BanyanNet—A Bidirectional Equivalent of ShuffleNet

K. Wendy Tang

***Abstract*—A multihop, wavelength division multiplex (WDM)-based network, BanyanNet, is proposed for the realization of terabit lightwave networks. BanyanNet can be considered as a the bidirectional equivalent of the popular ShuffleNet. Exploiting its representation, we developed a fast, decentralized, bidirectional routing algorithm for BanyanNet. The performance of BanyanNet is compared to that of the ShuffleNet and bilayered ShuffleNet. For $N=p^m \times k$ networks, the $p=2$ BanyanNet provides better performance in channel efficiency, total and user throughput than the corresponding ShuffleNet, and offers more flexible network configurations than the bilayered and $p=4$ ShuffleNet.**

***Index Terms*—LAN, MAN, lightwave network, wavelength division multiplex.**

I. Introduction

WITH THE RECENT advances in fiber optics, *lightwave networks* composed of optical fibers have embarked on an important role in telecommunications. The strength of the photonic technology includes an enormous bandwidth, noise immunity, and high security. The bandwidth offered by optical fibers is on the order of terahertz (THz), whereas that of conventional coaxial cable and twisted pair is only on the order of gigahertz or even megahertz. Furthermore, optical fibers are almost immune to noise and have excellent security. They are not affected by electromagnetic interference and are nearly impossible to wiretap without detection. These superior qualities over conventional methods made lightwave networks attractive candidates for large local and metropolitan area networks.

However, exploitation of the vast bandwidth in optical networks has been hindered by the speed of the electro-optic converter—a device converting electrical signals to optical signals and vice versa. These electronic devices can only operate in gigabits per second. Such mismatch in bandwidth between the electronic components and the optical fibers is the main obstacle in the realization of *terabit lightwave networks.* Much research effort has been directed toward resolving this dilemma [1]–[3]. This includes multiple users sharing an optical fiber via time and wavelength (frequency) multiplexing.

While time division multiplexing is limited by electronic speeds, wavelength division multiplexing (WDM) is preferred for large-scale concurrency on a single fiber [4]. There are two classes of WDM-based systems: *single-hop* and *multihop* [5], [6]. Single-hop systems imply that nodes communicate in one hop. Typically, a node has a small number of optical transmitters (lasers) and optical receivers (filters). These optical devices must be wavelength-agile; that is, they are capable of rapidly tuning to the transmission frequency. Also, pretransmission communication must exist to coordinate the transmission time and frequency between transmitters and receivers [7].

The multihop approach, on the other hand, assigns fixed transmission frequency to each communication link, and therefore eliminates the need for pretransmission communications and rapidly tunable transmitters and receivers [6]. Again, each node has a small number of transmitters, transmitting and receiving signals in an assigned and fixed wavelength. This arrangement allows simultaneous transmission among multiple users and thus attains the terabit capacity of the network. For example, the typical bandwidth for the low-loss region in a single-mode optical fiber is about 25–30 THz, and the electronic processing speed is a few Gb/s. In other words, a single fiber can accommodate up to 10^4 electronic-grade channels [6]. As the work *multihop* suggests, a message may be required to route through intermediate nodes, each retransmits the message on a different wavelength until it reaches the destination.

The establishment of an efficient multihop lightwave network relies heavily on the proper assignment of wavelengths to communication links of each node. The goals are to ensure that there is at least one path between any pair of nodes, and that the average and maximum number of hops for a message to reach its destination should be small. Such assignments are based on an interconnection topology. Since this topology is not directly related to the physical connection of nodes, it is referred to as a *virtual topology.* A number of virtual topologies have been proposed [8]. These include: ShuffleNet [9], [10], [7], Hypercube [11], Generalized Hypercube [8], DeBruijn [12], and MSN (Manhattan Street Network) [13]. There are advantages and disadvantages to the different options. A review can be found in [6] and [8]. Of the many options, ShuffleNet is one of the most popular topology [6]. It has been shown that a 64-node ShuffleNet has better performance than the corresponding MSN [14].

The ShuffleNet was first proposed by Acampora *et al.* [9] and later extended and generalized by Hluchyi and Karol [10], [7]. Conceptually, it is a unidirectional, cylindrically connected Omega network [15]. In general, there are $N = p^m \times m$ modes arranged in p^m rows and m columns. Interconnection between adjacent columns is a perfect shuffle [16]. Fig. 1 shows an

Manuscript received May 24, 1993; revised September 12, 1994.
The author is with the Department of Electrical Engineering, SUNY at Stony Brook, Stony Brook, NY 11794-2350.
IEEE Log Number 9405768.

0733-8724/94$04.00 © 1994 IEEE

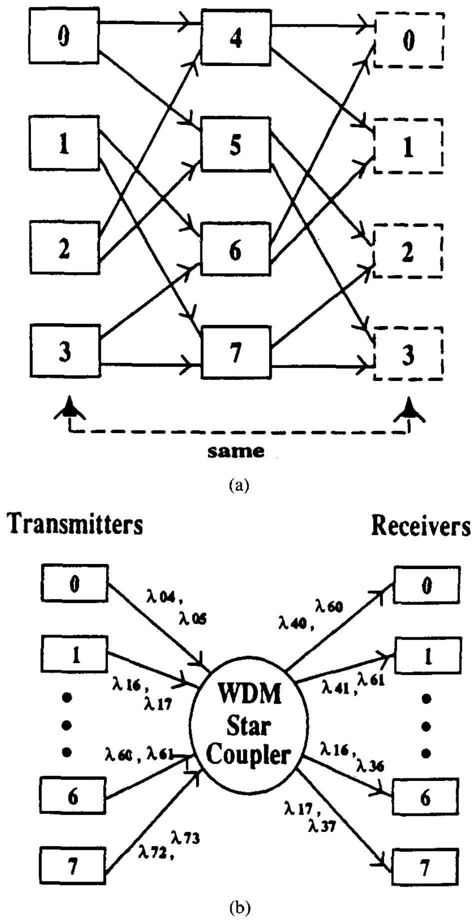


Fig. 1. An 8-node ShuffleNet and its implementation. (a) ShuffleNet. (b) Star implementation.

