A New Architecture for Multihop Optical Networks

A. Jaekel[1], S. Bandyopadhyay[1] and A. Sengupta[2]
[1] School of Computer Science, University of Windsor
Windsor, Ontario N9B 3P4

[2] Dept. of Computer Science, University of South Carolina
Columbia, SC 29208

Abstract. Multihop lightwave networks are becoming increasingly popular in optical networks. It is attractive to consider regular graphs as the logical topology for a multihop network, due to its low diameter. Standard regular topologies are defined only for networks with the number of nodes satisfying some rigid criteria and are not directly usable for multihop networks. Only a few recent proposals (e.g., GEMNET [10]) are regular and yet allow the number of nodes to have any arbitrary value. These networks have one major problem - node addition requires a major redefinition of the network. We present a new logical topology which has a low diameter but is not regular. Adding new nodes to a network based on this topology involves a relatively small number of edge definition/redefinitions. In this paper we have described our new topology, a routing scheme that ensures a low diameter and an algorithm for adding nodes to the network.

1 Introduction

Optical networks have become one of the emerging technologies for computer communication and provides low-loss transmission over a frequency range of about 25THZ. Signals can be transmitted over large distances at high speeds before amplification or regeneration is needed. Wavelength division multiplexing(WDM) allows the simultaneous data transmission at different carrier frequencies over the same optical fiber. A WDM network can offer lightpath service where a lightpath is a circuit switched connection between two nodes in the network and it is set up by assigning a dedicated wavelength to it on each link in its path [1]. We are looking at networks that provide permanent lightpaths which are set up at the time the network is initially deployed or is subsequently updated by addition of new nodes. Local area optical WDM networks may be categorized as single-hop or multi-hop networks. A single hop network allows communication between any source-destination pair using a single lightpath from the source to the destination. In a multihop network [2], to communicate from a source to a destination, it is necessary to use a composite lightpath consisting of a number of lightpaths. In other words, to communicate from a source node u to a destination node v, we have to select nodes $x_1, x_2, ..., x_{k-1}$ such that there is a lightpath $\boldsymbol{L}_0$ from node u to node x_1, a lightpath $\boldsymbol{L}_1$ from node x_1 to node x_2, ..., a lightpath $\boldsymbol{L}_{k-1}$ from node x_{k-1} to node v. Our composite lightpath $\boldsymbol{L}$ for communication from u to v consists of the lightpaths $\boldsymbol{L}_0, \boldsymbol{L}_1, ... , \boldsymbol{L}_{k-1}$ and, in order to use this composite lightpath to send some data from *u* to *v*, we have to first send the data from node *u* to node $x_1$using lightpath $\boldsymbol{L}_0$. This data will be buffered at node x_1 using electronic media until the lightpath $\boldsymbol{L}_1$ from node x_1 to node x_2 is available. When $\boldsymbol{L}_1$ is available, the data will be communicated to node x_2 using lightpath $\boldsymbol{L}_1$. This process continues until the data is finally communicated from node x_{k-1} to destination node *v*. In this case we say that this communication required k lightpaths and hence k *hops*. In optical networks, communication using a single lightpath is relatively fast and the process of buffering at intermediate nodes using electronic media is relatively slow. Each stage of electronic buffering at intermediate nodes x_1, x_2, ..., x_{k-1} adds an additional delay in the communication and limits the speed at which data communication can be made from the source node *u* to the destination node *v*. Thus a network which requires, on an average, a lower number of hops to go from any source to any destination is preferable.

