Analysis of the Bidirectional Shufflenet Multihop Topology*

Emilio Leonardi†, Prasasth Palnati and Mario Gerla
{leonardi,palnati,gerla}@cs.ucla.edu
Computer Science Department, UCLA,
405 Hilgard Avenue,
Los Angeles CA 90024.
Tel: (310) 825-1888; Fax: (310) UCLA CSD

Abstract

Bidirectional shufflenet is a virtual multihop topology for optical networks that has been motivated by a need to support backpressure flow control mechanism in the wormhole routing context [14]. The bidirectional shufflenet topology has several desirable features including low average hop distance, support for backpressure flow control, high throughput and deadlock-free routing. In this paper, we analyze the properties of this topology. More specifically, we show how to compute shortest paths and derive the average number of hops. We compare the average hop results with those of other virtual topologies (shufflenet and bilayered shufflenet).

Keywords: multihop virtual topology, shortest path routing, average hop count analysis.

1 Introduction

A multihop virtual topology is composed of stations (each with a fixed number of transmitters and receivers) and 'links' between these stations such that any station can reach any other station by making one or more hops along the links. A 'link' is a point-to-point channel established for communication between two stations. Stations on the path from source to destination act as intermediate nodes forwarding the traffic along the next link on the path to the destination.

*This work was supported by the USDOD ARPA/CSTO under Contract DABT63-93-C-0055. The Distributed Supercomputer Supernet – A Multi Service Optical Intelligent Network.

†Department of Electronics, Politecnico di Torino, Torino 10129

Examples of multihop virtual topologies include Shufflenet, Manhattan Street Networks and de Bruijn Graphs [12].

Multihop topologies are virtual topologies as they are built upon a physical topology (usually a star topology). The 'links' of the virtual topology are different wavelengths that the transmitters and the corresponding receivers (as defined by the topology) are tuned to. Optical fibers offer a vast bandwidth of about 25,000 gigahertz at the 1.55 μm passband (as also at the 0.85 and 1.3 μm passbands). Throughputs higher than traditional electronic networks can be achieved by employing dense Wavelength Division Multiplexing (WDM) [3] to access this vast bandwidth offered by the optical fiber.

In the context of wormhole routing[8] (as in the Supercomputer SuperNet project[9, 10, 5]), losses are possible due to buffer overflows at the input of the nodes. A bidirectional topology allows us to support backpressure flow control in order to avoid such loss. The usefulness of the bidirectional shufflenet multihop topology in this regard was demonstrated elsewhere[14]. The bidirectional shufflenet also lends itself to several approaches for deadlock-free routing[13].

Other variations of the shufflenet, reverse channel augmented shufflenet (RC-shufflenet) [15] and bilayered shufflenet [2] have been explored earlier. Both variations have a shorter diameter than the traditional shufflenet. However, neither is as naturally suited to hop-by-hop flow control as the bidirectional shufflenet. Since there is no expression for the average hop distance in RC-shufflenet, we plot and compare the average hop distances for bidirectional shufflenet with the traditional shufflenet and the bilayered shufflenet only.

In this paper, we describe the bidirectional shufflenet multihop topology (section 2) and examine its properties in section 3. We show how to compute the shortest paths and also calculate the average number of hops in this topology. We plot and compare the average hop distances for various topologies in section 4. We present our conclusions in section 5.

2 Bidirectional Shufflenet

Bidirectional shufflenet is derived from the shufflenet multihop topology [1, 6, 12, 7, 4]. A recirculating perfect shuffle (shufflenet) with parameters p and k has $N = kp^k$ nodes. Each node has p transmitters and receivers. p is also called the connectivity of the network. The N nodes are arranged in k columns of p^k nodes each. Thus, each node can be uniquely identified by a tuple (c, r) where $c \in [0, k-1]$ and $r \in [0, p^k - 1]$. Since each node in a shufflenet has p transmitters, it is connected to p nodes belonging to the next column. The maximum path length (in hops) for a shufflenet is $2k - 1$. An example of an eight node shufflenet is shown in Fig. 1.

The bidirectional shufflenet replaces the unidirectional links of the shufflenet topology with bidirectional links. Thus, though a bidirectional shufflenet with parameters p and k has $N = kp^k$ nodes, each node has $2p$ transmitters and receivers i.e. the network connectivity is $2p$. The N nodes (as in shufflenet) are arranged in k columns of p^k nodes each; each node thus has a unique tuple (c, r) as before. Out of the $2p$ transmitters at each node, p are connected to nodes in the next column and p are connected to nodes in the previous column. An example of an eight node bidirectional shufflenet is shown in Fig. 1.

We define the following notation. The k nodes in the same position of the different columns are said to belong to the same "row" of the shuffle. A node i in row position r_i of column c_i is connected to $2p$ nodes of which p nodes are at positions $|pr_i|_{p^k}, |pr_i + 1|_{p^k}, \cdots, |p(r_i + 1) - 1|_{p^k}$ of column $|c_i + 1|_k$ and p nodes are at positions $(|r_i|_p), (|r_i|_p + p^{k-1}) \cdots (|r_i|_p + (p-1)p^{k-1})$ of column $|c_i - 1|_k$; connections are wrapped around from the k_{th} column to the first, i.e., the topology ideally covers the lateral surface of a cylinder.