$N = 2^2 \times 2 = 8$ ShuffleNet. In this case, each node has two transmitters and two receivers, each with a fixed and assigned frequency $(\lambda_{ij}, i, j = 0, \cdots, 7)$. A single transmitter and receiver can also be used at each node if p users in each column are allowed to share the same transmission frequency. In that case, however, multiple access problems and inefficiency are possible [9], [7]. Physically, the network topology can be arbitrary, provided that direct transmission exists between adjacent nodes in the ShuffleNet. Popular topologies in local and metropolitan area networks such as the bus, star, or tree networks are sufficient. As indicated in [8], Fig. 1(b) shows a star implementation of the 8-node ShuffleNet.

One advantage of this unidirectional ShuffleNet is its simple routing algorithm. Since messages usually require multiple hops to get to destinations, the goal of routing is to determine an appropriate outgoing link for each incoming message. A simple, *distributed, self-routing* algorithm that can identify shortest paths based only on address of the destination exists for the unidirectional ShuffleNet [7]. With this algorithm, the maximum distance (in hops) for a message to get to its destination is $2m - 1$ for $N = p^m \times m$ nodes [9]. Using graph terminology, this distance is called the *diameter* of the network [17]. Obviously, a small diameter implies a potentially small communication delay.

While the advantage of the ShuffleNet is its simple routing, its disadvantages include 1) a nonsymmetric node distance and 2) limited number of nodes. Due to ShuffleNet's unidirectional property, distance from node i to j does not equal that from j to i. For example, if node i sends a message to its immediate neighbor, the reply/acknowledge message may take diameter D_S steps. Furthermore, the number of nodes for a ShuffleNet is restricted to $N = p^k \times k$. When p is large, the possible number of nodes becomes limited.

With the growing number of computer users and network size, it is desirable to reduce the diameter and the number of hops by considering bidirectional ShuffleNet. The original representation (layout) of ShuffleNet, however, does not facilitate bidirectional routing. Ayadi *et al.* proposed the use of a bilayered ShuffleNet [18]. In this case, the basic principles are similar to the original ShuffleNet except that stations in each column are also connected to the "mirror" image of stations from the previous column. Like the ShuffleNet, there are $N = p^m \times m$ nodes; but unlike the ShuffleNet, each node has $2p$ neighbors. Fig. 2 shows an $N = 2^2 \times 2$ bilayered ShuffleNet. The advantage of the bilayered ShuffleNet is that by doubling the number of connections, network performance is improved. However, this is achieved at a higher cost of more connections and the disadvantages associated with the original ShuffleNet, namely, 1) a nonsymmetric node distance and 2) limited number of nodes, still remain. Furthermore, no simple bidirectional self-routing algorithm is provided for the bilayered ShuffleNet in [18].

In this paper, we propose the use of a different representation (layout) of the ShuffleNet that facilitates a simple, bidirectional, self-routing algorithm. In the design of multistage interconnection networks for a multiprocessor system, it is known that the SW-Banyan network is topologically equivalent to the ShuffleNet [19]. In other words, the two networks have the same properties and are only different in their representation (layout). However, we observe that the SW-Banyan representation facilitates the design of bidirectional routing. As a result, we propose to use the SW-Banyan network representation to pursue the bidirectional equivalent of the original ShuffleNet. Analogous to the unidirectional ShuffleNet, we call this bidirectional version the *BanyanNet.*

This paper is organized as follows. In Section II, we present the BanyanNet as a bidirectional multihop lightwave network. Diameter analysis and the routing algorithm are discussed in Section III. Section IV evaluates and compares the performance of BanyanNet, ShuffleNet, and the bilayered ShuffleNet. Finally, in Section V we present a summary and conclusions.

II. BANYANNET

Goke and Lipovski proposed a general class of dynamic networks for multiprocessor interconnection, called *Banyan networks* [20], [21]. These networks are essentially made up of superimposed trees. ("Banyan" is the name of a multiply rooted tree in India.) Of the general class of Banyan networks,

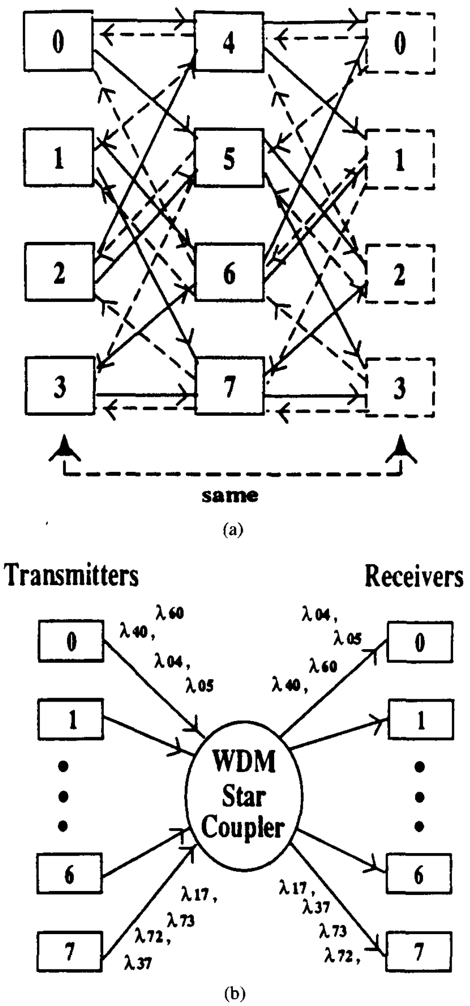


Fig. 2. An 8-node bilayered ShuffleNet and its implementation. (a) Bilayered ShuffleNet. (b) Star implementation.

a subclass called *regular SW-Banyan* is of special interest to us. Mathematically, these networks can be defined as follows.

