Nonblocking Networks for Fast Packet Switching

Riccardo Melen and Jonathan S. Turner
Computer and Communications Research Center
Washington University, St. Louis

Abstract

We define and study an extension of the classical theory of nonblocking networks that is applicable to multirate circuit and fast packet/ATM switching systems. We determine conditions under which the Clos, Cantor and Beneš networks are strictly nonblocking. We also determine conditions under which the Beneš network and variants of the Cantor and Clos networks are rearrangeable. We find that strictly nonblocking operation can be obtained for multirate traffic with essentially the same complexity as in the classical context.

1. Introduction

In this paper we introduce a generalization of the classical theory of nonblocking switching networks to model communication systems designed to carry connections with a multiplicity of data rates. The theory of nonblocking networks was motivated by the problem of designing telephone switching systems capable of connecting any pair of idle terminals, under arbitrary traffic conditions. From the start, it was recognized that crossbar switches with n terminals and n^2 crosspoints could achieve nonblocking behavior, only at a prohibitive cost in large systems. In 1953, Charles Clos [4] published a seminal paper giving constructions for a class of nonblocking networks with far fewer crosspoints, providing much of the initial impetus for the theory that has since been developed by Beneš [1], Pippenger [11] and many others [3,6,8].

The original theory was developed to model electromechanical switching systems in which both the external links connecting switches and the internal links within them were at any one time dedicated to a single telephone conversation. During the 1960's and 1970's technological advances led to digital switching systems in which information was carried in a multiplexed format, with many conversations time-sharing a single link. While this was a major technological change, its impact on the theory of nonblocking networks was slight, because the new systems could be readily cast in the existing model. The primary impact was that the the traditional complexity measure of crosspoint count had a less direct relation to cost than in the older technology.

During the last ten years, there has been growing interest in communication systems that are capable of serving applications with widely varying characteristics. In particular, such systems are being to designed to support connections with arbitrary data rates, over a range from a few bits per second to hundreds of megabits per second [5,7,14]. These systems also carry information in multiplexed format, but in contrast to earlier systems, each connection can consume an arbitrary fraction of the bandwidth of the link carrying it. Typically, the information is carried in the form of independent blocks, called *packets* or *cells* which contain control information, identifying which of many connections sharing a given link, the packet belongs to. One way to operate such systems is to select for each connection, a path through the switching system to be used by all packets belonging to that connection. When selecting a path it is important to ensure that the available bandwidth on all selected links is sufficient to carry the connection. This leads to a natural generalization of the classical theory of nonblocking networks, which we explore in this paper. Note that such networks can also be operated with packets from a given connection taking different paths; reference [15] analyzes the worst-case loading in networks operated

Riccardo Melen is with Centro Stude E Laboratori Telecomunicazioni (CSELT), Torino, Italy and his work has been supported in part by Associazione Elettrotecnica ed Elettronica Italiana, Milano, Italy. This work was done while on leave at Washington University.

Jonathan Turner is with the Computer Science Department, Washington University, St. Louis, MO 63130. His work has been supported by the National Science Foundation (grant DCI 8600947), Bell Communications Research, Bell Northern Research, Italtel SIT and NEC.

CH2702-9/89/0000/0548$01.00 © 1989 IEEE

in this fashion. The drawback of this approach is that it makes it possible for packets in a given connection to pass one another, causing them to arrive at their destination out of sequence.

In Section 2, we define our model of nonblocking multirate networks in detail. Section 3 contains results on strictly nonblocking networks, in particular showing the conditions that must be placed on the networks of Clos and Cantor in order to obtain nonblocking operation in the presence of multirate traffic. We also describe two variants on the Clos and Cantor network that are wide-sense nonblocking in the multirate environment. Section 4 gives results on rearrangeably nonblocking networks, in particular deriving conditions for which the networks of Beneš and Cantor are rearrangeable.

2. Preliminaries

We denote a network N by a quadruple (S, L, I, O), where S is a set of vertices, called *switches*, L is a set of arcs called *links*, I is a set of *input terminals* and O a set of *output terminals*. Each link is an ordered pair (x, y) where $x \in I \cup S$ and $y \in O \cup S$. We require that each input and output terminal appear in exactly one link. Links that include an input terminal are called input links or simply inputs. Those including output terminals are called outputs. The remaining links are called *internal* links. A network with n inputs and m outputs is referred to as an (n, m)-network. An (n, n)-network is also called an n-network.

We consider only networks that can be divided into a sequence of *stages*. We say that the input vertices are in stage 0 and for $i > 0$, a vertex v is in stage i if for all links (u, v), u is in stage $i-1$. An link (u, v) is said to be in stage i if u is in stage i. In the networks we consider, all output terminals are in the same stage, and no other vertices are in this stage. When we refer to a k stage network, we generally neglect the stages containing the input and output vertices.

There are two basic components from which we construct networks. The first is the m input n output crossbar, denoted $X_{m,n}$. If σ is a permutation on $\{0 \ldots, n-1\}$, we also let σ denote the network (S, L, I, O) where $I = \{u_0, \ldots, u_{n-1}\}$, $O = \{v_0, \ldots, v_{n-1}\}$, $S = \emptyset$ and $L = \{(u_i, v_{\sigma(i)}) \mid 0 \leq i \leq n-1\}$. If d_1 and d_2 are positive integers, we define τ_{d_1,d_2} to be the permutation on $\{0, \ldots, d_1 d_2 - 1\}$ satisfying $\tau_{d_1,d_2}(jd_1 + i) = id_2 + j$ for $0 \leq i \leq d_1 - 1$ and $0 \leq j \leq d_2 - 1$.

Networks are constructed using several basic operations. The *concatenation* of two networks N_1 and N_2 is denoted $N_1; N_2$ and is obtained by identifying output link i of N_1 with input link i of N_2. This operation effectively deletes the output terminals of N_1 and the input terminals of N_2. We require of course that the number of outputs of N_1 match the number of inputs of N_2.