Each node i in row r_i has a representation $W(i)$ that is a base p string of length k. $W(i)$ is the representation of r_i in base p. In order to distinguish between nodes at the same position in different columns, we also associate with each node a column identifier. Using $W(i)$, it is possible to derive the representations of the nodes to which node i is connected. A node i with representation $W(i)$ belonging to column c_i has bidirectional links to nodes belonging to column $|c_i + 1|_k$ whose representations are obtained from $W(i)$ by shifting the string $W(i)$ one position to the left (discarding the most significant digit of $W(i)$ in the process) and inserting an arbitrary base p digit in the least significant position. For example, a node with representation $xxx0$ in column j has links to $xx0p'$ in the column $|c_i + 1|_k$, where p' is a base p digit ($0 \le p' \le p - 1$).[1] Bidirectional links also exist between i and the nodes belonging to column $|c_i - 1|_k$ whose representations are obtained from $W(i)$ by shifting the string $W(i)$ one position to the right (discarding the least significant digit of $W(i)$ in the process) and inserting an arbitrary base p digit in the most significant position.

Starting from a node and carrying out the process of shifting representations and inserting symbols iteratively, it is possible to arrive at the representation of any node in the network. The sequence of right or left shifts define the direction (backward and forward, respectively) in which the hops are taken. The base p digit shifted in at each hop identifies the next transmitter on the route.

[1] For the rest of the paper, symbols like p', p'', x_i, y_i etc denote arbitrary base p digits.

3 Properties of Bidirectional Shufflenet

In this section, we examine some of the interesting properties of the bidirectional shufflenet. First, we show how to compute shortest paths for routing in this topology. The computation of shortest paths leads us to a derivation of the average number of hops taken by traffic sent through the multihop network. Then we discuss the natural support that the bidirectional shufflenet topology provides for hop-by-hop flow control.

3.1 Shortest Path Routing

As the bidirectional shufflenet topology is a regular topology, for our analysis, we choose one node at random and describe how to compute the shortest paths from this node to all other nodes.

Every path on a bidirectional shuffle is composed of sequences of forward hops and/or backward hops. Let us examine a sequence composed of h_f forward hops followed by h_b ($h_b < h_f$) backward hops. The only nodes reachable from a node i in column c_i with representation $W(i) = x_{k-1}x_{k-2}\dots x_2x_1x_0$, through a sequence of h_f forward hops are those belonging to column $|c_i+h_f|_k$ whose representation is $W(x) = x_{k-1-h_f}x_{k-2-h_f}\dots x_0P'$ where $P' = p'p'\dots p'$ and $|P'| = h_f$. Taking h_b backward hops from $W(x)$, it is possible to reach the nodes of column $|c_i + h_f - h_b|_k$ whose representation is

$$W(y) = P''x_{k-1-h_f}x_{k-2-h_f}\dots x_1x_0P'$$

where $P'' = p''p''\dots p''$, $|P''| = h_b$, $P' = p'p'\dots p'$ and $|P'| = h_f - h_b$.

It is important to note that after the shifting process associated with the path of $h_f + h_b$ hops, only h_f digits are present in $W(y)$ since $h_f > h_b$. In general, if after h_+ hops in one direction, h_- hops in the other direction are made, the arbitrary symbols inserted during the first part of the shifting process are in part or totally discarded during the second part of the shifting process, and the arbitrary number of symbols left in the shift register after the entire process is $max(h_+, h_-)$. This observation is formally proved in the Appendix.

As a consequence, at least k hops in the same direction are necessary to reach a node whose representation has no common substring with the source representation. The maximum path length in the bidirectional shufflenet topology is $\lfloor\frac{3k-1}{2}\rfloor$ hops (k hops to reach any other node belonging to the source column and at most $\lfloor\frac{k-1}{2}\rfloor$ more hops are necessary to reach the destination column.)

It is important to realize that paths composed of many sequences of forward hops and backward hops are not optimal since after the entire process of shifting only few inserted symbols are left in the shift register. The following example can help to clarify this concept.

Let the node in column c_i whose representation $W(i)$ is $000\dots000$ be the source of packets routed through paths composed by 8 hops: ffbbffbb, where f is a forward hop and b is a backward

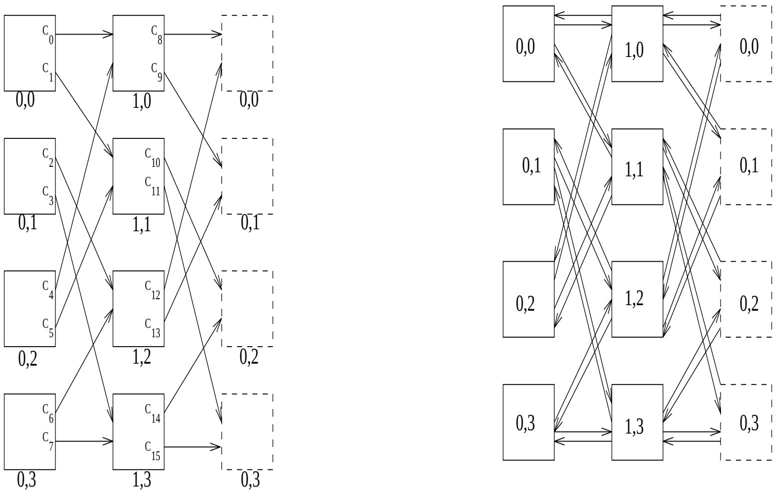

Figure 1: An 8-node shufflenet and an 8-node bidirectional shufflenet multihop topology. The first and last columns represent the same nodes.