The physical topology of an optical network defines how the nodes in the network consisting of computers (also called *endnodes*) and optical routers are connected using optical fibers. A popular technique for implementing multihop networks is the use of multistar networks where each endnode broadcasts to and receives optical signals from a passive optical coupler. To define a lightpath from node *x* to node *y*, the node *x* must broadcast, to the passive optical coupler, at a wavelength $\lambda_{x,y}$ and the node *y* must be capable of receiving this optical signal by tuning its receiver to the same wavelength $\lambda_{x,y}$. This scheme allows us to define a lightpath between any pair of nodes. The logical topology of a network defines how different endnodes are connected by lightpaths. It is convenient to represent a logical topology by a directed graph (or digraph) **G** = (**V**, **E**) where **V** is the set of nodes and **E** is the set of the edges. Each node of **G** represents an endnode of the network and each edge (denoted by $x \to y$) represents a lightpath from end node *x* to endnode *y*. In the graph representation of the logical topology of a multihop network, for communication from *u* to *v*, we can represent the composite lightpath $\boldsymbol{L}$ consisting of the lightpaths $\boldsymbol{L}_0$, $\boldsymbol{L}_1$, ... , $\boldsymbol{L}_{k-1}$ by the graph-theoretic path $u \to x_1 \to x_2 \to \ldots \to x_{k-1} \to v$.

If we use the shortest length path for all communication between source-destination pairs, the maximum hop distance is the *diameter* of **G**. It is desirable that the logical topology has a low average hop distance. A rough rule of thumb is that networks with small diameters have a low average hop distance. To ensure a high throughput and low cost, a logical topology should be such that

- **G** has a small diameter
- **G** has a low average hop distance.
- The indegree and outdegree of each node should be low. The last requirement is useful to reduce the number of optical transmitters and receivers.

These conditions may be contradictory since a graph with a low degree usually implies an increase in the diameter of the digraph and hence the throughput rate. Several different logical topologies have been proposed in the literature. An excellent review of multihop networks is presented in [3].

Both regular and irregular structures have been studied for multihop structures [4],

[5], [6], [7], [8], [9]. Existing regular topologies(e.g., shuffle nets, de Bruijn graphs, torus, hypercubes) are known to have simple routing algorithms and avoid the need of complex routing tables. Since the diameter of a digraph with n nodes and maximum outdegree d is of $O(log_d\, n)$, most of the topologies attempt to reduce the diameter to $O(log_d\, n)$. One problem with these network topologies is that the number of nodes in the network must be given by some well-defined formula involving network parameters. This makes it impossible to add nodes to such networks in any meaningful way and the network is not scalable. This problem has been avoided in the GEMNET architecture [3], [10], which generalizes the shufflenet [6] to allow any number of nodes. A similar idea of generalizing the Kautz graph has been studied in [11] showing a better diameter and network throughput than GEMNET. Both these topologies are given by regular digraphs.

One simple and widely used topology that has been studied for optical networks is the bidirectional ring network. In such networks, each node has two incoming lightpaths and two outgoing lightpaths. In terms of the graph model, each node has one outgoing edge to and one incoming edge from the preceding and the following node in the network. Adding a new node to such a ring network involves redefining a fixed number of edges and can be repeated indefinitely. This is an attempt to develop a topology which has the advantages of a ring network with respect to scalability and the advantages of a regular topology with respect to low diameter.

In this paper we introduce a new scalable topology for multihop networks where the graph is not, in general, regular. The major advantage of our scheme is that, as a new node is added to the network, most of the existing edges of the logical topology are not changed, implying that the routing schemes between the existing nodes need little modification. The diameter of a network with n nodes is $O(log_d\, n)$ and we need to redefine only $O(d)$ edges. In section 2, we describe the proposed topology and derive its pertinent properties. In section 3 we present a routing scheme for the proposed topology and establish that the diameter is $O(log_d\, n)$. We have conclude with a critical summary in section 4.

2 Scalable Topology for Multihop Networks

In this section we will define the interconnection topology of the network by defining a digraph **G**. In doing this we need two integers - n and d, where n represents the number of nodes in the network and $d \le n$, is used to determine the maximum indegree/outdegree of a node. As mentioned earlier, the digraph is not regular - the indegree and outdegree of a node varies from 1 to d+1.