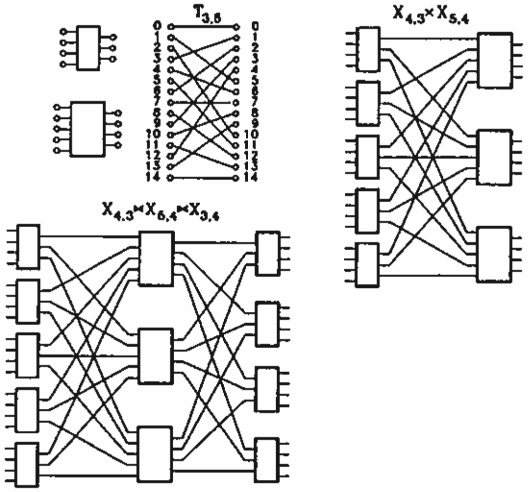

Figure 1: Network Construction Operations

The *reverse* of a network N is the network obtained by exchanging inputs and outputs and reversing the directions of all links and is denoted by N'.

If i is a positive integer and N is an (n, m)-network, then $i \cdot N$ denotes the network obtained by taking i copies of N, without interconnecting them. Inputs and outputs to N are numbered in the obvious way, with the first copy receiving inputs $0, \ldots, n-1$ and outputs $0, \ldots, m-1$ and so forth.

Let N_1 be a network with n_1 outputs and N_2 be a network having n_2 inputs. The *product* of N_1 and N_2 is denoted $N_1 \times N_2$ and is defined as

$$(n_2 \cdot N_1); \tau_{n_1,n_2}; (n_1 \cdot N_2).$$

Informally, the product is obtained by taking n_2 copies of N_1 and connecting them to n_1 copies of N_2 with a single link joining each pair of subnetworks.

We also define a *three-fold product* which we denote with the symbol $\bowtie$. If N_1 has n_1 outputs, N_2 has n_2 inputs and n_3 outputs and N_3 has n_1 inputs, the product $N_1 \bowtie N_2 \bowtie N_3$ is defined as

$$(n_2 \cdot N_1); \tau_{n_1,n_2}; (n_1 \cdot N_2); \tau_{n_3,n_1}; (n_3 \cdot N_3)$$

These definitions are illustrated in Figure 1.

A *connection request* for a network N is a pair (x, y, ω) where x is an input, y an output and $0 \leq \omega \leq 1$. We refer to ω as the *weight* of the connection and it represents the bandwidth required by the connection. A *route* is a path joining an input to an output together with a weight. A route *satisfies* a request (x, y, ω) if it connects x to y and has weight ω.

A set of connection requests is said to be *compatible* if for all inputs and outputs x, the sum of the weights of all connections involving x is ≤ 1. A set of routes is *compatible* if for all links ℓ the sum of the weights of all routes involving ℓ is ≤ 1. A *state* of a network is a set of mutually compatible routes. If we are attempting to add a connection (x, y, ω) to a network in a given state, we say that a vertex u is *accessible* from x if there is path from x to u, all of whose links have a weight of no more than $1 - \omega$.

A network is said to be *rearrangeably nonblocking* (or simply *rearrangeable*) if for every set C of compatible connections, there exists a state that realizes C. A network is *strictly nonblocking* if for every state S, realizing a set of connections C, and every connection c compatible with C, there exists a route r that realizes c and is compatible with S. For strictly nonblocking networks, one can choose routes arbitrarily and always be guaranteed that any new connections can be satisfied without rearrangements. We say that a network is *wide-sense nonblocking* if there exists a routing algorithm, for which the network never blocks; that is, for an arbitrary sequence of connection and disconnection requests, we can avoid blocking if routes are selected using the appropriate routing algorithm and disconnection requests are performed by deleting routes.

Sometimes, improved performance can be obtained by placing constraints on the traffic imposed on a network. We will consider two such constraints. First, we restrict the weights of connections to the the interval $[b, B]$. We also limit the sum of the weights of connections involving an input or output x to β. Note that $0 \leq b \leq B \leq \beta \leq 1$. We say a network is strictly nonblocking for particular values of b, B and β if for all sets of connections for which the connection weights are in $[b, B]$ and the input/output weight is β, the network cannot block. The definitions of rearrangeably nonblocking and wide-sense nonblocking networks are extended similarly. The practical effect of a restriction on β is to require that a network's internal data paths operate at a higher speed than the external transmission facilities connecting switching systems, a common technique in the design of high speed systems. The reciprocal of β is commonly referred to as the *speed advantage* for a system.

Two particular choices of parameters are of special interest. We refer to the traffic condition characterized by $B = \beta$, $b = 0$ as unrestricted packet switching (UPS), and the condition $B = b = \beta = 1$ as pure circuit switching (CS). Since the CS case is a special case of the multirate case, we can expect solutions to the general problem to be at least as costly as the CS case and that theorems for the multirate case should include known results for the CS case.

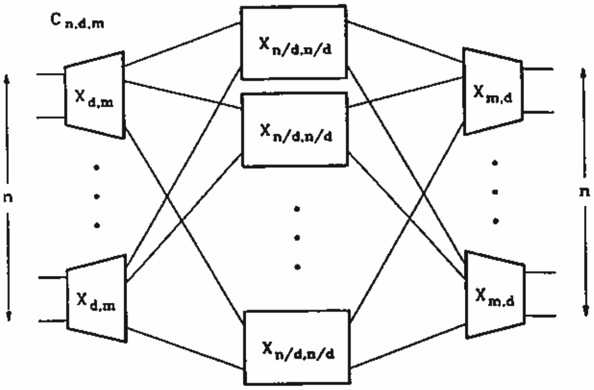

Figure 2: Clos Network

The classical complexity measure for switching networks is the crosspoint count n_C. In our graph model, this can be taken as the sum of the products of the number incoming links and outgoing links for all switches. While the crosspoint count is an appropriate measure for electromechanical switching systems and remains useful, it doesn't give an adequate indication of cost when switching systems are constructed from custom integrated circuits in which input/output constraints at the chip level limit the amount of circuitry that can be placed on a given package. Consequently, we also find it useful to include the *package count* n_P as an additional complexity measure with the understanding that the number of inputs and outputs per package is limited to 2δ. Typical values of δ would be in the range 30–50.

When comparing multirate networks, we also need to take into account the effect of different values of β that may be required by the different networks in order to allow them to achieve comparable performance. We do this by assuming that the speed advantage implied by a given value of β is obtained by providing parallelism in the data paths. This makes the complexity of networks inversely proportional to β.