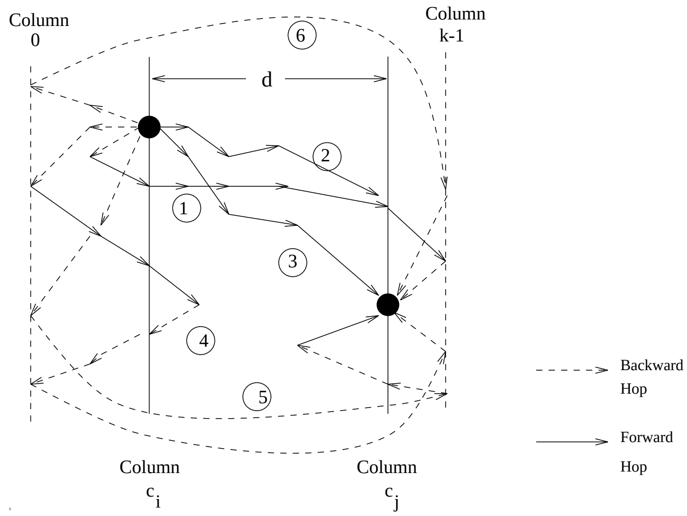

Figure 2: Two types of paths to reach destination node in column c_j from source node in column c_i – Paths 1 (bfb), 2 (fb) and 3 (f) are essentially composed of forward hops (Approach F); paths 4 (bfb), 5 (bf) and 6 (b) are composed essentially of backward hops (Approach B).

hop. The nodes reached are those belonging c_i whose representation is $p'p'000\ldots0$. The same set of nodes can be reached through paths of only four hops (instead of 8): ffbb.

The previous example can be formalized in the following assertion whose proof is given in appendix: shortest paths in the bidirectional shufflenet topology are of the type **f**, **b**, **fb**, **bf**, **fbf** and **bfb** only, where **f** and **b** are sequences of one or more hops in the forward direction and the backward direction, respectively. If a path is of **fbf** or **bfb** type, the longest sequence of hops is the middle one. We consider only shortest paths for the rest of the paper.

Let the source node i in column c_i have a representation $W(i) = x_{k-1}x_{k-2}\ldots x_2x_1x_0$. Let the destination node j in column c_j have a representation $W(j) = y_{k-1}y_{k-2}\ldots y_2y_1y_0$.

Let $d' = (|c_j - c_i|_k)$. Let $d = min(d', k - d')$ be the distance between source and destination columns. A node d' columns away can be reached through paths of type **f**, **b**, **fb**, **bf**, **fbf** and **bfb** using two possible approaches – mainly by hops in the forward direction (Approach F) or mainly by hops in the backward direction (Approach B) (See Fig. 2). In approach F, at least d' hops are made in the forward direction. In approach B, at least $k - d'$ hops are made in the backward direction. Shortest paths have a length not greater than $k + d$ hops, since through $k + d$ hops in the same direction (an **f** or **b** path), it is possible to reach all the nodes of the destination column. To discover the shortest path, we calculate the shortest path by each approach and then select the shorter one.

Consider first approach F paths. In this case at least d' leftmost symbols of the source address are discarded during the shifting process and the remaining digits are shifted left by d' positions with respect to their original position.

To find the shortest path from source to destination we evaluate the minimum number of new digits that should be inserted in order to get $W(j)$ from $W(i)$.

We split the representations for nodes i and j into two substrings: $W(i)$ $A'A$ and $W(j) = BB'$. The substrings A and B are each of length $k - d'$ and the substrings A' and B' are each of length d'. Now, compare the substrings A and B digitwise. Let the result $comp(A, B) = amb$, where a, m and b are strings (possibly empty). m is the maximal substring of B which matches digits in the same places of the string A. a and b are the strings to the left and right of m respectively, where B and A differ. For example, if $A = 000110000$ and $B = 101111011$, then $a = 101$, $|a| = 3$, $b = 1011$, $|b| = 4$, $m = 11$ and $|m| = 2$.

The minimum number of new digits that should be inserted to get $W(j)$ from $W(i)$ is $d' + |a| + |b|$. The digits in m are common to the source and destination representation and must not be replaced. Since the number of arbitrary symbols left after the shifting process in the destination address is not greater than $max(h_f, h_b)$, and since approach F paths are considered, it follows that $h_f \geq d' + |a| + |b|$. The number of backward hops is $h_f - d'$ since $h_f - h_b = d'$. If a path with

$h_f = d' + |a| + |b|$ exists it will necessarily be a shortest path.

Based upon the size of strings a and b, we can use Table 1 to decide the path (as sequence of hops) and the path length using approach F. It must be noted that every path in the table satisfies the condition $h_f = d' + |a| + |b|$ and as a consequence is a shortest path among approach F paths.

Now, consider approach B. This case is analogous to the one presented above with the substitution of $k - d$ wherever d was used and b where f was used. Thus, now we split the representations for nodes i and j, $W(i) = A^{'}A$ and $W(j) = BB^{'}$, where $|A| = |B| = k - d$ and $|A^{'}| = |B^{'}| = d$. Compare the substrings $A^{'}$ and $B^{'}$ digitwise. Let the result $comp(A^{'}, B^{'}) = amb$. a, m and b have the same meaning as before. In this case, we can use Table 2 to decide the path and the path length using mainly backward hops.

Apart from the two approaches discussed, another path exists in which $k+d$ forward (if $d' \leq k-d'$) or backward hops (if $k - d' \leq d'$) are made. The three paths so obtained are compared on the basis of their length and then the shortest one is chosen. The routing is derived from Tables 1 and 2.