2.1 Definitions and notations

Let Z_p be the set of all p-digit strings choosing digits from $Z = \{0, 1, 2, \ldots, d-1\}$ and let any string of Z_p be denoted by $x_0 x_1 \ldots x_{p-1}$. We divide Z_p into p+1 sets $S_0, S_1, \ldots, S_p$ such that all strings in Z_p having x_j as the left-most occurrence of 0 is included in S_j, $0 \le j < p$ and all strings with no occurrence of 0 (i.e. $x_j \ne 0$, $0 \le j < p$) are included in S_p. We now define an ordering relation between every pair of strings in Z_p. Each string in S_i is smaller than each string in S_j if $i < j$. For two strings $\sigma_1, \sigma_2 \in S_j$, $0 \le j \le p$, if $\sigma_1 = x_0 x_1 \ldots x_{p-1}$, $\sigma_2 = y_0 y_1 \ldots y_{p-1}$ and t is the smallest integer such that $x_t \ne y_t$ then $\sigma_1 < \sigma_2$ if $x_t < y_t$.

Let k be the integer such that $d^k < n < d^{k+1}$. We note that, for a string in Z_{k+1}, $|S_{k+1}| = (d-1)^{k+1}$ and $|S_j| = (d-1)^j d^{k-j}$, $0 \le j \le k$.

Definition 1. Let σ_1 be a string of Z_p, where $\sigma_1 = x_0 x_1 \ldots x_i x_{i+1} \ldots x_{p-1} \in S_i$, $0 \le i < p$, so that i is the left-most digit which is 0.

For $0 \le i < p - 1$: We will call the string σ_2 obtained by interchanging the digits in the ith and the (i+1)th position in σ_1, the *image* of σ_1. Clearly, if $x_{i+1} = 0$, σ_1 and σ_2 represent the same string. Otherwise, $\sigma_2 \in S_{i+1}$. An *image* is not defined for any string in group S_p or in group S_{p-1}.

We consider a network of n nodes, where $d^k < n < d^{k+1}$. We will represent each node of the interconnection topology by a distinct string $x_0 x_1 \ldots x_k$ of Z_{k+1}. As $d^k < n < d^{k+1}$, all strings of Z_{k+1} will not be used to represent the nodes in **G**. We will use the n smallest strings from Z_{k+1} to represent the nodes of **G**. We will use S_M to denote the set containing the largest string. We will use the term *used* string to denote a string of Z_{k+1} which has been already used to represent some node in **G**. We will call all other strings of Z_{k+1} as *unused* strings. We will use a node and its string representation interchangeably.

Definition 2. We consider a string $\sigma = x_0 x_1 \ldots x_k \in Z_{k+1}$. We will call a string $x_0 x_1 \ldots x_{i-1}$ ($x_j x_{j+1} \ldots x_k$) as a *prefix(suffix)* of the string $\sigma = x_0 x_1 \ldots x_k$.

Definition 3. A *pivot* in a string $\sigma = x_0 x_1 \ldots x_k$ is the left-most 0 in the string. If $\sigma \in S_j$, $0 \le j \le k$, then x_j is the pivot for σ. A pivot is not defined for a string $\sigma \in S_{k+1}$.

Properties 1 -3 given below are for strings used to denote nodes in a network with n nodes so that these strings are all in Z_{k+1} .

Property 1. All strings of S_0 are used strings.

Property 2. If $\sigma \in S_j$ is an used string, then all strings of $S_0, S_1, \ldots, S_{j-1}$ are also used strings.

Property 3. If $\sigma_1 = 0x_1 \ldots x_k$, $x_1 \neq 0$ and σ_2, the image of σ_1, is an unused string, then all strings of the form $x_1 x_2 \ldots x_k j$, $0 \leq j \leq d-1$ are unused strings.

The proofs for Properties 1 - 3 are trivial and are omitted.

2.2 Proposed Interconnection Topology

We now define the edge set of the digraph **G**. Let any node *u* in **G** be represented by $u = x_0 x_1 \ldots x_k$. The outgoing edges from node *u* can be of four types and are defined as follows:

Type A: There is an edge $u = x_0 x_1 x_2 \ldots x_k \rightarrow x_1 x_2 \ldots x_k j$ whenever $x_1 x_2 \ldots x_k j$ is an used string, for some $j \in Z$,

Type B: There is an edge from $u \rightarrow v$, where *v* be the image of *u*, as defined earlier, whenever the following conditions hold:

a) $u \in S_i$, $0 \leq i < k-1$

b) *v* represents an used string

c) $u \neq v$ and

d) a type A edge $u \rightarrow v$ does not exist in the network.