3. Strictly Nonblocking Networks

A three stage Clos [4] network with n input and output vertices is denoted by $C_{n,d,m}$, where d and m are parameters, and is defined by $C_{n,d,m} = X_{d,m} \bowtie X_{n/d,n/d} \bowtie X_{m,d}$ (see Figure 2). Note that $n_C = (mn/\beta)(2+n/d^2)$. To determine the package count, we must partition the large crossbars in the network into smaller portions that meet the pin constraints. Note that at most δ^2 crosspoints can be placed in a single package with 2δ signal pins, so we take n_P to be n_C/δ^2, effectively assuming an ideal situation in which d, m and n/d are multiples of δ so that no fragmentation occurs.

The standard reasoning to determine the nonblocking condition for the Clos network (see [4]) can be extended in a straightforward manner, yielding the following theorem.

THEOREM 3.1. *The Clos network $C_{n,d,m}$ is strictly nonblocking if*

$$m > 2 \max_{b \leq \omega \leq B} \left\lfloor \frac{\beta d - \omega}{s(\omega)} \right\rfloor$$

where $s(\omega) = \max\{1-\omega, b\}$.

Proof. Suppose we wish to add a connection (x, y, γ) to an arbitrary state. Let u be the stage 1 vertex adjacent to x and note that the sum of the weights on all links out of u is at most $\beta(d-1)+(\beta-\gamma) = \beta d - \gamma$. Consequently, the number of links out of u that carry a weight of more than $(1-\gamma)$ is $\leq \lfloor(\beta d - \gamma)/s(\gamma)\rfloor$, and hence the number of inaccessible middle stage vertices is

$$\leq \left\lfloor \frac{\beta d - \gamma}{s(\gamma)} \right\rfloor \leq \max_{b \leq \omega \leq B} \left\lfloor \frac{\beta d - \omega}{s(\omega)} \right\rfloor < m/2$$

That is, less than half the middle stage vertices are inaccessible from x. By a similar argument, less than half the middle stage vertices are inaccessible from y, implying that there is at least one middle stage vertex accessible to both. □

Let us examine some special cases of interest. If we let $b = B = \beta = 1$, the effect is to operate the network in CS mode and the theorem states that we get nonblocking operation when $m \geq 2d-1$, as is well-known. In the UPS case, the condition on m becomes $m > 2(\beta/(1-\beta))(d-1)$. So $m = 2d-1$ is sufficient here also if $\beta = 1/2$.

For the UPS case, if we choose $d = \sqrt{n/2}$ and $m = 1+2(\beta/(1-\beta))(d-1)$ the crosspoint count of the Clos network becomes

$$\frac{4}{1-\beta}\left[\sqrt{2}n^{3/2} - 2n\right] + 4n/\beta$$

Notice that the complexity becomes unbounded if β is either too close to 0 or too close to 1. Our next result provides a lower bound on the complexity of strictly nonblocking networks when β is unrestricted.

THEOREM 3.2. *Any (m,n)-network that is strictly nonblocking for traffic with $b = 0$ and $B = \beta = 1$ must have at least mn crosspoints.*

Proof. Consider any pair of inputs and outputs x and y. If for each path in the network from x to y there is some link ℓ that is on a path from u to v where $u \neq x$ and $v \neq y$, then the network is not strictly nonblocking, since in this case every path from x to y may contain a link with nonzero weight, which is sufficient to block a connection $(x, y, 1)$. Consequently, there must be at least one crosspoint that can be used only to connect x to y and hence there are at least mn crosspoints. □

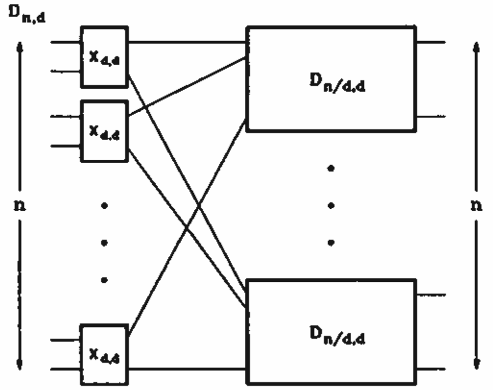

Figure 3: Delta Network

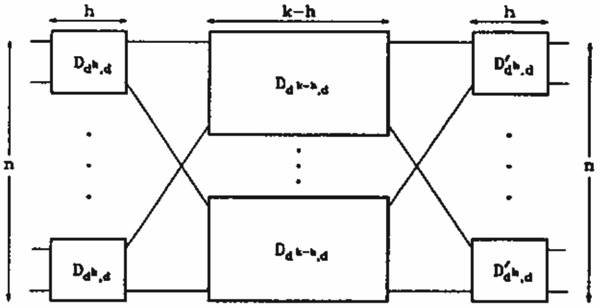

Figure 4: Extended Delta Network

Theorem 3.2 tells us that we can obtain subquadratic complexity and strictly non-blocking operation, only if we restrict the traffic. Note that Theorem 3.2 leaves open the possibility of rearrangeable or wide-sense nonblocking networks of less than quadratic complexity. In fact, using Theorem 3.1, we can construct a wide-sense nonblocking network for unrestricted traffic by placing two Clos networks in parallel and segregating connections in the two networks based on weight. In particular if we let $m = 4d-1$, the network $X_{1,2} \bowtie C_{n,d,m} \bowtie X_{2,1}$ is wide-sense nonblocking if all connections with weight $\leq 1/2$ are routed through one of the Clos subnetworks and all the connections with weight $> 1/2$ are routed through the other. The complexity of this network is $16\sqrt{2}n^{3/2} - 4n$ or roughly four times that of the strictly nonblocking network for the circuit switching case.