3.2 Average number of hops

In order to calculate the average distance in the network, we pick a node at random and calculate the average distance from that node to every other node. In the previous section we provided an algorithm to find the shortest path between any pair of nodes. To evaluate the average distance in a bidirectional shuffle, we list all possible shortest paths obtained from the algorithm and count the length of each path thereby, getting the length distribution.

Paths of several lengths are possible from the node i belonging to column c_i to nodes belonging to a column $|i + d'|_k$, as discussed earlier. In table 3, we show the decomposition of such paths in terms of the number of forward hops and number of backward hops for the case when k is an odd integer. A corresponding table (Table 4) is presented for the even k case. In both tables, h_+ represents the number of forward hops if $d' \leq k - d'$, and the number of backward hops if $k - d' \leq d'$. On the other hand, h_- represents the number of backward hops if $d' \leq k - d'$, and the number of forward hops if $k - d' \leq d'$.

Paths shorter than $k - d$ hops belonging to only one of the two approaches exist. Possible lengths for these paths are d (all the hops in the same direction), $d + 2$ ($d + 1$ in one direction and 1 in the other) etc. All the possible lengths can be obtained by solving the following equations when $h < k - d$:

$$h_+ + h_- = d + 2i \quad \forall i \geq 0 \tag{1}$$

$$h_+ - h_- = d \tag{2}$$

When $h = k - d$, paths due to the other approach from node i to the destination column composed

Size $\|a\|$	of $\|b\|$	Direction of hops	Number of hops	String to shift into $W(i)$
0	0	f	d'	B'
> 0	0	f	$d' + \|a\|$	B', don't care
		b	$\|a\|$	a
0	> 0	b	$\|b\|$	don't care
		f	$\|b\| + d'$	b, B'
> 0	> 0	b	$\|b\|$	don't care
		f	$\|b\| + d' + \|a\|$	b, B', don't care
		b	$\|a\|$	a

Table 1: Shortest path routing from $W(i)$ to $W(j)$ using essentially forward hops (Approach F).

Size $\|a\|$	of $\|b\|$	Direction of hops	Number of hops	String to shift into $W(i)$
0	0	b	$k - d'$	B
> 0	0	f	$\|a\|$	don't care
		b	$\|a\| + k - d'$	a, B
0	> 0	b	$\|b\| + k - d'$	don't care, B
		f	$\|b\|$	b
> 0	> 0	f	$\|a\|$	don't care
		b	$\|a\| + k - d' + \|b\|$	a, B, don't care
		f	$\|b\|$	b

Table 2: Shortest path routing from $W(i)$ to $W(j)$ using essentially backward hops (Approach B).

solely of hops in the opposite direction exist. This is reflected in the tables. For odd k, the paths of the two different families have different length: $k-d-1$, $k-d+1$, $k-d+3$ and so on for approach F paths; $k-d$, $k-d+2$, $k-d+4$ and so on for approach B paths. For even k, we have paths belonging to the two approaches for the same $k-d$, $k-d+2$ etc. Paths due to approach B satisfy the following equations:

$$h_+ + h_- = k-d+2i \quad \forall i \geq 0 \tag{3}$$

$$h_+ - h_- = k-d \tag{4}$$

We compute $N(h)_d$ as the number of nodes that can be reached with paths of length less than or equal to h given that the destination column is d away from the source column.

In the following we assume, without loss of generality, that node $00\ldots00$ of column c_i is the source, and we consider a destination column such that $d' = d < k - d'$.

If the minimal pathlength is d, the corresponding paths are approach F paths. Through these paths it is possible to reach all the nodes of the destination column whose representation contains zeros in the $k-d$ leftmost positions. The number of such nodes is p^d.

Paths of length $d+2$ can be divided into two groups: G^1_{d+2} paths in which a backward hop is followed by $d+1$ forward hops and G^2_{d+2} paths, in which $d+1$ forward hops are followed by 1 backward hop. G^1_{d+2} paths reach all the p^{d+1} nodes whose representation contains arbitrary symbols in the leftmost position and in the d rightmost positions, and zeros elsewhere. G^2_{d+2} paths reach all the p^{d+1} nodes whose representations have zeros in the $k-d-1$ left positions. An intersection between node reachable through G^1_{d+2} and G^2_{d+2} paths exists, it contains all those nodes whose representations have zeros in $k-d$ leftmost positions (the same nodes reachable through paths of d length). Thus, $N(d+2)_d = 2 * p^{d+1} - p^d$.

All approach F paths of length $h = d+2i$ reach nodes whose representation has at least $k-d-i$ consecutive zeros in the $k-d$ leftmost symbols. In fact for nodes satisfying this condition, and only for these nodes, the shortest path algorithm described in the previous section provides a path of length not greater than $d+2i$. Assuming $k=10$, $d=2$, $i=3$, the only nodes reachable through approach F paths of length not greater than $d+2i=8$ are:

$$0\;0\;0\;0\;0\;p'p'p'p'p' \tag{5}$$

$$p'0\;0\;0\;0\;0\;p'p'p'p' \tag{6}$$

$$p'p'0\;0\;0\;0\;0\;p'p'p' \tag{7}$$

$$p'p'p'0\;0\;0\;0\;0\;p'p' \tag{8}$$

The first group of nodes is reached through paths along bbbfffff; the second group through paths bbfffffb, etc. The number of nodes whose representation contains at least $k-d-i$ consecutive

h	h_+	h_-
d	d	0
$d+2$	$d+1$	1
$d+4$	$d+2$	2
$\vdots$	$\vdots$	$\vdots$
$k-d-1$	$d+(k-2d-1)/2$	$(k-2d-1)/2$
$k-d$	0	$k-d$
$k-d+1$	$d+(k-2d+1)/2$	$(k-2d+1)/2$
$k-d+2$	1	$k-d+1$
$k-d+3$	$d+(k-2d+3)/2$	$(k-2d+3)/2$
$\vdots$	$\vdots$	$\vdots$
$k+d$	$k+d$	0