Type C: There is an edge $u = 0x_1 x_2 \ldots x_k \rightarrow 0y_1 \ldots y_k$, where $x_1 x_2 \ldots x_k, y_1 y_2 \ldots y_k \in Z_k$ and $y_1 y_2 \ldots y_k$ is the image of $x_1 x_2 \ldots x_k$ whenever the following conditions hold:

b) $x_1 \neq 0$ and $x_1 0 x_2 \ldots x_k$, the image of *u*, is an unused string

or

c) $x_1 = 0$

Type D: There is an edge $0x_1 x_2 \ldots x_k \rightarrow 0x_2 \ldots x_k j$ for all $j \in Z$ whenever the following conditions hold:

a) $x_1 \neq 0$ and

b) $x_1 0 x_2 \ldots x_k$, the image of *u*, is an unused string

We note that if $u \in S_j$, $0 < j \leq k$ node $v = x_1 x_2 \ldots x_k j$ always exists (from property 2, since $v \in S_{j-1}$).

As an example, we show a network for *d* = 2, *k* = 2 with 5 and 6 nodes respectively in figure 1(a) and (b). We have used a solid line for an edge of the Type A, a line of dots for edges of Type B and a line of dashes and dots for edges of Type D. In this small network there are no edges of Type C. We note that Type D edges are the only ones that may need to be redefined as more nodes are added to the network. the edge from 010 to 100 satisfies the condition for both an edge of the type $x_0x_1x_2\ldots x_k \rightarrow x_1x_2\ldots x_kj$ and an edge of the type $0x_1x_2\ldots x_k \rightarrow x_10x_2\ldots x_k$.

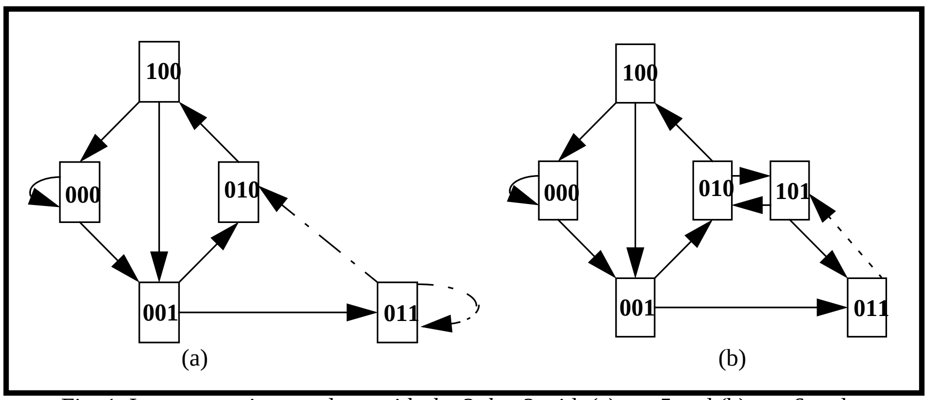

Fig. 1. Interconnection topology with d = 2, k = 2 with (a) n = 5 and (b) n = 6 nodes.

2.3 Limits on Nodal Degree

In this section, we derive the upper limits for the indegree and the outdegree of each node in the network. We will show that, by not enforcing regularity, we can easily achieve scalability. As we add new nodes to the network, minor modifications of the edges in the logical topology suffice, in contrast to large number of changes in the edge-set as required by other proposed methods.

Theorem 1: In the proposed topology, each node has an outdegree of up to d+1.