For an $N = p^m \times k$ network, any node $(x, y), x \in \{0, \cdots, k-1\}$,

$$y = (y_0, \cdots, y_{m-1}) = y_0 p^{m-1} + \cdots + y_{m-1} p^0 \in \{0, \cdots, p^m - 1\},$$

$$(x, y) \sim \begin{cases} ([x+1]_k, y_0, \cdots, y_{r-1}, 0, y_{r+1}, \cdots, y_{m-1}) \\ ([x+1]_k, y_0, \cdots, y_{r-1}, 1, y_{r+1}, \cdots, y_{m-1}) \\ \vdots \\ ([x+1]_k, y_0, \cdots, y_{r-1}, p, y_{r+1}, \cdots, y_{m-1}) \\ ([x-1]_k, y_0, \cdots, y_{r-2}, 0, y_r, \cdots, y_{m-1}) \\ ([x-1]_k, y_0, \cdots, y_{r-2}, 1, y_r, \cdots, y_{m-1}) \\ \vdots \\ ([x+1]_k, y_0, \cdots, y_{r-2}, p, y_r, \cdots, y_{m-1}) \end{cases} \quad (1)$$

where $r = x \pmod m$, $y_i \in \{0, 1, \cdots, p-1\}$, the symbol $\sim$ signifies connections, and $[x+1]_k$ denotes $x + 1 \pmod k$. Similar to the original and bilayered ShuffleNets, we consider modular wraparound connections exist between stage 0 and $k-1$. But unlike the ShuffleNets, we consider bidirectional networks and the number of stages k is not limited to m. Instead, k can be any multiples of m. (Obviously, when k is much larger than m, average number of hops, diameter, and hence propagation delay, becomes unacceptably large. However, our performance analysis in Section IV shows that when k is moderately large, the network has a good total throughput.) The result is a cylindrically and multiply cascaded version of the SW-BanyanNet by Goke and Lipovski [20], [21]. We called this network an $N = p^m \times k$ BanyanNet. These additional flexibilities reduce the diameter and increase network performance such as channel efficiency, network, and user throughputs, as will be discussed in Section IV.

As obvious from the above definition, there are $2p$ communication links at each node. In this paper, we focus on the binary case $p = 2$, although most of our results can be easily extended to the general case. For $p = 2$, there are four connections at each node: *forward straight* and *exchange,* and *reverse straight* and *exchange*. They are defined as

$$(x, y) \sim \begin{cases} ([x+1]_k, y) \\ \quad \text{(forward straight)} \\ ([x+1]_k, y_0, \cdots, \overline{y}_r, \cdots, y_{m-1}) \\ \quad \text{(forward exchange)} \\ ([x-1]_k, y) \\ \quad \text{(reverse straight)} \\ ([x-1]_k, y_0, \cdots, \overline{y}_{r-1}, \cdots, y_{m-1}) \\ \quad \text{(reverse exchange).} \end{cases} \quad (2)$$

Fig. 3(a) shows an $N = 2^2 \times 4$-node BanyanNet, and Fig. 3(b) shows the star implementation of the network. Like the original and bilayered ShuffleNets, the particular topology is of no significance. The BanyanNet topology is used for wavelength assignment. Similar to the bilayered ShuffleNet, each user has four bidirectional links capable of transmitting and receiving signals at four different but fixed wavelengths. These wavelengths are labeled as $\lambda_{ij}, i, j = 0, \cdots, 15$, corresponding to transmitter i and receiver j. For example, consider node 0 is sending a message to node 9. Based on the routing algorithm introduced in the next section, the router at node 0 sends this message out with wavelength λ_{04}. Since node 4 is the only node that can receive signals at this wavelength, the message is transmitted to node 4, which then retransmits the signal with wavelength λ_{49}, and node 9 will be the only user capable of receiving the message. Consequently, the message takes two hops to get to its destination.

As illustrated in this example, the performance of the network is affected by 1) the assignment of wavelengths to the communication links and 2) the routing algorithm that determines the transmitting wavelength of a message. Inappropriate assignment may result in unnecessary long communication delay or even nonexisting paths among users. An ideal assignment ensures that all nodes are connected with the minimum number of hops. Furthermore, to fully exploit a high-speed lightwave network, the routing algorithm needs to provide fast, decentralized decisions. A distributed, self-

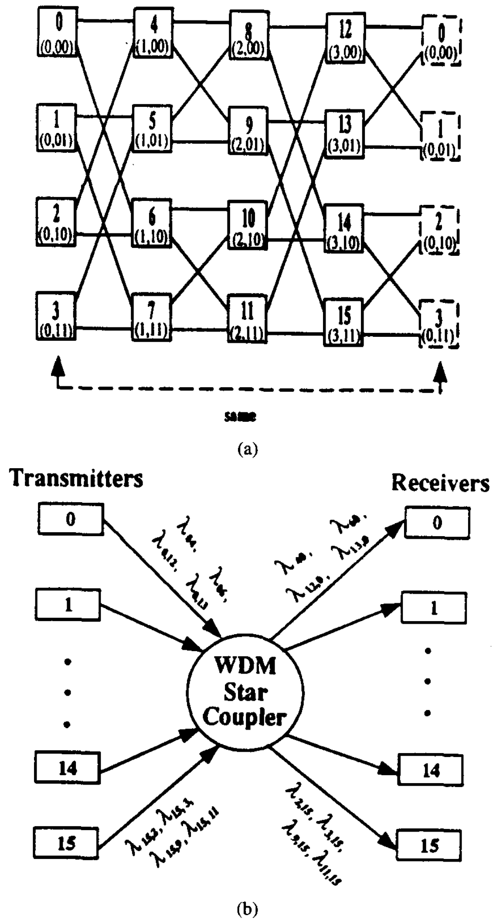


Fig. 3. An $N = 2^2 \times 4$ BanyanNet and its implementation. (a) BanyanNet. (b) Star implementation.

routing scheme based only on addresses of the source and destination is, therefore, highly desirable.

As mentioned earlier, BanyanNet can be considered as a bidirectional equivalent of the original ShuffleNet. Being a bidirectional network, the BanyanNet is capable of providing a wavelength assignment with a smaller average number of hops and hence a smaller propagation delay. More importantly, this representation possesses the advantage that all nodes on the same row are connected. In other words, an $N = 2^m \times k$ BanyanNet can be viewed as composed of 2^m interconnected rings of k nodes. This observation allows us to develop a relatively simple self-routing algorithm based on addresses of the users in the network. The details of this algorithm and the diameter analysis are discussed in next section.