Table 3: Decomposition of paths (in hops) for odd k to reach nodes d columns away from the source

h	h_+	h_-
d	d	0
$d+2$	$d+1$	1
$d+4$	$d+2$	2
$\vdots$	$\vdots$	$\vdots$
$k-d$	$d+(k-2d)/2$	$(k-2d)/2$
$k-d$	0	$k-d$
$k-d+2$	$d+(k-2d)/2+1$	$(k-2d)/2+1$
$k-d+2$	1	$k-d+1$
$\vdots$	$\vdots$	$\vdots$
$k+d$	$k+d$	0

Table 4: Decomposition of paths for even k to reach nodes d columns away from the source

zeros in the first $k-d$ positions is given by:

$$S_i p^d \qquad \forall h < k \tag{9}$$

$$(S_i - T_{z_1})p^d \qquad \forall h \geq k \tag{10}$$

$$\tag{11}$$

where S_i, T_i, z_1 are defined below:

$$S_i = 0 \qquad \forall i < 0 \tag{12}$$

$$S_i = (i+1)p^i - ip^{(i-1)} \qquad \forall i \geq 0 \tag{13}$$

$$T_i = 0 \qquad \forall i < 0 \tag{14}$$

$$T_i = i(p-1)p^{i-1} + \frac{i(i-1)}{2}(p-1)p^{i-2} \qquad \forall i \geq 0 \tag{15}$$

$$i = \lfloor [(h-d)/2] \rfloor \tag{16}$$

$$j = \lfloor [(h-k+d)/2] \rfloor \tag{17}$$

$$z_1 = d + 2i - k \tag{18}$$

$$z_2 = k - d + 2j - k \tag{19}$$

For paths of length at least $k-d$ hops, approach B paths exist. Similar arguments allow us to evaluate the number of nodes reachable through approach B paths. $N(h)_d$ with $h \geq k-d$ is given by the the sum of sets of nodes that can be reached through approach F and approach B. The final result is shown in table 5.

We now calculate $L(d)$, the sum total of all paths (in hops) from the node under observation to all the nodes belonging to a given column d columns away as:

$$L(d) = \sum_{i=0}^{i=l_{MAX}} [N(l_{MAX})_d - N(i)_d] \tag{20}$$

where l_{max} is the length of the longest path linking the source node to the destination nodes d columns away.

The average distance $E[d]_{bs}$ can now be obtained by summing over all values of d and dividing by the total number of nodes to which paths of non-zero length exist, i.e., all nodes except for the source node. This gives us the following equation:

$$E[d]_{bs} = \frac{\sum_{w=-\lfloor[(k-1)/2]\rfloor}^{\lfloor[k/2]\rfloor} L(w)}{kp^k - 1} \tag{21}$$

In Table 6, we show the average number of hops ($E[d]_{bs}$) in a bidirectional multihop topology for different values of p and k using the equation for $E[d]_{bs}$. We also show values for the shufflenet and the bilayered shufflenet; we discuss those in the next section.

h	$N(h)_d$
$h < d$	0
d	p^d
$d < h < k-d$	$S_i p^d$
$k-d \leq h < k$	$S_i p^d + S_j p^{k-d} - S_i S_j$
$k \leq h < k+d$	$((S_i - T_{z_1})p^d + (S_j - T_{z_2})p^{k-d} - (S_i - T_{z_1})(S_j - T_{z_2})$
$h = k+d$	$P^k - N(k+d-1)$

Table 5: Distribution of $N(h)_d$

p	K	N	$E[d]_{bs}$	$E[d]_{us}$	$E[d]_{bil}$	p	K	N	$E[d]_{bs}$	$E[d]_{us}$	$E[d]_{bil}$
2	2	8	1.5	2.0	2.00	4	2	32	2.00	2.26	2.25
2	3	24	2.29	3.26	2.40	4	3	192	2.99	3.69	2.68
2	4	64	3.375	4.63	3.63	4	4	1024	4.22	5.17	4.17
2	5	160	4.312	6.069	4.22	4	5	5120	5.31	6.67	4.69
2	6	384	5.44	7.53	5.53	4	6	24576	6.53	8.20	6.17
2	7	896	6.45	9.017	6.14	8	2	128	2.25	2.37	2.38
2	8	2048	7.60	10.50	7.51	8	3	1536	3.33	3.86	2.80
2	9	4608	8.66	12.00	8.11	8	4	16384	4.62	5.36	4.35
2	10	10240	9.81	13.50	9.50						
2	11	22528	10.90	15.00	10.05						
2	12	49152	12.06	16.50	11.50						

Table 6: Average distances for bidirectional shufflenets ($E[d]_{bs}$), unidirectional shufflenets ($E[d]_{us}$) and finally bilayered shufflenets ($E[d]_{bil}$).