Proof: Let u be a node in the network given by $x_0x_1\ldots x_k \in S_j$ and let w be the image of u, if an image can be defined for the node. We consider the following three cases:

i) $0 < j \leq k$: For every v given by $x_1x_2\ldots x_kt$ for all t, $0 \leq t \leq d-1$ is an used string since $v \in S_{j-1}$. Therefore the edge $u \rightarrow v$ exists in the network. There are the only d such edges from u. In addition there may be one outgoing edge $u \rightarrow w$ from u to its image w. Hence, u has a minimum outdegree d and maximum outdegree d+1.

ii) $j = 0$: According to our topology defined above, u will have an edge to $x_1x_2\ldots x_kj$ whenever $x_1x_2\ldots x_kj$ is an used string for some $j \in Z$. We have two subcases to consider:

- If p ($p > 0$) of the strings $x_1 x_2 \ldots x_k j$ are used strings, for some j, $0 \le j < d$ and w is also an used string, then u has edges to all the p nodes with used strings of the form $x_1 x_2 \ldots x_k j$ and to w. Hence u has outdegree $p + 1$. So, u has an outdegree of at least 1 and at most d+1.
- Otherwise, if w is an unused string, then all strings of the form $x_1 x_2 \ldots x_k j$ are unused strings (**Property 3**) and u has d outgoing edges of Type D to nodes of the form $0 x_2 x_3 \ldots x_k j$, $0 \le j < d$. and one outgoing edge of Type C. Hence u has outdegree d+1.

iii) $j = k + 1$: If p of the strings $x_1 x_2 \ldots x_k j$ are used strings, for some j, $0 \le j < d$, then u has outdegree of p. We note that $x_1 x_2 \ldots x_k 0 \in S_k$ is an used string. Therefore $1 \le p \le d$, and u has an outdegree of at least 1 and at most d, since an image of u is not defined in this case.

Theorem 2. In the proposed topology, each node has an indegree of up to d+1.

Proof: Let us consider the indegree of any node v given by $y_0 y_1 \ldots y_k \in S_j$. As described in 2.2, there may be four types of edges to a node v as follows:

- An edge (Type A) $t y_0 y_1 \ldots y_{k-1} \to y_0 y_1 \ldots y_k$ whenever $t y_0 y_1 \ldots y_{k-1}$ is an used string, for some $t \in Z$. There may be at most d edges of this type to v.
- There may be an edge (Type B) $u \to v$, where v is the image of u.
- If $y_0 = 0$, v may have one incoming edge of Type C.
- If $y_0 = 0$ and $t y_0 y_1 \ldots y_{k-1}$ is an unused string for some $t \in Z$, there is an edge (Type D) $0 t y_1 \ldots y_{k-1} \to y_0 y_1 \ldots y_k$. There may be at most d such edges to v.

We have to consider 2 cases, j = 0 and j > 0.

If j > 0, v can have at most one incoming edge $u \to v$, where v is the image of u and up to d edges of the type $t y_0 y_1 \ldots y_{k-1} \to y_0 y_1 \ldots y_k$. Therefore, the maximum possible indegree of v is d+1.

If j = 0, an edge of the type $0 t y_1 \ldots y_{k-1} \to y_0 y_1 \ldots y_k$ exists if and only if the corresponding edge of $t y_0 y_1 \ldots y_{k-1} \to y_0 y_1 \ldots y_k$ (Type A) does not exist in the network. Therefore, there is always a total of exactly d incoming edges to v of these two types. Node $v \in S_0$ may also have one incoming edge of Type C. Therefore, the maximum indegree is d+1.

2.4 Node Addition to an Existing Network

In this section we consider the changes in the logical topology that should occur when a new node is added to the network. We show that at most $O(d)$ edge changes in **G** would suffice when a new node is added to the network. When a multistar implementation is considered, this means $O(d)$ retunings of transmitters and receivers, whereas for a wavelength routed network, this means redefinition of $O(d)$ lightpaths. In contrast, other proposed topologies [10], [11] require $O(nd)$ number of edge modifications. As discussed in the previous section, the nodes are assigned the smallest strings defined earlier. When adding a new node u, we will assign the smallest unused string to the newly added node. Let the smallest unused string be $x_0x_1\ldots x_k \in S_j$. To add a node we have to consider the following two cases:

Case i) $0 < j \le k$: Let $v = x_1x_2\ldots x_kt$, $0 \le t \le d-1$, then $v \in S_{j-1}$ and must exist in the network. Let $w_0 = 0x_0x_1\ldots x_{k-1}$ then $w_0 \in S_0$ and is guaranteed to exist in the network. Let w_c be a node in the network, such that u is the image of w_c then $w_c \in S_{j-1}$ and is guaranteed to exist in the network.