The *delta network* [10] $D_{n,d}$ is defined by

$$D_{d,d} = X_{d,d} \qquad D_{n,d} = X_{d,d} \times D_{n/d,d}$$

and illustrated in Figure 3. Note that $k = \log_d n$ must be an integer. The delta network has k stages and provides exactly one path between each input/output pair. More flexible networks can be obtained by adding additional stages of switching. We define the *extended delta network* $D^*_{n,d,h}$ by

$$D^*_{n,d,h} = D_{d^h,d} \bowtie D_{d^{k-h},d} \bowtie D'_{d^h,d}$$

(see Figure 4). An equivalent definition is

$$D^*_{n,d,0} = D_{n,d} \quad D^*_{n,d,h} = X_{d,d} \bowtie D^*_{n/d,d,h-1} \bowtie X_{d,d}$$

Between each input/output pair there are d^h different paths, giving greater routing flexibility than the ordinary delta networks. A Beneš network [1], $B_{n,d}$ is equivalent to $D^*_{n,d,k-1}$ where $k = \log_d n$. For the extended delta network, $n_C = dn(h+k)$ and we take $n_P = n_C/\delta^2$ for $d \geq \delta$ and $n_P = n_C/d\delta \log_d \delta$ for $d \leq \delta$, again assuming an ideal situation in which no fragmentation occurs.

THEOREM 3.3. *The extended delta network $D_{n,d,h}$ is strictly nonblocking if*

$$\frac{\beta}{s(B)} \leq \left[\frac{d^{h-1}}{\lceil d^h/2 \rceil}\left(1+(d-1)h+d^{\lceil (k-h+1)/2 \rceil}-d\right)\right]^{-1}$$

Proof. Let $r = \lfloor (k+h)/2 \rfloor$ and suppose we wish to add a connection (x, y, ω) to an arbitrary state. Note that there are d^h links in stage r that lie on paths from x to y. We will show that at most $\lceil d^h/2 \rceil$ of these links are inaccessible from x if the inequality in the statement is satisfied. By a symmetric argument, at most $\lceil d^h/2 \rceil$ of the links in stage $h+k-r$ that lie on x-y paths are inaccessible from y. Consequently, there must be at least one available path from x to y.

Define W_i to be the set of all links (u, v) in stage i, for which u is accessible from x, but v is not. Define λ_i to be the sum of the weights on all links in W_i and note that $\lambda_i \geq |W_i| s(\omega)$. The number of links in stage r that are not accessible from x is given by

$$\begin{aligned}
& \sum_{i=1}^{h} d^{h-i}|W_i| + \sum_{i=h+1}^{r} |W_i| \\
\leq\ & \frac{1}{s(\omega)}\left[\sum_{i=1}^{h} d^{h-i}\lambda_i + \sum_{i=h+1}^{r} \lambda_i\right] \\
<\ & \frac{1}{s(B)}\left[\beta d^{h-1} + \sum_{i=1}^{h} d^{h-i}(d^i - d^{i-1})\beta \right. \\
& \qquad \left. + \sum_{i=h+1}^{r} (d^i - d^{i-1})\beta\right] \\
=\ & \frac{\beta}{s(B)} d^{h-1}\left[1+(d-1)h+d^{r-h+1}-d\right] \\
=\ & \frac{\beta}{s(B)} d^{h-1}\left[1+(d-1)h+d^{\lceil (k-h+1)/2 \rceil}-d\right] \\
\leq\ & \lceil d^h/2 \rceil
\end{aligned}$$

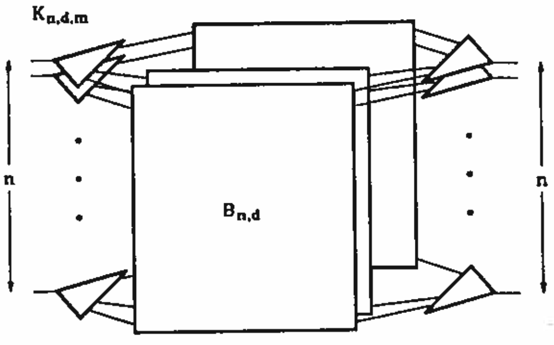

Figure 5: Cantor Network

By the argument above, the theorem follows. □

The following corollaries follow easily from the theorem by substituting the appropriate values of h.

COROLLARY 3.1. *The delta network $D_{n,d}$ is strictly nonblocking if*

$$\frac{\beta}{s(B)} \leq d^{-\lfloor k/2 \rfloor}$$

COROLLARY 3.2. *The Beneš network $B_{n,d}$ is strictly nonblocking if*

$$\frac{\beta}{s(B)} \leq \left[\frac{2}{d}(1+(d-1)\log_d(n/d))\right]^{-1}$$

From this corollary it follows that for networks with $b = 0$ and $B = \beta$, an r stage Beneš network is strictly nonblocking if it has a speed advantage of $r - (2/d)\log_d(n/d)$. So for example, the five stage Beneš network with $d = 32$ and $n = 2^{15}$ is strictly nonblocking if it has speed advantage of 4.875.

The Cantor network $K_{n,d,m} = X_{1,m} \bowtie B_{n,d} \bowtie X_{m,1}$ and is shown in Figure 5 [3]. The next theorem captures the condition on m required to make the Cantor network strictly nonblocking.

THEOREM 3.4. *The Cantor network $K_{n,d,m}$ is strictly nonblocking if*

$$m \geq \frac{2\beta}{d\, s(B)}(1+(d-1)\log_d(n/d))$$

The proof of this theorem is similar to the one for the previous theorem. When we apply it to the CS case for $d = 2$, we find that the condition on m reduces to $m \geq \log_2 n$, as is well known. For the UPS case with $d = 2$, we have $m \geq 2(\beta/(1-\beta))\log_2 n$; that is, we again need a speed advantage of two to match the value of m needed in the CS case.

We can construct wide-sense nonblocking networks for $\beta = 1$ and $b = 0$ by increasing m. We divide the connections into two subsets, with all connections of weight $\leq 1/2$ segregated from those with weight $> 1/2$. Applying Theorem 3.4 we find that $m \geq (8/d)(1 + (d-1)\log_d(n/d))$ is sufficient. That is, the complexity is four times that required for strictly nonblocking operation in the circuit switching case.

4. Rearrangeably Nonblocking Networks

Although in most applications of switching networks it is not practical to operate networks rearrangeably, the property of rearrangeability is important nonetheless, because it implies a topological richness that leads to low blocking probabilities even when the network is not operated in a rearrangeable fashion. In this section, we determine conditions under which the Beneš, Cantor and Clos networks are rearrangeable for multirate traffic.

A d-ary Beneš network [1], can be defined recursively as follows: $B_{d,d} = X_{d,d}$ and $B_{n,d} = X_{d,d} \bowtie B_{n/d,d} \bowtie X_{d,d}$. The Beneš network is rearrangeable in the CS case [1] and efficient algorithms exist to reconfigure it [9]. We start by reviewing a proof of rearrangeability for the CS case, as we will be extending the technique for this case to the multirate situation.