III. Diameter Analysis and Routing

As a direct consequence of bidirectional communications, the diameter of a BanyanNet is much smaller than that of a ShuffleNet. More specifically, the diameter of an $N = 2^m \times k$ (k is a multiple of m) BanyanNet is

$$D_B = \begin{cases} m + \lfloor m/2 \rfloor, & \text{if } k = m \\ \max(2m,\ \lfloor k/2 \rfloor), & \text{if } k > m \end{cases} \tag{3}$$

where $\lfloor k/2 \rfloor$ denotes the largest integer smaller than $k/2$. For $k = m$, the BanyanNet is a bidirectional and cylindrical version of the original SW-Banyan network. In this case, using the original routing algorithm, the distance between nodes on the same column is at most m. The distance between nodes separated by i columns, where $1 \le i \le \lfloor m/2 \rfloor$ is at most $m+i$ because it takes at most m steps to get to a node on the same column as the source and the same ring as the destination, and finally at most another i hops through the ring to the destination. Hence, the diameter is $D_B = m + \lfloor m/2 \rfloor$ for $k = m$ stages.

For $k > m$, again it takes m steps to get to a node m stages from the source. For nodes separated by less than m stage, at most $2m$ hops are needed because m steps are required to traverse to a node on the same ring as the destination but m stages from the source, then at most another m hops through the ring are necessary to arrive at the destination. As for nodes separated by i stages where $m < i \le \lfloor k/2 \rfloor$, their distance is i because $m < i$ hops are required to get to the correct ring m stages from the source and then $i - m$ steps over the ring to the final destination. Hence, the diameter is $D_B = \max(2m,\ \lfloor k/2 \rfloor)$.

For the purpose of comparison, we generalize the original ShuffleNet to $N = p^m \times k$ nodes, where k is a multiple of m. Similar to the original definition, the generalized ShuffleNet has k columns of p^m nodes and adjacent columns are connected by the shuffle permutation. Since Omega networks are topologically equivalent to SW-Banyan networks, the diameter of a unidirectional ShuffleNet is the same as that of a unidirectional BanyanNet. Using a similar argument for the bidirectional BanyanNet, the diameter of an $N = 2^m \times k$ (k is a multiple of m) unidirectional ShuffleNet is

$$D_S = k + m - 1. \tag{4}$$

That is, it takes m hops to traverse to any node m stages from the source. For nodes, separated by i stages, where $m < i \le k - 1$, a total of i hops are needed because the first m hops will get to the destination ring and then another $i - m$ hops to the destination column. For nodes i stages apart, where $0 \le i < m$, a total of $k + i$ hops are needed because it takes k hops to any node in the same column as the source and then another i hops to the final destination. Hence, $D_S = k + m - 1$. Fig. 4 compares the diameter for the BanyanNet and the ShuffleNet for $m = 3, 8$. From (3) and (4) and Fig. 4, it is clear that $D_B < D_S$ except for the trivial case $k = m = 2$. Furthermore, the optimal ratio occurs when $k/m \approx 4$ or 5.

From the definition of BanyanNet in (2), we observe that for any node $(x, y_0, \cdots, y_{m-1})$, the forward exchange connection changes the rth bit and the reverse exchange connection changes the $r' = r - 1$th bit, where $r = x \pmod m$. This observation, coupled with our argument in the diameter analysis facilitated a simple self-routing algorithm. This algorithm is summarized in Table I.

To prevent a message from being shuffled back and forth in the forward and reverse directions, two Boolean parameters—FRD and RVS—are associated with each message. The initial values of these parameters are set

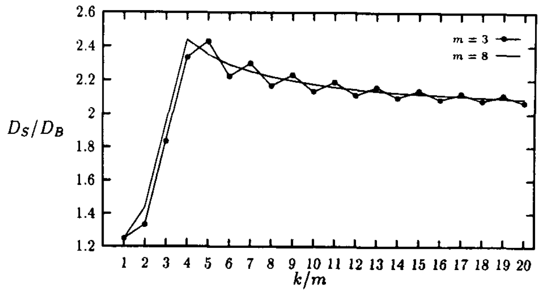


Fig. 4. Diameter comparison of $2^m \times k$ BanyanNet and ShuffleNet.

TABLE I
A ROUTING ALGORITHM FOR $N = 2^m \times k$ BANYANNET

Routing between $(x^s, y_0^s, \ldots, y_{m-1}^s)$ and $(x^d, y_0^d, \ldots, y_{m-1}^d)$.

Evaluate $x = < x^d - x^s >_k$, where $< x >_k = \begin{cases} x, & \text{if } |x| \le \lfloor \frac{k}{2} \rfloor; \\ x - k & \text{if } x > \lfloor \frac{k}{2} \rfloor; \\ x + k & \text{if } x < -\lfloor \frac{k}{2} \rfloor. \end{cases}$

If FRD and RVS are $FALSE$,
$FRD = TRUE$ if $x \ge 0$, else $RVS = TRUE$.

Case 1: $|x| > m$.
If $x > m$, both forward straight and exchange are optimal.
If $x < -m$, both reverse straight and exchange are optimal.

Case 2: $|x| \le m$. Let $r = x^s$, $r' = r - 1 \bmod m$.
Subcase 1: $y^s = y^d$.
If $x > 0$ use forward straight; use reverse straight, otherwise.
Subcase 2: $y^s \ne y^d$ and $FRD = TRUE$.
If $(y_r^s = y_r^d)$, use forward straight; use forward exchange, otherwise.
Subcase 3: $y^s \ne y^d$ and $RVS = TRUE$.
If $(y_{r'}^s = y_{r'}^d)$, use reverse straight; use reverse exchange, otherwise.

according to the relative column distance between the source and destination node. Routing for nodes separated by more than m stages (Case 1) is achieved by first sending the message to any node m stages from the destination. The nature of the original topology guarantees a path of length m between any two nodes separated by m stages. Hence, in this case, both the straight and exchange connections in the appropriate direction (forward or reverse) can be used.

For nodes within m stages apart, we differentiate three subcases. Subcase 1 corresponds to source on the same ring as the destination, and therefore straight connections in the appropriate directions can be used. For Subcases 2 and 3, our goal is to route to an intermediate nodes (x', y') with $y' = y^d$. This is achieved by comparing the rth and the $r' = (r-1)$th bit of y^s and y^d, where $r = x^s \pmod m$. If FRD is true, we consider the rth bit; whereas if RVS is true, we consider the r'th bit. If the rth or r'th bit is different, forward or reverse exchange is used, respectively. Otherwise, straight through connections are used. As examples, Tables II and III illustrate how the algorithm is used to identify paths between nodes $(0, 0)$, $(1, 01)$, and $(3, 11)$ for the $N = 2^2 \times 4$ node BanyanNet in Fig. 3(a). In both cases, a message is generated in step 0, and arrives at destination in 3 steps. The notations $\surd$ and $\times$ denote true and false.