3.3 Flow Control

Bidirectional shufflenet provides a natural way for implementing hop-by-hop backpressure flow control. For this purpose, a fixed size buffer, called slack buffer, is associated with each receiver in the nodes. Flow control signals STOP and GO are used to regulate the flow of traffic across the optical link. A STOP/GO signal is sent in the direction opposite to the one on which the traffic is being received (thus, the need for bidirectional links). Suitable high-water and low-water marks are used to decide when to send a STOP signal and when to send a GO signal respectively. The study of flow control on the bidirectional shufflenet topology has been presented elsewhere[14].

4 Performance Comparisons

In this section, we compare the bidirectional shufflenet with the traditional shufflenet [6] and another variation of the shufflenet topology called the bilayered shufflenet [2]. The bilayered shufflenet uses the same number of resources as the bidirectional shufflenet ($2p$ transmitters and receivers per node). In addition to the connectivity of the regular shufflenet, p links from each node are connected to the previous column to a "mirror" image of stations. Thus, the resulting topology is such that all stations are wrapped twice around the same cylinder (one shufflenet in one direction and another shufflenet in the opposite direction). See Fig. 5 for a $p = 2$, $k = 3$, 24 node bilayered shufflenet. Figs. 3 and 4 show a 24 node unidirectional shufflenet and bidirectional shufflenet for comparison.

In Table 6, we show the average number of hops encountered by multihop traffic for three topologies: a bidirectional shufflenet, shufflenet and the bilayered shufflenet. In Fig. 6, the average distances for the three topologies are plotted for comparison. As is evident, the average hop distance in the bilayered shufflenet and the bidirectional shufflenet are comparable. Both these topologies have a greater average hop distance than shufflenets when the same number of resources (transmitter/receiver pairs) are considered per node. However, bidirectional shufflenet provides support for hop-by-hop backpressure flow control thus avoiding worm loss and providing higher throughputs in the wormhole routing context [14] which is not possible in unidirectional shufflenets. Thus, it is clear that despite having a higher average hop distance, performance does not suffer. Also, bidirectional topologies, in general, have two further advantages: if deflection routing is used, the cost of deflection is two hops (for a unidirectional shufflenet, the cost would be k hops) [11], and also, it is possible to place two nodes that exchange a lot of traffic next to each other, thus avoiding congestion in the rest of the network (for a unidirectional shufflenet, the traffic in at least one direction has to compete with other traffic).

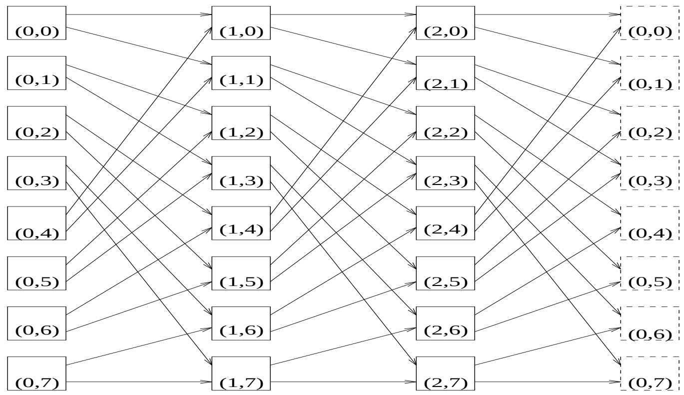

Figure 3: A 24 node shufflenet with $p = 2$, $k = 3$.

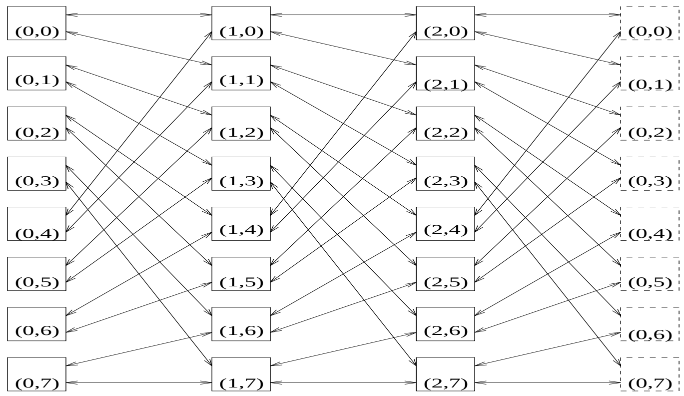

Figure 4: A 24 node bidirectional shufflenet with $p = 2$, $k = 3$.

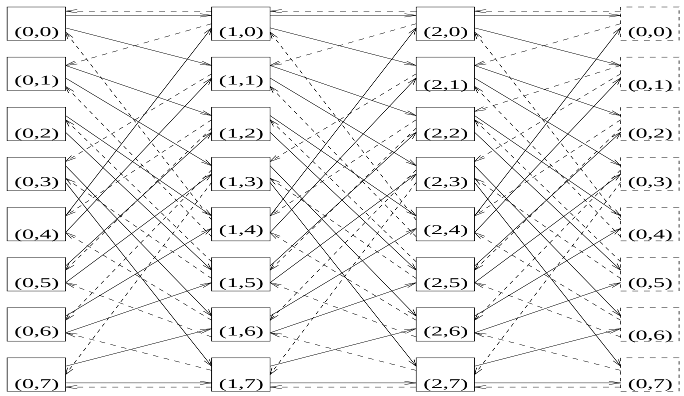

Figure 5: A 24 node bilayered shufflenet with $p = 2$, $k = 3$.