Consider a set of connections $C = \{c_1, \ldots, c_r\}$ for $B_{n,d}$, where $c_i = \{x_i, y_i, 1\}$ and there is at most one connection for each input and output port. The recursive structure of the network allows us to decompose the routing problem into a set of subproblems, corresponding to each of the stages in the recursion. The top level problem consists of selecting, for each connection, one of the d subnetworks $B_{n/d,d}$ to route through. Given a solution to the top level problem, we can solve the routing problems for the d subnetworks independently. We can solve the top level problem most readily by reformulating it as a graph coloring problem. To do this, we define the connection graph $G_C = (V_C, E_C)$ for C as follows.

$$\begin{aligned} V_C &= \{u_j, v_j \,|\, 0 \leq j < n/d\} \\ E_C &= \{\{u_{\lfloor x_i/d \rfloor}, v_{\lfloor y_i/d \rfloor}\} \,|\, 1 \leq i \leq r\} \end{aligned}$$

To solve the top level routing problem, we color the edges of G_C with colors $\{0, \ldots, d-1\}$ so that no two edges with a common endpoint share the same color. The colors assigned to the edges correspond to the subnetwork through which the connection must be routed. Because G_C is a bipartite multigraph with maximum vertex degree d, it is always possible to find an appropriate coloring [2]. In brief, given a partial coloring of G_C, we can color an uncolored edge $\{u, v\}$ as follows. If there is a color $i \in \{0, \ldots, d-1\}$ that is not already in use at both u and v, we use it. Otherwise, we let i be any unused color at u and j be any unused color at v. We then find a maximal *alternating path* from v; that is a longest path with edges colored i or j and v as one of its endpoints. Because the graph is bipartite, the alternating path must end at some vertex other than u or v. Then, we interchange the colors i and j for all edges on the path and use i to color the edge $\{u, v\}$.

To prove results for rearrangeability in the presence of multirate traffic, we must generalize the graph coloring methods used in the CS case. We define a connection graph G_C for a set of connections C as previously, with the addition that each edge is assigned a weight equal to that of the corresponding connection. We say that a connection graph is *(β, d)-permissible* if the edges incident to each vertex can be partitioned into d groups whose weights sum to no more than β. A *legal (β, m)-coloring* of a connection graph is an assignment of colors in $\{0, \ldots, m-1\}$ to each edge so that at each vertex u, the sum of the weights of the edges of any given color is no more than β.

Now, suppose we let $Y = Y_1 \bowtie Y_2 \bowtie Y_3$, where Y_1 is a (d, m)-network, Y_2 is an $(n/d, n/d)$-network and Y_3 is an (m, d)-network and also let $0 \leq \beta_1 \leq \beta_2 \leq 1$. Then if Y_1, Y_2, Y_3 are rearrangeable for connection sets with $\beta \leq \beta_2$ and every (β_1, d)-permissible connection graph for Y has a legal (β_2, m) coloring then Y is rearrangeable for connection sets with $\beta \leq \beta_1$,

Our first use of the coloring method is in the analysis of $B_{n,d}$. We apply it in a recursive fashion. At each stage of the recursion, the value of β may be slightly larger than at the preceding stage. The key to limiting the growth of β is the algorithm used for coloring the connection graph at each stage. We describe that algorithm next.

Let $G_C = (V_C, E_C)$ be an arbitrary connection graph. We construct a new graph G'_C by splitting each vertex u with with $x > d$ edges into $r = \lceil x/d \rceil$ vertices $u_0, \ldots, u_{r-1}$ with the d "heaviest" edges assigned to u_0, the next d heaviest edges assigned to u_1 and so forth. When this operation is complete, we are left with a bipartite graph in which every vertex has at most d edges and we can d-color G'_C as before and then color the edges of G_C in the same way that the corresponding edges are colored in G'_C. We refer to this as the balanced vertex splitting algorithm (BVS) algorithm. We can route a set of connections through $B_{n,d}$ by applying BVS recursively. Our first theorem gives conditions under which this routing is guaranteed not to exceed the capacity of any link in the network.

Theorem 4.1. *The* BVS *algorithm successfully routes* *all sets of connections for $B_{n,d}$ for which*

$$\beta \leq \left[1 + \frac{d-1}{d}(B/\beta)\log_d(n/d)\right]^{-1}$$

Proof. Let G_C be any (β_1, d)-permissible connection graph with maximum edge weight B and $\beta_1 \leq 1 - B(d-1)/d$. We start by showing that the BVS algorithm produces a legal (β_2, d)-coloring for some $\beta_2 \leq \beta_1 + B(d-1)/d$.

Let u be any vertex in G_C. The largest weight that can be associated with any color at u is the sum of the weights of the heaviest edges at each of the corresponding u_i in G'_C. Because of the way u's edges were distributed among the u_is this weight is at most $B + (d\beta_1 - B)/d = \beta_1 - B(d-1)/d$.

Given this, if we route a set of connections through $B_{n,d}$ by recursive application of the BVS algorithm, we will succeed if

$$\beta + \left(\frac{d-1}{d}\right) B \log_d(n/d) \leq 1$$

which is implied by the hypothesis of the theorem. □

As an example, if $n = 2^{15}$, $d = 32$ and $B = \beta$, it suffices to have a speed advantage of 3. We can improve on this result by modifying the BVS algorithm. Because the basic algorithm treats each stage in the recursion completely independently, it can in the worst-case concentrate traffic unnecessarily. The algorithm we consider next attempts to balance the traffic between subnetworks when constructing a coloring. We describe the algorithm only for the case of $d = 2$, although extension to larger values is possible.