This algorithm provides fast decentralized routing decision. However, it is not optimal in that the shortest path (in hops)

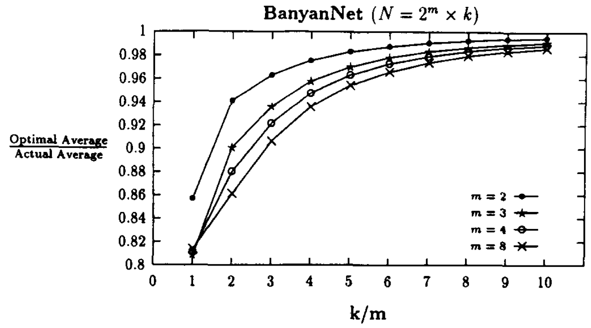


Fig. 5. Routing evaluation for BanyanNet.

between any two nodes is not guaranteed for nodes separated by $\le m$ stages. To evaluate the performance of the algorithm, we implement the algorithm with a computer program. A message is sent from an arbitrary source node, say, node $(0, 0)$ to all other nodes in the network. The path length for each message is recorded. We found that the maximum path length from the algorithm equals the diameter D_B, the optimal upperbound.

Fig. 5 shows the ratio of the *optimal* average to the *actual* average path length. Here optimal average refers to the average length if shortest paths are obtained, whereas actual average refers to the average path length obtained through the algorithm in Table I. Obviously, this ratio is upperbounded by one. When it approaches unity, the algorithm provides path length close to optimal. From this figure, the routing performance increases with decreasing m and increasing k. Fig. 6 depicts the path length distribution for $N = 2^m \times k$ BanyanNet with $m = 8$ and $k = m, 5m$. The y-axis shows the probability of a path with length x. This value is calculated by dividing the total number of paths with length x by $N - 1$, where N is the number of nodes. The curve labeled *Actual* refers to the result from the routing algorithm; whereas the label *Optimal* refers to shortest path distribution. When $k = m$, the actual distribution is shifted toward the right of the optimal distribution, accounting for its higher average length. But when k increases, the actual distribution approaches that of the optimal as in the case of $k = 5m$.

IV. PERFORMANCE EVALUATION AND COMPARISON

In this section, we present the performance of BanyanNet and compare it to that of the original (unidirectional) ShuffleNet and bilayered (bidirectional) ShuffleNet. Analogous to the work in [9], [10] and [18], the performance attributes considered are channel efficiency η, network throughput C, and user throughput c. Assuming the traffic load is uniformly distributed, these attributes are defined as

$$\eta = \frac{1}{E[\text{number of hops}]}$$
$$C = \eta W$$
$$c = \eta \omega$$

where W is the total number of channels in the network and ω is the number of channels per user.

TABLE II
ROUTING STEPS FOR (0, 0) AND (1, 01) IN AN $N = 2^2 \times 4$ BANYANNET

Step	(x^s, y_0^s, y_1^s)	(x^d, y_0^d, y_1^d)	x	r	r'	FRD	RVS	$y^s = y^d$	Link
0	(0,00)	(1,01)	1	0	1	√	×	×	forward straight
1	(1,00)	(1,01)	0	1	0	√	×	×	forward exchange
2	(2,01)	(1,01)	−1	0	1	√	×	√	reverse straight
3	(1,01)	(1,01)	–	–	–	–	–	–	–

TABLE III
ROUTING STEPS FOR (0, 0) AND (3, 11) IN AN $N = 2^2 \times 4$ BANYANNET

Step	(x^s, y_0^s, y_1^s)	(x^d, y_0^d, y_1^d)	x	r	r'	FRD	RVS	$y^s = y^d$	Link
0	(0,00)	(3,11)	−1	0	1	×	√	×	reverse exchange
1	(3,01)	(3,11)	0	1	0	×	√	×	reverse exchange
2	(2,11)	(3,11)	1	0	1	×	√	√	forward straight
3	(1,01)	(3,11)	–	–	–	–	–	–	–

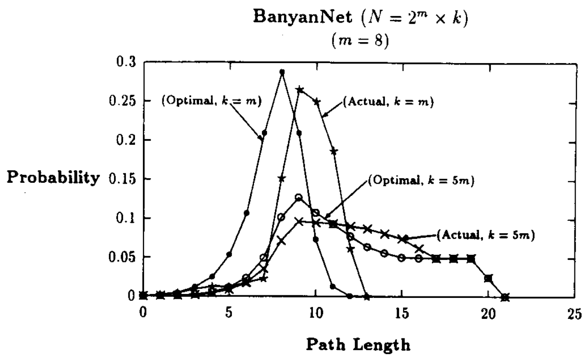


Fig. 6. Path length distribution for BanyanNet.

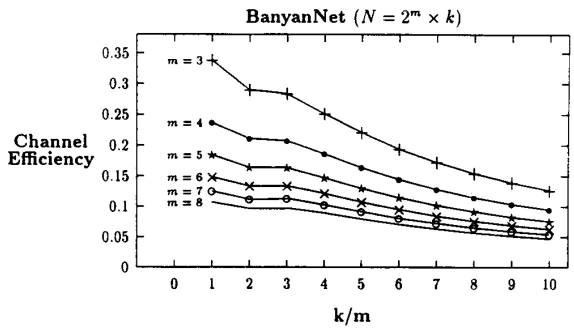


Fig. 7. Channel efficiency for BanyanNet.

For an $N = p^m \times k$ (bidirectional) BanyanNet, we do not have a closed-form solution for the expected number of hops. Instead, we use the average path length obtained by our computer implementation of the routing algorithm to determine channel efficiency, η_B. For the total number of channels, the number of channels per user, the total, and the user throughput, we have

$$\begin{aligned} W_B &= 2kp^{m+1} \\ \omega_B &= 2p \\ C_B &= \eta_B 2kp^{m+1} \\ c_B &= \eta_B 2p. \end{aligned} \tag{5}$$

Assuming a 1-Gb/s user transmission rate, these performance attributes for an $N = 2^m \times k$ BanyanNet with different values of m and k are plotted in Figs. 7–9. As a result of a larger number of users, channel efficiency and throughput per user decrease with increasing m and k. The total network throughput, on the other hand, increases with m and k. However, network throughput tends to saturate when $k/m > 5$, suggesting that for maximum throughput, k should not be more than $5m$. This observation agrees with intuition because when k is much larger than m, the average number of hops between users increases, and messages are routed through a large number of intermediate users.