- We add an edge $u \to v$ for each node $v = x_1x_2\ldots x_kt$ in the network.
- We add a new edge $w_0 = 0x_0x_1\ldots x_{k-1} \to u$ to the network
- If $w_c \ne w_0$, we add an edge $w_c \to u$ to the network.
- If $u \in S_1$, we delete the edge $w_c \to v$ for each node $v = 0x_2\ldots x_kt$ in the network.

Every string $tx_0x_1\ldots x_{k-1}$, $1 \le t \le d-1$, $w \in S_{j+1}$ and is an unused string. Therefore w_0 and w_c are the only predecessors of u. Also, if w is the image of u, then $w \in S_{j+1}$ and does not exist in the network.

Case ii) $j = k+1$: Let $v = x_1x_2\ldots x_kt$, $0 \le t \le p-1$ be used strings in the network. Also, let $w = tx_0x_1\ldots x_{k-1}$, $0 \le t \le q-1$ be used strings

- We add a new edge $w \to u$ to the network for each $w = tx_0x_1\ldots x_{k-1}$, $0 \le t \le q-1$.
- We add a new edge $u \to v$ to the network for each $v = x_1x_2\ldots x_kt$, $0 \le t \le p-1$.

We note that $w_0 = 0x_0x_1\ldots x_{k-1} \in S_0$ is an used string. Therefore, there is at least one w such that $w \rightarrow u$ exists. Similarly, $x_1x_2\ldots x_k0 \in S_k$ is an used string. Therefore, there is at least one v such that $u \rightarrow v$ exists.

Fig. 2.(a) shows again the network with 6 nodes given in Figure 1. We choose the smallest unused string $u = 110$ to represent the new node being inserted. The following new edges are added to the network.

i) The node u will have outgoing edges of Type A to nodes 100 and101.

ii) The node 110 is the image of node 101. Hence there will be a Type B edge from 101 to 110.

iii) Finally, there will be a Type A edge from 011 to 110.

The final network with 7 nodes is shown in Fig. 2.(b).

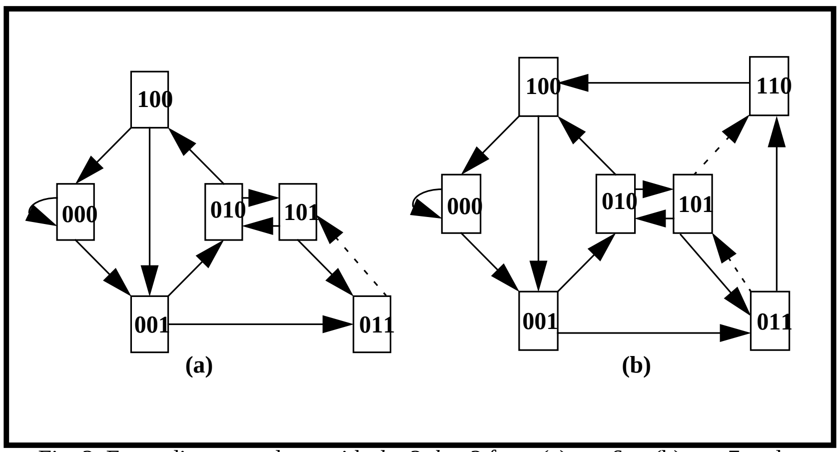

Fig. 2. Expanding a topology with d = 2, k = 2 from (a) n = 6 to (b) n = 7 nodes.