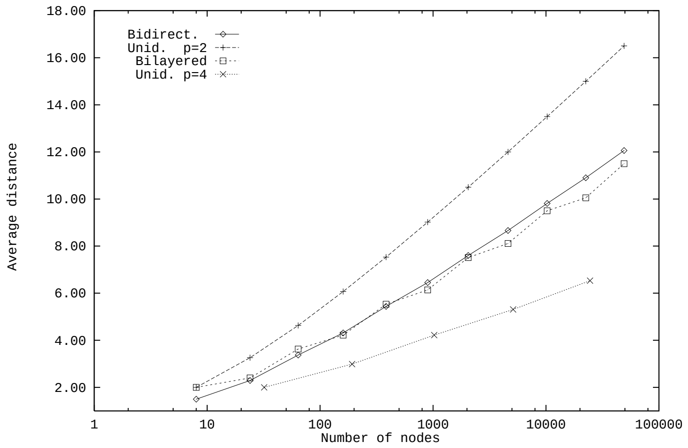

Figure 6: The average distance (in hops) for the following topologies: shufflenet with $p = 2$, shufflenet with $p = 4$, bidirectional shufflenet with $p = 2$ and bilayered shufflenet.

5 Conclusions

The bidirectional shufflenet is a multihop topology derived from the traditional shufflenet topology. It provides a natural way to implement a hop-by-hop backpressure flow control mechanism. The need for such a mechanism arises due to worm loss caused by buffer overflow. The use of the bidirectional shufflenet topology along with the hop-by-hop backpressure flow control mechanism eliminates worm loss.

Computation of shortest paths in the bidirectional shufflenet was presented. Also, the calculation of the average number of hops was shown. The average hop measure for the bidirectional shufflenet is comparable with that of the bilayered shufflenet. However, both these topologies have a greater average hop distance than a shufflenet that has the same number of transmitter/receiver pairs (2p per node).

The bidirectional shufflenet has several features that make it attractive: flow control capability[14], higher throughputs and capability to do deadlock-free routing[13]. This multihop topology fares very well in comparison with the traditional shufflenet. Besides, it also lends itself to design of practical networks that use the wormhole routing and backpressure flow control mechanisms such as the SSN project.

References

[1] A. S. Acampora. A Multichannel Multihop Local Lightwave Network. In *Proceedings of*

GLOBECOM 87, pages 1459–1467, November 1987.

[2] F. Ayadi, J. Hayes, and M. Kavehrad. WDM Cross-Connected Star Topology for the Bilayered Shufflenet. *Journal of Lightwave Technology*, 12(9):1668–1678, September 1994.

[3] C. A. Brackett. Dense Wavelength Division Multiplexing Networks: Principles and Applications. *IEEE Journal on Selected Areas in Communications*, 8:948–964, August 1990.

[4] M. Eisenberg and N. Mehravari. On the Performance of Multihop, Multichannel Lightwave Networks and Non-uniform Traffic. *IEEE Journal on Selected Areas in Communications*, 6(7):1063–1078, August 1988.

[5] M. Gerla, B. Kannan, and P. Palnati. Protocols for an Optical Star Interconnect for High Speed Mesh Networks. In *Proceedings of INFOCOM 95*, pages 146–153, April 1995.

[6] M. Hluchyj and M. Karol. Shufflenet: An Application of Generalized Perfect Shuffles to Multihop Lightwave Networks. *IEEE/OSA Journal of Lightwave Technology*, 9(10):1386–1397, October 1993.

[7] M. Karol and S. Shaikh. A Simple Adaptive Routing Scheme for Shufflenet Multihop Lightwave Networks. In *Proceedings of GLOBECOM 88*, pages 1640–1647, 1988.

[8] P. Kermani and L. Kleinrock. Virtual Cut-through: A New Computer Communication Switching Technique. *Computer Networks*, 3(3):267–286, September 1979.

[9] L. Kleinrock, N. Bambos, J. Cong, E. Gafni, M. Gerla, and L. Bergman. Optimic: A Scalable Distributed, All-Optical Terabit Network. *Journal of High Speed Networks*. To appear in Special Issue on All-Optical Networks.

[10] L. Kleinrock et al. The Supercomputer Supernet Testbed: A WDM Based Supercomputer Interconnect. *IEEE JSAC/JLT*. To appear in Special Issue on Multiple Wavelength Optical Technologies and Networks.

[11] E. Leoanardi, F. Neri, P. Palnati, and M. Gerla. Congestion Control Techniques in Asynchronous Wormhole Routing Networks. Submitted for publication.

[12] B. Mukherjee. WDM-based Local Lightwave Networks Part II: Multi-Hop Systems. *IEEE Network*, 6(4):20–32, July 1992.

[13] P. Palnati, M. Gerla, and E. Leonardi. Deadlock-free Routing in a Hierarchical Supercomputer Interconnection Network. Submitted for publication.

[14] P. Palnati, E. Leonardi, and M. Gerla. Bidirectional Shufflenet: A Multihop Topology for Backpressure Flow Control. September 1995. To appear in *Proceedings of* 4^{th} *International Conference on Computer Communications and Networks.*

[15] A. Pavan, S. Bhattacharya, and D. Du. Reverse Channel Augmented Multihop Lightwave Networks. In *Proceedings of INFOCOM 94*, volume 3, pages 1127–1134, March 1994.

A Assertion 1

In the following, h_f refers to the number of forward hops while h_b refers to the number of backward hops.