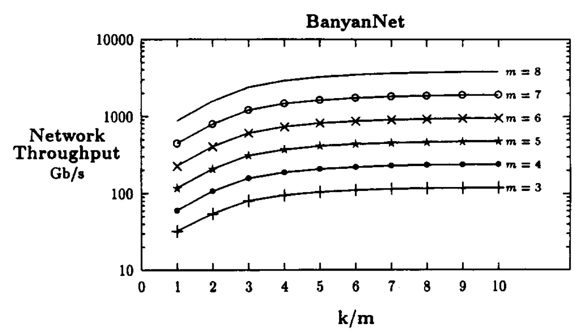


Fig. 8. Network throughput for BanyanNet.

For the purpose of comparison, we consider the generalized ShuffleNet with $N = p^m \times k$ nodes, where k is a multiple of m. For an $N = p^m \times k$ generalized ShuffleNet, the expected number of hops can be calculated by considering the number of users, N_h, h hops from the source node, where $h = 1, \cdots, k + m - 1$. More specifically,

$$N_h = \begin{cases} p^h & \text{if } h = 1, \cdots, m-1 \\ p^m & \text{if } h = m, \cdots, k-1 \\ p^m - p^{h-k} & \text{if } h = k, \cdots, k+m-1. \end{cases}$$

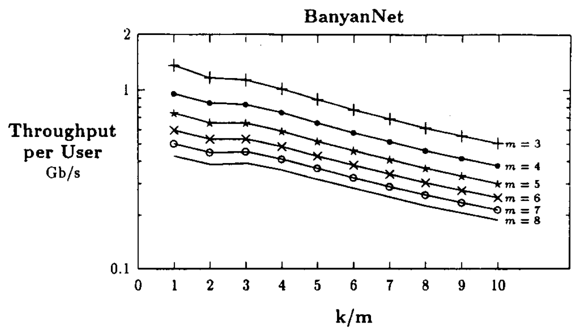


Fig. 9. User throughput for BanyanNet.

Expected number of hops

$$= \frac{1}{p^m k - 1}\left[\sum_{h=1}^{m-1} h p^h + \sum_{h=m}^{k-1} p^m h + \sum_{h=k}^{k+m-1} h(p^m - p^{h-k})\right]$$

$$= \frac{kp^m(p-1)(k+2m-1) - 2k(p^m-1)}{2(p-1)(p^m k - 1)}.$$

Hence, channel efficiency for generalized ShuffleNet η_S is

$$\eta_S = \frac{2(p-1)(p^m k - 1)}{kp^m(p-1)(k+2m-1) - 2k(p^m-1)}.$$

The total number of channels, the number of channels per user, the total, and the user throughput are, respectively,

$$\begin{aligned} W_S &= kp^{m+1} \\ \omega_S &= p \\ C_S &= \eta_S k p^{m+1} \\ c_S &= \eta_S p. \end{aligned} \quad (6)$$

The generalization of bilayered $N = p^m \times k$ ShuffleNet to $k > m$ stages is not trivial. Furthermore, the fact that there are no known self-routing algorithms for these networks makes the generalization which results in more stages of nodes not practical. For this reason, we do not consider the generalized bilayered ShuffleNet. For an $N = p^m \times m$ bilayered ShuffleNet, from [18], the corresponding channel efficiency, total number of channels, number of channels per user, and the total and user throughput are shown at the bottom of this page. From (5), (6), and (7), for the same value of $p, m, k,$ we have

$$\frac{C_B}{C_S} = \frac{c_B}{c_S} = \frac{2\eta_B}{\eta_S}$$

$$\frac{C_B}{C_{BS}} = \frac{c_B}{c_{BS}} = \frac{\eta_B}{\eta_{BS}} \qquad \text{(assume } k = m\text{)}.$$

Therefore, it suffices to compare, simply, the channel efficiency η of these networks. The network and user throughput is just a constant factor of η. With bidirectional channels, an $N = p^m \times k$ BanyanNet has a better performance than the corresponding ShuffleNet. For illustration, we plot the ratio η_B/η_S in Fig. 10. We observe that, for $p = 2$, the superiority of BanyanNet over ShuffleNet increases with k/m but tends to saturate for $k/m > 5$. For $k/m \geq 5$, $\eta_B \approx 2.15\eta_S$. However, this superior performance is achieved at the higher cost of bidirectional links. For $p = 2$, the BanyanNet requires $2p = 4$ transmitters and $2p = 4$ receivers at each node, whereas the ShuffleNet needs only $p = 2$ transmitters and $p = 2$ receivers per node. A better comparison is to compare the $N = 2^m \times k$ BanyanNet (BN, $p = 2$) to $N = 4^m \times k$ ShuffleNet (SN, $p = 4$) and $N = 2^m \times m$ bilayered ShuffleNet (B-SN, $p = 2,\ k = m$). Fig. 11 plots the channel efficiency of these networks versus the number of users. From this figure, we observe that, indeed, the ShuffleNet with $N = 4^m \times m$ (SN, $p = 4,\ k = m$) has the best channel efficiency. In fact, the relative channel efficiency can be summarized

$$\begin{aligned} &\eta_S(p = 4,\ k = m) \\ &> \eta_{BS}(p = 2,\ k = m) \\ &> \eta_B(p = 2,\ k = 3m) \\ &> \eta_B(p = 2,\ k = m) > \eta_S(p = 2,\ k = m). \end{aligned}$$