3 Routing strategy

In this section, we present the routing scheme in the proposed topology from any source node ***S*** to any destination node ***D***. Let ***S*** be given by the string $X = x_0x_1\ldots x_k \in S_j$ and ***D*** be given by the string $Y = y_0y_1\ldots y_k \in S_t$. In general, a path from ***S*** to ***D*** may require a number of hops, where each hop represents a traversal of an edge in the network. We will first informally describe the routing scheme and then illustrate it through an example. A formal proof of correctness is omitted due to lack of space. The outline of the routing algorithm is given below.

```
Find_path

(digits_to_be_inserted, pivot_position) ←
       Find-best-suffix-prefix(source, destination);
current-node  ←  source
path  ←  List consisting of source only

While (current-node != destination)
(current-node,path,pivot-position) ←
     Insert-digits(path,digits_to_be-inserted,pivot-position);
(current-node, path, Pivot-position)  ←  Move-pivot(path, pivot-position);
EndWhile;
return path;
```

In this section, we will explain the three main functions, *Find-best-suffix-prefix*, *Insert-digits* and *Move-pivot* through an example. The details of the functions are omitted due to lack of space.

The function *Find-best-suffix-prefix* first finds the length (L1) of the longest set of digits from source string *X* which can also be used as digits in destination string *Y*. The suffix of *X* of length L1 must contain the same digits as the corresponding prefix of *Y* of length L1 and the ordering of the digits may differ only in the position of the *pivot*. This function determines the position of this pivot (pivot position) in the source string and the list of new digits that must be shifted in (list of digits to be inserted) to obtain the destination string.

The function *Insert-digits*, calculates the next node in the path by traversing an edge of Type A or Type D from the current node and inserting the next digit from the list of digits to be inserted. This is done only if the following conditions are satisfied.

a. the destination has not been reached

b. the next node calculated by *Insert-digit***s** exists in the network

c. there is at least one more digit to be inserted and

d. the left-most digit of the current node will not be used as part of destination string *Y.*

The function *Move-pivot* calculates the next node by traversing an edge of Type B from the current node. This essentially causes the pivot to move one digit to the right. This may be necessary for one of two reasons:

i) the next node calculated in *Insert-digits* does not exist or

ii) it is necessary to move the pivot to its proper location with respect to the destination string *Y.*

The next node calculated in this step is used as part of the path only if it exists in the network.

Example. We consider the topology of Fig. 2(a). Let, the source node be *S*=001 and the destination node be *D*=011. We can find a path from *S* to *D*, by following edges of Type A from each node to the next and inserting successive digits of the destination. The final path is given by $P_1 = 001 \rightarrow 010 \rightarrow 101 \rightarrow 011$. Since each node in the path exists in the network, P1 is a valid path.

We now consider a second example where *S*=011 and *D*=000. We try to follow a Type A edge from node 011 to node 110. But 110 is an *unused* string and the corresponding node does not exist in the network. Therefore, we must first follow an edge of Type B from 011 to its image 101. The path from node 101 to node 000 can be constructed using only Type A edges as in the previous example. The final path is given by $P_2 = 011 \rightarrow 101 \rightarrow 010 \rightarrow 100 \rightarrow 000$.

Theorem 3. The function *Find-path* will always perform correctly if there is no node *u* in the network such that $u \in S_{k+1}$.

We omit the proof, due to lack of space.

Theorem 4. The function *Find-path* will find a path of at most 2k+1 hops, if there is no node *u* in the network such that $u \in S_{k+1}$.

Proof.

Part 1: The maximum possible number of digits in list of digits to be inserted is k+1. So, Insert-digits introduces at most k+1 hops traversing edges of Type A or Type D.

Part 2: We consider a source string $X = x_0 x_1 \ldots x_k$ with *m*+1 0's. Let the left-most 0 occur in the *i*th position, the next 0 in the $(i+p_1)$th position, the next in the $(i+p_2)$th position and so on until the last 0 in the $(i+p_m)$th position. Each of these 0's may become the pivot in the function move-pivot. Using the j^{th} 0 as the pivot, we can traverse at most $(i+p_{j+1})-(i+p_j)-1$ edges of type B. Therefore, the maximum number of Type B edges traversed by *Move-pivot* is

$(i+p_1)$-i-1 + $(i+p_2)$-i+p_1)-1 + $(i+p_3)-(i+p_2)-1$ ... + $k-(i+p_m)-1 = k-(m+1)$

Therefore, the maximum number of hops = $(k+1) + (k-m-1) = 2k-m = 2k$, for m=0.