Let G_C be a connection graph for $B_{n,2}$. G_C comprises vertices $u_0, \ldots, u_{(n/2)-1}$ corresponding to switches in stage one of $B_{n,2}$ and vertices $v_0, \ldots, v_{(n/2)-1}$ corresponding to switches in stage $2(\log_2 n - 1)$. We have an edge from u_i to v_j corresponding to each connection to be routed between the corresponding switches of $B_{n,2}$. We note that for $0 \leq i < n/4$, the switches corresponding to u_{2i} and u_{2i+1} have the same successors in stage two of $B_{n,2}$. Similarly, the switches in $B_{n,2}$ corresponding to v_{2i} and v_{2i+1} have common predecessors. We say such vertex pairs are *related.*

Let a and b be any pair of related vertices in G_C. The idea behind the modified coloring algorithm is to balance the coloring at a and b so that the total weight associated with each color is more balanced, thus limiting the concentration of traffic in one subnetwork. The technique used to balance the coloring is to constrain it so that when appropriate, the edges of largest weight at a and b are assigned different colors, and hence the corresponding connections are routed through distinct subnetworks. For any vertex v in G_C, let $\omega_0(v) \geq \omega_1(v) \geq \cdots$ be the weights of the edges defined at v, let $W_0(v) = \sum_{i>0} \omega_{2i}$, $W_1(v) = \sum_{i\geq 0} \omega_{2i+1}$ and $W(v) = W_0(v) + W_1(v)$. Also, let $x(v) = W_0(v) - W_1(v)$.

The *modified* BVS *algorithm* proceeds as follows. For each pair of related vertices a and b in G_C, if $x(a) + x(b) > B$, add a dummy vertex z to G_C with edges of weight two connecting it to a and b. We then color this modified graph as in the original BVS algorithm and on completion we simply ignore the added vertices and edges. The effect of adding the dummy vertex is to constrain the coloring at a and b so that the edges of maximum weight are assigned distinct colors. We apply this procedure recursively except that in the last step of the recursion we use the original BVS algorithm.

THEOREM 4.2. *The modified* BVS *algorithm successfully routes all sets of connections for $B_{n,2}$ for which*

$$\beta \leq \left[1 + \frac{1}{4}(B/\beta)\log_2 n\right]^{-1}$$

Proof. Let a and b be related vertices with $\omega_0(a) \geq \omega_0(b)$. Let $z_1 = \max\{W(a), W(b)\}$ and let z_2 be the total weight on edges colored 0 at a and b. If $x(a) + x(b) \leq B$, no dummy vertex is added and we have that

$$\begin{aligned} z_2 &\leq W_0(a) + W_0(b) \\ &\leq (z_1 + x(a))/2 + (z_1 + x(b))/2 \\ &\leq z_1 + B/2 \end{aligned}$$

Similarly, if $x(a) + x(b) \geq B$, a dummy vertex is added and we have that

$$\begin{aligned} z_2 &\leq \omega_0(a) + W_1(a) + W_1(b) \\ &\leq \omega_0(a) + (z_1 - x(a))/2 + (z_1 - x(b))/2 \\ &\leq z_1 + B/2 \end{aligned}$$

Thus, the total weight on a vertex in stage i is at most $2\beta + (i-1)B/2$. In particular, this holds for $i = \log_2 n - 1$. Also note that for an edge (u, v) in stage $j \leq \log_2 n - 2$, the maximum weight is at most B plus half the weight on u. For an edge (u, v) in stage $\log_2 n - 1$, the weight is at most $B/2$ plus the maximum weight at u, since in this last step the original BVS algorithm was used. Consequently, no edge carries a weight greater than $\beta + (B/4)\log_2 n$. □

Theorem 4.2 implies for example that if $\beta = B$, a binary Beneš network with 2^{16} input and output vertices is rearrangeable, if it has a speed advantage of 5. Theorem 4.1, on the other hand gives rearrangeability in this case only with a speed advantage of about 8.5. It turns out that we can obtain a still stronger result by exploiting some additional properties of the original BVS algorithm.

THEOREM 4.3. *The* BVS *algorithm successfully routes all sets of connections for $B_{n,d}$ for which*

$$\beta \leq [\max\{2, \lambda - \ln\lfloor\beta/B\rfloor\}]^{-1}$$

where $\lambda = 2 + \ln\log_d(n/d)$.

So, for example if $d = 32$, $n = 2^{15}$ and $\beta = B$, a speed advantage of 2.7 will suffice for rearrangeability. The proof of Theorem 4.3 requires the following lemmas.

LEMMA 4.1. *Let r be any positive integer. If a set of connections for $B_{n,d}$ is routed by repeated applications of the* BVS *algorithm, no link will carry more than r connections of weight $> \beta/(r+1)$.*

Proof. By induction; the condition is true by definition for the external links. If the assertion holds at a given level of recursion, the connection graph G_C for the next stage will have at most rd edges of weight greater than $\beta/(r+1)$ at any given vertex u. These edges are all incident to $u_0, u_1, \ldots u_{r-1}$ in G'_C, implying that the BVS algorithm will use a single color for at most r of them. □

If ℓ is a link in $B_{n,d}$, we define S_ℓ^j to be the set of links ℓ' in stage j for which there is a path from ℓ' to ℓ. If a given set of connections uses a link ℓ, we refer to one connection of maximum weight as the *primary connection* on ℓ and all others as *secondary connections.* We note that if the BVS algorithm is used to route a set of connections through $B_{n,d}$, then if there are $r+1$ connections of weight $\geq \omega$ on a link $\ell = (u, v)$, there are at least $1 + dr$ connections of weight $\geq \omega$ on the links entering u.

LEMMA 4.2. *Let $0 \leq i \leq \log_d(n/d)$, let ℓ be a stage i link in $B_{n,d}$ carrying connections routed by the* BVS *algorithm and let the connections weights be $\omega_0 \geq \omega_1 \geq \cdots \geq \omega_h$. For $0 \leq t \leq h$ and $0 \leq s \leq \min\{i, t\}$, there are at least $(t-s+1)d^s + sd^{s-1}$ connections of weight $\geq \omega_t$ on the links in S_ℓ^{i-s}.*