Fig. 11 also reiterates the drawbacks of the ShuffleNet. Despite its superior channel efficiency, the $p = 4$ ShuffleNet has the smallest possible network configurations. In the range of 100 to 100 000 users, the $p = 4$ ShuffleNet has only 4 possible configurations. The bilayered version offers twice as many (i.e., 8) possible configurations, however, the disadvantage of nonsymmetric node distance of ShuffleNet still exists. More importantly, it is not clear that a self-routing algorithm exists for these networks. The $N = 2^m \times k$ BanyanNet, with k being multiples of m, and the use of bidirectional channels, offers a total of 40 configurations (8 options for each of the 5 possible

$$\eta_{BS} = \frac{(1-p)^2(N-1)}{(1-p)^2 mN + (1-p)N - (1-p)p^m - (1-p)^2 p^{m-1} - 2(1 - p^{(m-1)/2})} \qquad \text{for m even}$$

$$\eta_{BS} = \frac{(1-p)^2(N-1)}{-2/p - 2 + p^{m/2-2}[m + (2-m)p + 2p^2] + N[m + 3/2 - 2mp - 2p + mp^2 + p^2/2]} \qquad \text{for m even}$$

$$\begin{aligned} W_{BS} &= W_B \\ \omega_{BS} &= \omega_B \\ C_{BS} &= \eta_{BS} 2mp^{m+1} \\ c_{BS} &= \eta_{BS} 2p. \end{aligned} \quad (7)$$

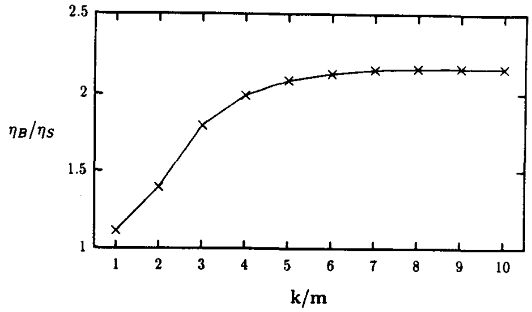


Fig. 10. Performance comparison of $N = 2^m \times k$ ShuffleNet and BanyanNet.

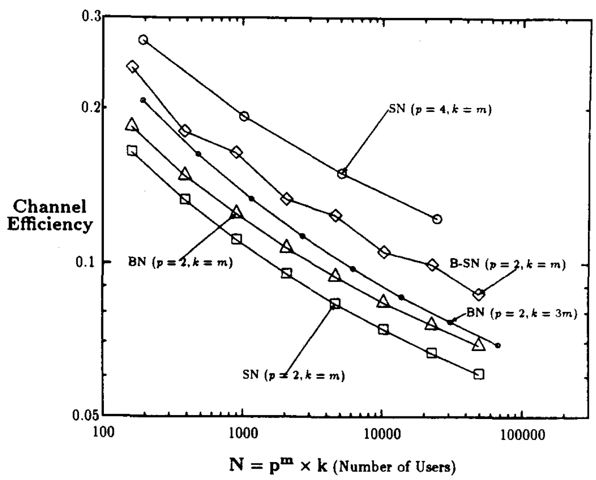


Fig. 11. Channel efficiency versus number of users.

values of $k, k = m, 2m, \cdots, 5m$) and a channel efficiency better than that of the $p = 2$ ShuffleNet.

V. Conclusions

For multihop, WDM-based lightwave networks, a user has a small number of transmitters and receivers, each transmitting and receiving signals in a fixed and assigned wavelength. The wavelength assignment is based on a virtual topology. An efficient virtual topology implies that a large number of users are connected through a small number of hops. Of the many topologies, the unidirectional ShuffleNet is one of the most popular options [6]. However, its limitations and disadvantages include 1) a restricted number of nodes, $p^m \times m$, and 2) nonsymmetric transmission distance between two nodes. For example, if node i connects to node j in one hop, a reply/acknowledgment message from node j to node i may take diameter D_S steps.

To alleviate these problems, our initial approach is to consider a bidirectional ShuffleNet with $p^m \times k$ nodes, where k is a multiple of m. However, a decentralized, bidirectional routing algorithm for ShuffleNet is not obvious. We therefore turned our attention to the SW-Banyan network, a topology proven to be equivalent to the ShuffleNet but possessing a different representation (layout). Analogous to the unidirectional ShuffleNet, we called the resultant network the *BanyanNet.* We emphasize, however, that the BanyanNet is not a novel topology. In fact, due to the isomorphic nature of the two networks, all properties of BanyanNet, including its routing algorithm, can be established for the ShuffleNet with some proper transformation algorithms. The BanyanNet, simply, provides a convenient layout that facilitates the development of the bidirectional routing algorithm.

In general, a BanyanNet has $p^m \times k$ nodes, and each user has $2p$ transmitters and $2p$ receivers. Because of its specific representation, it can be considered as p^m interconnected rings of k nodes. Exploiting this observation, we developed a decentralized, bidirectional routing algorithm for BanyanNet with $p = 2$. This algorithm can be generalized to any value of p. Furthermore, for $p > 2$, multiple users can be assigned to one channel and thus reduce the number of required transmitters and channels [7]. However, in this case, multiple access problems and possible inefficiencies may arise. For $p = 2$, we show that a bidirectional $N = 2^m \times k$ (k is a multiple of m) BanyanNet has diameter $D_B = m + \lfloor m/2 \rfloor$ for $k = m$ and $D_B = \max(2m, \lfloor k/2 \rfloor)$ for $k > m$. On the other hand, for an $N = 2^m \times k$ (k is a multiple of m) unidirectional ShuffleNet, its diameter is $D_S = k + m - 1$. Plotting the ratio of D_S/D_B versus k/m, we found that $D_S > D_B$ and $D_S \approx 2.4 D_B$ when $k = 4$ or 5 is the optimal ratio.

We further evaluate performance of the BanyanNet in terms of *channel efficiency* η_B, network and user *throughput* C_B, c_B. We found that channel efficiency and user throughput decrease with increasing k/m. Network throughput, on the other hand, increases with k/m, but tends to saturate for $k/m > 5$. For comparison, we consider the generalized ShuffleNet ($N = p^m \times k$) with $k \geq m$ stages. We plot η_B/η_S versus k/m. Obviously, $\eta_B > \eta_S$ for all cases, and the superiority of BanyanNet increases with k/m but tends to saturate for $k/m > 5$.