Theorem 5. If a network contains all nodes in $S_0, S_1, \ldots, S_k$ then

- there exists an edge $\boldsymbol{S} \rightarrow \gamma = x_1 x_2 \ldots x_k 0$ and
- $\sigma(\gamma, \boldsymbol{D})$ represents a path from γ to $\boldsymbol{D}$ of length that cannot exceed k+1.

Proof. Since the network contains all nodes in $S_0, S_1, \ldots, S_k$, $\gamma \in S_j$ for some j, $j \le k$ and must exist. Our topology (section 2.2) ensures that the edge $\boldsymbol{S} \rightarrow \gamma$ exists. The path given below consists only strings belonging to groups S_i, $0 \le i \le k$ and hence are used strings:

$x_1 x_2 \ldots x_k 0 \rightarrow x_2 \ldots x_k 0 y_0 \rightarrow x_3 \ldots x_k 0 y_0 y_1 \rightarrow \ldots \rightarrow y_0 y_1 \ldots y_k$

The number of edges in the above path is *k*+ 1, hence the proof.

Theorem 6: The diameter of the network in 2*k*+1.

If there is no node *u* in the network such that $u \in S_{k+1}$, then *Find-path* will return a path of 2*k* or less hops (**Theorem 4**).

If there is a node *u* in the network such that $u \in S_{k+1}$, then **Theorem 5** can be used to find a path of *k*+1 hops from ***S*** to ***D***, hence the proof.

4 Conclusions

In this paper we have introduced a new graph as a logical network for multihop networks. We have shown that our network has an attractive average hop distance compared to existing networks. The main advantage of our approach is the fact that we can very easily add new nodes to the network. This means that the perturbation of the network in terms of redefining edges in the network is very small in our architecture. The routing scheme in our network is very simple and avoids the use of routing tables.

References

[1] R. Ramaswami and K. N. Sivarajan, “Opical Networks: A Practical Perspective”, Morgan-Kaufmann, 1998.

[2] B. Mukherjee, “Optical Communication Networks', McGraw-Hill, 1997.

[3] B. Mukherjee, "WDM-based local lightwave networks part II: Multihop systems," *IEEE Network, vol. 6*, pp. 20-32, July 1992.

[4] K. Sivarajan and R. Ramaswami, “Lightwave Networks Based on de Bruijn Graphs,” *IEEE/ACM Transactions on Networking, Vol. 2, No. 1*, pp. 70-79, Feb 1994.

[5] K. Sivarajan and R. Ramaswami, “Multihop Networks Based on de Bruijn Graphs,” *IEEE INFOCOM ‘91*, pp. 1001-1011, Apr. 1991.

[6] M. Hluchyj and M. Karol, “ShuffleNet: An application of generalized perfect shuffles to multihop lightwave networks,” *IEEE/OSA Journal of Lightwave Technology*, vol. 9, pp.1386-1397, Oct. 1991.

[7] B. Li and A. Ganz, “Virtual topologies for WDM star LANs: The regular structure approach,” *IEEE INFOCOM ‘92*, pp.2134-2143, May 1992.

[8] N. Maxemchuk, “Routing in the Manhattan street network,” *IEEE Trans. on Communications*, vol. 35, pp. 503-512, May 1987.

[9] P. Dowd, “Wavelength division multiple access channel hypercube processor interconnection,” *IEEE Trans. on Computers*, 1992.

[10] J. Innes, S. Banerjee and B. Mukherjee, “GEMNET : A generalized shuffle exchange based regular, scalable and modular multihop network based on WDM lightwave technology”, IEEE/ACM Trans. Networking, Vol 3, No 4, Aug 1995.

[11] A. Venkateswaran and A. Sengupta, “On a scalable topology for Lightwave networks”, Proc IEEE INFOCOM’96, 1996.

}