Assertion 1: Starting from node i with representation $W(i)$, in $h = h_f + h_b$ hops, it is possible to reach only those nodes whose representations $W(j)$ contains at least one substring of length $l = k-\max(h_b, h_f)$ that is also a substring of $W(i)$. If $W(i)$ and $W(j)$ have no common substring then $\max(h_b, h_f) \geq k$.

Proof: We have already described how the representation of the node(s) reached by a single forward (backward) hop are obtained from the current representation (in a shift register) by shifting it left (right); discarding the most (least) significant digit and inserting a digit in the least (most) significant position. It is easy to see that a forward hop followed immediately by a backward hop (or vice-versa) results in a representation that differs from the original one in exactly one bit position.

Let $X(i) = X_f(i) - X_b(i)$, $i = 0, 1, 2, \ldots, h$. $X(i)$ represents the difference between the number of forward hops $X_f(i)$ and backward hops $X_b(i)$ after the first i hops of the h total hops. Let $X(0) = 0$. $X(i)$ can be negative, positive or zero. Define $X_{min}(i)$ to be the minimum value reached by $X(i)$ during the first i hops. Define $X_{max}(i)$ to be the maximum value taken by $X(i)$ during the first i hops.

After each hop, $X(i)$ could increase in value or decrease. If a symbol p_0 is inserted into the shift register when $X(i) = X_0$, then p_0 will be shifted out of the register the next time $X(i)$ reaches the value X_0 again (while shifting another symbol at the other end of the string). After a hop if $X(i)$ attains a value for the first time (i.e. a new minimum or maximum), a new symbol is inserted into $W(i)$. On the other hand, if $X(i)$ reaches a value that it has had earlier, then the number of symbols inserted in $W(i)$ remains the same as before (one shifted out and one shifted in). Thus, after j hops, $j \leq h$, $|X_{max}(j) - X_{min}(j)|$ equals the number of new symbols inserted into $W(i)$ in the j hops. Note that $|X_{max}(j) - X_{min}(j)|$ is not greater than the maximum number of hops made in one of the two possible directions in the first j hops. Thus, after h hops, $\max(h_b, h_f)$ is the number of new symbols inserted into $W(i)$. $W(j)$ then differs from $W(i)$ by not more than $\max(h_b, h_f)$ symbols. This implies that at least $l = k-\max(h_b, h_f)$ symbols of $W(i)$ are still present in the final representation $W(j)$. It follows that if $W(i)$ and $W(j)$ have no common strings then it must be the case that $\max(h_b, h_f) \geq k$.

B Assertion 2

In the following, f represents a sequence of one or more forward hops while b represents a sequence of one or more backward hops.

Theorem: Shortest paths in the bidirectional shufflenet topology are of the type f, b, fb, bf, fbf and bfb only. Any other sequence of hops would not be a shortest path. For every shortest path of kind fbf or bfb the longest sequence of hops in the same direction must be the middle one.

Proof: (By Contradiction)

Assume that a path of type fbfb is a shortest path of length $h = h_{f1}+h_{b1}+h_{f2}+h_{b2}$, where $f1$, $b1$, $f2$ and $b2$ are the number of hops in the sequences of forward and backward hops and h_{f1}, h_{b1}, h_{f2} and $h_{b2} > 0$, between a source node in column c_i (with representation $W(i)$) and a destination node in column c_j (with representation $W(j)$). Let $0 \leq d \leq \lfloor (k-1)/2 \rfloor$ be the distance between the two columns. Thus, $h \geq d$. Consider the function $X(i)$ defined earlier for $i = 0, 1, 2, \ldots, h$. Assume $X(0) = 0$.

Consider first the case where one of either h_{b1} or h_{f2} equals 1. Assume, without loss of generality, that it is the backward hop sequence which is of length one (i.e. $h_{b1} = 1$). We focus on the sequence: a forward hop followed by a backward hop followed by another forward hop. It is clear that the backward hop and the following forward hop can be eliminated and we would still get the same representation, $W(j)$, for the final node reached. Hence, such paths cannot be shortest paths.

$X(h) = d$. Assume now that h_{b1} and $h_{f2} > 1$. Thus, there must be at least one value, say k, such that $X(i) = k$ after i_0, i_1 and i_2 hops, where $0 < i_0 < i_1 < i_2 < h$. It must be the case that i_0 and i_2 are hops in the same direction while i_1 is a hop in the opposite direction. The symbols inserted at hop i_0 and i_1 are both shifted out and hence do not help form the final representation $W(j)$. Thus, a shorter path in which the two hops at i_0 and i_1 are not present exists. This contradicts our assumption that fbfb was a shortest path.

The proof for bfbf is identical to the one above. Since all paths other than f, b, fb, bf, fbf and bfb, have either fbfb or bfbf as subsequences, it follows that none of them are shortest paths in the bi-directional shufflenet topology. Thus, we have proved that only paths of the type f, b, fb, bf, fbf and bfb provide shortest paths in this topology.

Arguments similar to assertion 1 allow us to state that for each shortest path of the kind fbf or bfb, the longest sequence of hops in the same direction must be the middle one. If the central sequence (assumed to be b without loss in generality) is of unit size, then a sequence of hops fbf (one forward hop followed by a backward hop followed by another forward hop) exists. We proved earlier that this is not optimal. If the central sequence is longer than 1 hop but not greater than the other two sequences, there must be at least one value k such that $X(i) = k$ for three different i_0, i_1 and i_2.