages of a network built of 2\( \times 2 \) switches, each containing two buffers of size eight, where \( t_c = t_r = 1 \).

<table>
<thead>
<tr>
<th>Analysis</th>
<th>0.2</th>
<th>0.4</th>
<th>0.6</th>
<th>0.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packets per cycle</td>
<td>1.063</td>
<td>1.167</td>
<td>1.375</td>
<td>2.265</td>
</tr>
<tr>
<td>Delay per stage</td>
<td>0.200</td>
<td>0.400</td>
<td>0.600</td>
<td>0.795</td>
</tr>
</tbody>
</table>

As we see, the predicted delays are in good agreement with the simulations. Besides statistical error, there are two reasons for the discrepancies between the two. Firstly, buffers have only finite size. However, the close agreement between the number of transmissions predicted and simulated indicates that packets were seldom refused by the network, so that limited buffer size is not a significant factor for the loads considered. Secondly, the distribution of the packet arrivals at the second and the subsequent stages is not time independent anymore. A clustering effect occurs, which tends to increase the average delays. Indeed, delays do seem to increase at successive stages, but not significantly.

It is possible to build \( \Delta \)-networks with different performances, by varying the degree of the switches. If each switch is implemented on one chip and the chip performance is pin limited, then without changing the transit time it is possible to increase the number of (logical) lines per switch by a factor of \( d \) while increasing the cycle time by the same factor \( t_c \). Thus, if the cycle time for a \( 2 \times 2 \) switch is \( t_c \) and the transit time is \( t_r \), the cycle time for a \( k \times k \) switch using the same technology is \( T_k = k \cdot t_c \). The average number of packets per cycle is \( P = k \cdot p / 2 \). Therefore, using the previous formula, one obtains that the average transit time through a network built of \( k \times k \) switches is
\[ T(k,N) = \log_N^k (t_T + t_c (1/(k-1)p + (t_c - t_c) \] 
\[ = \log_N^k (t_T + t_c (k-1)p + (k-1)t_c) \]

The last term accounts for the pipe setting delay. We have used this formula to compare the performance of networks built of \( 2^k \times 2^k \) switches, \( k = 1, 2, \ldots \) assuming that \( t_c = t_T \). Figure 7 indicates the domain where each type of switch yields the best performance. Larger switches perform better for low traffic intensities and large networks. Note however that a network built of \( 2 \times 2 \) switches has the same number of switches as a \( k \times k \) replication of a network built of \( 2^k \times 2^k \) switches. When the performance of networks with identical number of switches is compared we get another picture, which is illustrated in Figure 8. Networks with larger switches are capable of supporting higher traffic intensities. In particular, two networks built of \( 4 \times 4 \) switches have the same number of switches, and always outperform one network built of \( 2 \times 2 \) switches.

REFERENCES


Two topologically nonequivalent \( \Delta \)-networks of degree 2 and order 3. 
Figure 3.

Decomposition of reverse \( \Delta \)-network. 
Figure 4.

2-replication of (degree 2 and order 2) \( \Delta \)-network. 
Figure 5.

2-dilation of (degree 2 and order 2) \( \Delta \)-network. 
Figure 6.

\[
\begin{align*}
\log(\text{number of input ports}) & \quad \text{av. number of messages per cycle} \\
\end{align*}
\]

Domain where each configuration gives best performance. 
One network. 
Figure 7.

Replicated networks. 
Figure 8.