We observe, however, that this superiority is achieved at a higher cost for bidirectional routing. For $p = 2$, the BanyanNet requires $2p = 4$ transmitters and $2p = 4$ receivers at each node, whereas the ShuffleNet needs only $p = 2$ transmitters and $p = 2$ receivers per node. For fairness, we compare the $N = 2^m \times k (p = 2)$ BanyanNet to $N = 4^m \times k (p = 4)$ ShuffleNet and $N = 2^m \times m$ bilayered ShuffleNet. (In this case, all three networks require 4 transmitters and 4 receivers.) By plotting the channel efficiency versus the number of users, Fig. 11 shows that, indeed, the ShuffleNet with $N = 4^m \times m$ ShuffleNet has the best channel efficiency. However, it also indicates that for a range of 100–100 000 users, this $p = 4$ ShuffleNet has the most limited number of available configurations. Bilayered ShuffleNet offers more configurations, but the lack of a self-routing algorithm makes implementation impractical. BanyanNet, on the other hand, provides flexible network configurations, bidirectional self-routing, and a better performance than the $p = 2$ ShuffleNet.

In conclusion, we emphasize that there is no ideal universal topology. An efficient topology is application dependent and is subject to various physical and economical constraints. For WDM-based multihop networks, it is critical that the virtual topology provides a flexible number of nodes (network configurations), short delay (multiple hops between nodes), and a fast, decentralized self-routing algorithm. The development of BanyanNet is established toward these goals. By allowing the number of stages $k(N = p^m \times k)$ to be multiples of m, we have substantially increased the possible number of nodes. The choice of bidirectional channels further decreases the delay (multiple hops) between nodes. Finally, we note that since the BanyanNet is only a different representation of the ShuffleNet, the various properties of BanyanNet can be established in the bidirectional version of the ShuffleNet. In other words, the BanyanNet simply offers a more convenient representation (layout) for routing and can be regarded as the bidirectional equivalent of the ShuffleNet.

REFERENCES

[1] P. S. Henry, "Very-high capacity lightwave networks," in *Proc. Int. Commun. Conf.,* June 1988, pp. 1206–1209.

[2] J. Stone and L. W. Stulz, "Pigtailed high-finess tunable fibre Fabry–Perot interferometers with large, medium and small free spectral ranges," *Electron. Lett.,* vol. 23, pp. 781–783, July 1987.

[3] P. R. Prucnal, M. A. Santoro, and S. K. Sehgal, "Ultrafast all-optical synchronous multiple access fiber networks," *IEEE J. Select. Areas Commun.,* pp. 1484–1493, Dec. 1986.

[4] C. A. Brackett, "Dense wavelength division multiplexing networks: Principles and applications," *IEEE J. Select. Areas Commun.,* vol. 8, pp. 948–964, Aug. 1990.

[5] B. Mukherejee, "WDM-based local lightwave networks—Part I: Single-hop systems," *IEEE Network,* vol. 6, pp. 12–26, May 1992.

[6] ——, "WDM-based local lightwave networks—Part II: Multi-hop systems," *IEEE Network,* vol. 6, pp. 20–32, July 1992.

[7] M. G. Hluchyj and M. J. Karol, "ShuffleNet: An application of generalized perfect shuffles to multihop lightwave networks," *J. Lightwave Technol.,* vol. 9, Oct. 1991.

[8] B. Li and A. Ganz, "Virtual topologies for WDM star LAN's—The regular structures approach," in *Proc. IEEE INFOCOM '92,* May 1992, pp. 2134–2143.

[9] A. S. Acampora, M. J. Karol, and M. G. Hluchyj, "Terabit lightwave networks: The multihop approach," *AT&T Tech. J.,* vol. 66, pp. 21–34, Nov. 1987.

[10] M. G. Hluchyj and M. J. Karol, "ShuffleNet: An application of generalized perfect shuffles to multihop lightwave networks," in *Proc. INFOCOM 1988,* New Orleans, LA, Mar. 1988.

[11] P. W. Dowd, "Wavelength division multiple access channel hypercube processor interconnection," *IEEE Trans. Comput.,* vol. 41, pp. 1223–1241, Oct. 1992.

[12] K. Sivarajan and R. Ramaswami, 'Multihop lightwave networks based on de Bruijn graphs," in *Proc. IEEE INFOCOM '91,* Apr. 1991, pp. 1001–1011.

[13] N. F. Maxemchuk, "Regular mesh topologies in local and metropolitan area networks," *AT&T Tech. J.,* vol. 64, pp. 1659–1685, Sept. 1985.

[14] E. Ayanoglu, "Signal flow graphs for path enumeration and deflection routing analysis in multihop networks," in *Proc. IEEE GLOBECOM '89,* Nov. 1989, pp. 1022–1029.

[15] D. H. Lawrie, "Access and alignment of data in an array processor," *IEEE Trans. Comput.,* vol. 24, pp. 1145–1155, 1975.

[16] H. S. Stone, "Parallel processing with the perfect shuffle," *IEEE Trans. Comput.,* vol. C-20, pp. 153–161, Feb. 1971.

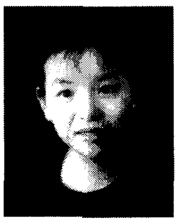
[17] C. Berge, *The Theory of Graphs.* New York: Wiley, 1972.

[18] F. Ayadi, J. F. Hayes, and M. Kavehrad, "Bilayered ShuffleNet: A new logical configuration for multihop lightwave networks," in *1993 IEEE Global Telecommun. Conf.* Houston, TX, Nov. 29–Dec. 2, 1993, pp. 1159–1163.

[19] C. L. Wu and T. Y. Feng, "On a class of multistage interconnection networks," *IEEE Trans. Comput.,* vol. 29, pp. 694–702, Aug. 1980.

[20] L. R. Goke and G. J. Lipovski, "Banyan networks for partitioning multiprocessor systems," in *Proc. 1st Annu. Comput. Arch. Conf.,* 1973, pp. 21–28.

[21] L. R. Goke, "Banyan networks for partitioning multiprocessor systems," Ph.D. dissertation, Univ. Florida, Gainesville, FL, 1976.



K. Wendy Tang received her Ph.D., M.S., and B.S. degrees in electrical engineering from the University of Rochester, Rochester, New York in 1991, 1987, and 1986, respectively.

Since 1991, she has been an assistant professor at the Department of Electrical Engineering in the State University of New York at Stony Brook. Her current research interests include parallel and neural computing, interconnection networks, communication networks, and robotics.

Dr. Tang is a member of the IEEE Computer Society, the Association for Computing Machiner, the Society for Industrial and Applied Mathematics, the International Neural Network Society, and the Society of Women Engineers.