Proof. The proof is by induction on s. When $s = 0$, the lemma asserts that there are $t+1$ connections of weight $\geq \omega_t$ which is trivially true. Assume then that the lemma holds for $s-1$; that is, there exist $(t-s+2)d^{s-1} + (s-1)d^{s-2}$ connections of weight $\geq \omega_t$ on the links in S_ℓ^{i-s+1}. Because $|S_\ell^{i-s+1}| = d^{s-1}$, by the pigeon-hole principle, at least $(t-s+1)d^{s-1} + (s-1)d^{s-2}$ of these are secondary connections. This implies that there are at least

$$\begin{aligned} d^{s-1} + d\left[(t-s+1)d^{s-1} + (s-1)d^{s-2}\right] \\ = (t-s+1)d^s + sd^{s-1} \end{aligned}$$

connections of weight $\geq \omega_t$ in S_ℓ^{i-s}. □

Proof of Theorem 4.3. Consider an arbitrary set of connections for $B_{n,d}$ satisfying the bound on β given in the theorem, and assume that the BVS algorithm is used to route the connections. Let ℓ be any link in stage i, where $i \leq \log_d(n/d)$, and let the weights of the connections on ℓ be $\omega_0 \geq \cdots \geq \omega_h$. Let r be the positive integer defined by $\beta/(r+1) < B \leq \beta/r$ (equivalently, $r = \lfloor\beta/B\rfloor$). By Lemma 4.2, S_ℓ^0 carries connections with a total weight of at least

$$\omega_0 + d\omega_1 + d^2\omega_2 + \cdots + d^{i-1}\omega_{i-1} + d^i(\omega_i + \cdots + \omega_h)$$

Since the total weight on S_ℓ^0 is at most βd^i, we have

$$\beta d^i \geq \sum_{j=0}^{i-1} d^j\omega_j + d^i\sum_{j=i}^{h}\omega_j$$

From this and Lemma 4.1, we have that

$$\begin{aligned} \sum_{j=0}^{r-1}\omega_j + \sum_{j=r}^{i-1}\omega_j + \sum_{j=i}^{h}\omega_j \quad &\leq \quad Br + \beta\sum_{j=r}^{i-1}\frac{1}{j+1} + \beta \\ &\leq \quad 2\beta + \beta\sum_{j=r+1}^{\log_d(n/d)}\frac{1}{j} \end{aligned}$$

If $\lfloor\beta/B\rfloor \geq \log_d(n/d)$, the last summation vanishes and we have that the weight on ℓ is $\leq 2\beta$. Otherwise, the weight is bounded by

$$\leq \beta\left(2 + \ln\frac{1}{r}\log_d(n/d)\right) = \beta(\lambda - \ln\lfloor\beta/B\rfloor)$$

So, if β satisfies the bound in the statement of the theorem, the weight on ℓ is no more than one. By a similar argument, the weight on any link in stage j for $j \geq \log_d n$ is at most one. □

The next theorem gives the conditions for rearrangeability for the Cantor network. The proof is omitted.

THEOREM 4.4. *Let $\epsilon > 0$ and $\lfloor\beta/B\rfloor \leq \log_d(n/d)$. $K_{n,d,m}$ is rearrangeable if*

$$\begin{aligned} m \quad \geq \quad & \lceil(1+\epsilon)(\lambda - \ln\lfloor\beta/B\rfloor)\rceil \\ & +2\,(2 + \log_2\lambda + \log_2(B/c)) \end{aligned}$$

where $\lambda = 2 + \ln\log_d(n/d)$ and $c = 1 - \beta\lambda/(1+\epsilon)(\lambda - \ln\lfloor\beta/B\rfloor)$.

The graph coloring methods used to route connections for $B_{n,d}$ can also be applied to networks that "expand" at each level of recursion. Let $C^*_{d,d,m} = X_{d,d}$ and for $n = d^i$, $i > 1$, let $C^*_{n,d,m} = X_{d,m} \bowtie C^*_{n/d,n/d} \bowtie X_{m,d}$. The following theorem gives conditions under which $C^*_{n,d,m}$ is rearrangeable.

THEOREM 4.5. *$C^*_{n,d,m}$ is rearrangeable if*

$$\beta \leq \left[1/\gamma^c + \frac{m-1}{m}\frac{B}{\beta}\frac{1-1/\gamma^c}{1-1/\gamma}\right]^{-1}$$

where $\gamma = m/d$ and $c = \log_d(n/d)$.

Proof. We use the BVS algorithm to route the connections. If we let β_i be the largest resulting weight on a link in stage i for $1 \leq i \leq \log_d(n/d)$, we have

$$\begin{aligned}\beta_i &\leq B + \frac{d\beta_{i-1} - B}{m} = (\beta_{i-1}/\gamma) + \frac{m-1}{m}B \\ &\leq (\beta_0/\gamma^i) + \frac{m-1}{m}B\frac{1-(1/\gamma)^i}{1-1/\gamma} \leq 1\end{aligned}$$

□

So, for example, $C^*_{n,d,2d-1}$ is rearrangeable if $B \leq 1/2$.

5. CONCLUSIONS

Figure 6 compares the complexity of a variety of different networks. The curves give the complexity of the following networks.

- X, an $n \times n$ crossbar.
- C, a three stage Clos network with $\beta = 1/2$ and m just large enough to make it strictly nonblocking.
- K_2 and K_{32}, Cantor networks with $d = 2$ and $d = 32$, $\beta = 1/2$ and m just large enough to make them strictly nonblocking.
- B_2 and B_{32}, Beneš networks with $d = 2$ and $d = 32$ and β chosen to make them strictly nonblocking.
- B'_2 and B'_{32}, Beneš networks with $d = 2$ and $d = 32$ and β chosen to make them rearrangeably nonblocking.
- S, a Batcher sorting network together with a Banyan routing network as used in the Starlite switching system [7].

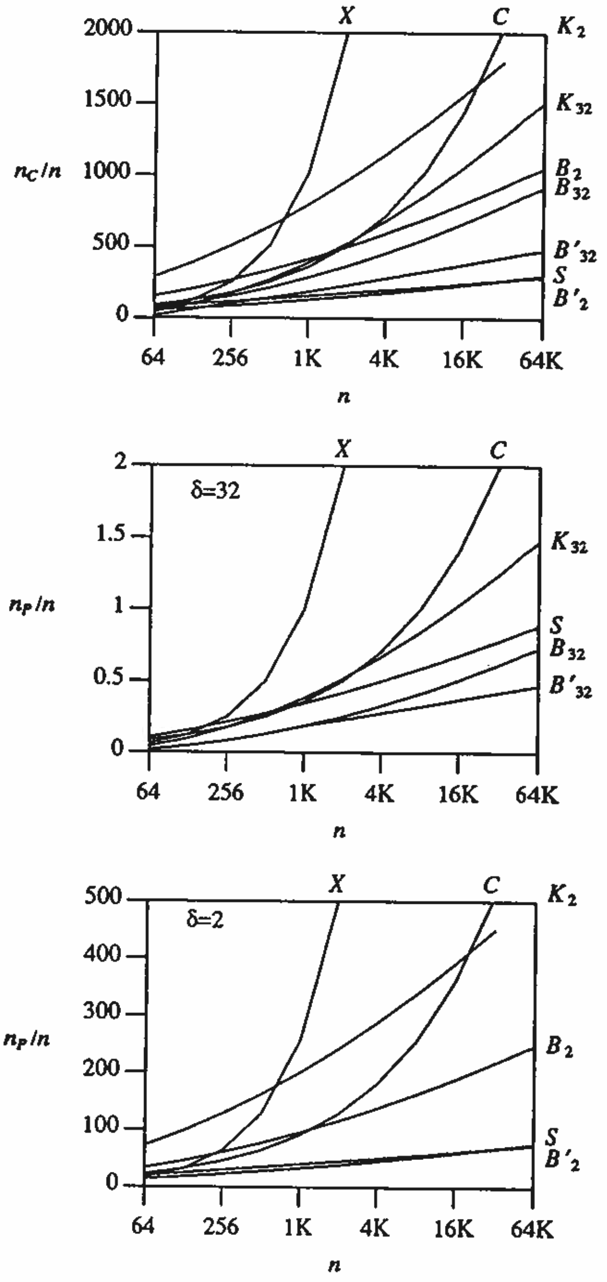

Figure 6: Complexity of Various Networks

The first plot gives the number of crosspoints per port, the second gives the number of packages per port when $\delta = 32$ and the third gives the number of packages per port when $\delta = 2$. In the crosspoint comparison, it's interesting to note the fairly modest difference in complexity attributed to the switch size in the Cantor and Beneš networks. There are two opposing effects at work here. For the Cantor networks, larger switches allows reduction in the value of m and in the Beneš network, reduction in the speed advantage. On the other hand, this is partially offset by the larger number of crosspoints in a 32 port switch compared to a five stage network constructed from two port switches. Note that in the package count comparison, the larger switch gives a reduction of more than two orders of magnitude for most of the networks. For the larger package sizes, the Beneš networks appear most attractive, although the sorting network certainly compares favorably. Notice that at switch sizes of 1024 and under there is no difference in the package counts of the strictly nonblocking and rearrangeable Beneš networks for $\delta = 32$. We believe that this indicates that our bound on β for rearrangeable operation can be improved. We suspect in fact that rearrangeable operation can be achieved with only a constant speed advantage. Also note that at switch sizes of 1024 and under, the difference in package counts between the Clos and rearrangeable Beneš networks is only a factor of two when $\delta = 32$. From an engineering viewpoint, this suggests that other factors may take precedence over complexity considerations.

An important message of these plots is that the traditional complexity measure of crosspoint count can be misleading. The package count is clearly the more useful cost measure for making engineering choices and it differs significantly from the crosspoint measure. One final caution regarding the package counts is that the absolute values can be misleading if used carelessly. These values are normalized so that $\beta = 1$ corresponds to systems with bit-serial data paths. If wider data paths are required for reasons outside those considered here, the package counts shown in Figure 6 must be scaled accordingly.

In this paper, we have introduced what we feel is an important research topic and have given some fundamental results. Our generalization of the classical theory is a natural and interesting one, which has direct application to practical systems now under development in various research laboratories [5,7,14]. There are several directions in which our work may be extended. While our results for strictly nonblocking networks are tight, we believe that our results for rearrangeably nonblocking networks can be improved. Another interesting topic is nonblocking networks for multipoint connections. While this has been considered for space-division networks [6,12], it has not been studied for networks supporting multirate traffic. Another area to consider is determination of blocking probability for multirate networks.

References

[1] Beneš V. E. *"Mathematical Theory of Connecting Networks and Telephone Traffic,"* Academic Press, New York, 1965.

[2] Bondy, J. A. and U. S. R. Murty. *Graph Theory with Applications*, North Holland, New York, 1976.

[3] Cantor, D. G. "On Non-Blocking Switching Networks," *Networks*, vol. 1, 1971, pp. 367–377.

[4] Clos, C. "A Study of Non-blocking Switching Networks," *Bell Syst. Tech. J.*, vol. 32, 3/53, pp. 406–424.

[5] Coudreuse, J. P. and M. Servel "Prelude: An Asynchronous Time-Division Switched Network," *International Communications Conference*, 1987.

[6] Feldman, P., J. Friedman and N. Pippenger "Non-Blocking Networks," *Proceedings of STOC 1986* 5/86, pp. 247–254.

[7] Huang, A. and S. Knauer "Starlite: a Wideband Digital Switch," *Proceedings of Globecom 84*, 12/84, pp. 121–125.

[8] Masson, G. M., Gingher, G. C., Nakamura, S. "A Sampler of Circuit Switching Networks," *Computer*, 6/79, pp. 145–161.

[9] Opferman, D. C. and N. T. Tsao-Wu "On a Class of Rearrangeable Switching Networks, Part I: Control Algorithm," *Bell Syst. Tech. J.*, vol. 50, 1971, pp. 1579–1600.

[10] Patel, J.K. "Performance of Processor-Memory Interconnections for Multiprocessors," *IEEE Transactions on Computers*, vol. C-30, 10/81, pp. 301–310.

[11] Pippenger, N. "Telephone Switching Networks," *Proceedings of Symposia in Applied Mathematics*, vol. 26, 1982, pp. 101–133.

[12] Richards, G. and F. K. Hwang "A Two Stage Rearrangeable Broadcast Switching Network," *IEEE Transactions on Communications*, 10/85, 1025–1035.

[13] Shannon, C. E. "Memory Requirements in a Telephone Exchange," *Bell Syst. Tech. J.*, vol. 29, 1950, pp. 343–349.

[14] Turner, J. S. "Design of a Broadcast Packet Network," *IEEE Transactions on Communications*, 6/88, pp. 734–743.

[15] Turner, J. S. "Fluid Flow Loading Analysis of Packet Switching Networks," Washington University Computer Science Department, WUCS-87-16, 7/87.