

# Lightwave Networks Based on de Bruijn Graphs

Kumar N. Sivarajan, *Member, IEEE*, and Rajiv Ramaswami, *Member, IEEE*

**Abstract**—We propose de Bruijn graphs as logical topologies for multihop lightwave networks. After deriving bounds on the throughput and delay performance of any logical topology, we compute the throughput and delay performance of de Bruijn graphs for two different routing schemes and compare it with our bounds and the performance of shufflenets. For a given maximum nodal in- and out-degree and average number of hops between stations, a logical topology based on a de Bruijn graph can support a larger number of stations than a shufflenet and this number is close to the maximum that can be supported by any topology. We also propose de Bruijn graphs as good physical topologies for wavelength routing lightwave networks consisting of all-optical routing nodes interconnected by point-to-point fiber links. The worst-case loss experienced by a transmission is proportional to the maximum number of hops (diameter). For a given maximum nodal in- and out-degree and diameter, a physical topology based on a de Bruijn graph can support a large number of stations using a relatively small number of wavelengths.

**Index Terms**—Optical networks, multihop, de Bruijn, shuffle.

## I. INTRODUCTION

THIS PAPER IS ABOUT lightwave network topologies. We distinguish between two types of topologies—the *physical topology* and the *logical topology*. By the *physical topology* we mean the actual underlying network topology which is commonly a broadcast star or bus. It could also be a wavelength routing network, which consists of point-to-point links interconnecting all-optical routing nodes, as will be described later. On any underlying physical topology, one can impose a carefully selected connectivity pattern that provides dedicated connections between certain pairs of stations. Traffic destined to a station that is not directly receiving from the transmitting station must be routed through intermediate stations. This overlaid topology is referred to as the *logical topology*. We use the term *multihop* to refer to any network where traffic may have to be routed through intermediate stations and electronically processed in between. This paper considers de Bruijn graphs as logical topologies for multihop networks and as physical topologies for wavelength routing networks.

The first part of this paper deals with logical topologies. A logical topology can be superposed on a physical topology by using many channels at different wavelengths. This technique, wavelength-division multiplexing (WDM), significantly

enhances the network capacity. In a simple example, the underlying physical topology is a broadcast passive star. Each station is provided with two transmitters and two receivers. Each transmitter in the network is at a different wavelength. Using its two transmitters, a station can be connected directly to two other stations whose receivers are tuned to the transmitted wavelengths.

The advantages of using a multihop logical topology are as follows [1]–[3]: It is possible to create the logical topology to reflect the traffic patterns in the network, and the logical topology can be configured so as to simplify the routing and flow control in the network and to improve the delay-throughput performance. Moreover, the logical topology can be implemented using fixed-tuned transmitters and receivers, unlike other “single-hop” architectures that require tunable components and have to deal with collision and contention resolution at the media-access layer.

The second part of the paper considers the design of the underlying physical topology. Conventional topologies such as the star and the bus are broadcast topologies. Wavelength routing networks use all-optical routing nodes interconnected by point-to-point fiber links, and avoid the splitting loss problems associated with broadcast topologies and also reuse wavelengths in the network.

For both the logical and the physical cases, we find that de Bruijn graphs provide a good class of topologies. We compare them with the class of topologies called shufflenets [2]–[4]. Shufflenets provide a regular logical topology with simple addressing and self-routing. We prove that topologies based on de Bruijn graphs can support a larger number of stations for the same performance measures while retaining the simple addressing and self-routing properties of shufflenets.

For the physical topology, we show that the number of wavelengths required to support a given traffic requirement is much smaller than in a broadcast star network (with or without a superimposed logical topology).

The outline of the rest of the paper is as follows. The next section considers logical topologies. After describing the performance measures that we will use to evaluate logical topologies and deriving bounds on them, we introduce de Bruijn graphs and discuss some of their relevant features. We then compare the performance of de Bruijn logical topologies with our bounds and with shufflenets. The following section describes wavelength routing networks. We again consider de Bruijn graphs as a possible physical topology, and also discuss the wavelength assignment problem in the network.

## II. LOGICAL TOPOLOGIES

We can represent the logical topology of a network by a directed graph, where each station is represented by a node in

Manuscript received November 1990; revised June 1993; transferred from IEEE TRANSACTIONS ON COMMUNICATIONS by IEEE/ACM TRANSACTIONS ON NETWORKING Editor-in-Chief Jim Kurose. This paper was presented in part at IEEE INFOCOM'91, Bal Harbour, FL, 1991.

The authors are with the IBM T.J. Watson Research Center, Yorktown Heights, NY 10598 (email: sivaraj@watson.ibm.com; rajiv@watson.ibm.com).

IEEE Log Number 9400262.

the graph and there is an edge from node A to node B in the graph if station A can transmit to station B.

From a practical point of view, we can provide only a limited number of transmitters or receivers to each station. This limits the number of other stations that a station transmits to, and hence the degree of each node in the logical topology. We define the loading on an edge as the number of source-destination pairs that use the edge to communicate.

The two metrics by which we shall evaluate the performance of a logical topology are the *average end-to-end packet delay* and the *network throughput*. The throughput is the least upper bound of the offered traffic for which the average delay is finite. The delay experienced by a packet is the sum of two components: a propagation delay component and a component due to queueing delays at all the nodes the packet passes through (this includes the packet transmission time). At low offered loads, the waiting times at the nodes are small and the average delay is then equal to the average number of hops multiplied by the sum of the propagation delay per hop and the packet transmission time. As the offered load increases, the queueing delay component increases and at high enough loads, it becomes much larger than the propagation delay. In the high-speed metropolitan-area environment, since the propagation delay is usually much larger than the packet transmission time, this dominance occurs only at heavy loads very close to the throughput. Hence minimizing the average number of hops minimizes the average delay at low offered loads. The throughput of the network is determined by the edge with the maximum loading, assuming all edges have the same capacity. Lower the maximum loading, higher the throughput. Both the delay and throughput will depend on the routing scheme used.

We consider only the uniform traffic case, i.e., a new packet arriving at a station is equally likely to be destined to any one of the other stations in the network. Let  $C$  denote the capacity of a link<sup>1</sup> in bits/s. Assume all links have the same capacity. For each source-destination pair, let packets arrive at the source as a Poisson process with rate  $\lambda$  packets/s. Packet lengths are assumed to be exponentially distributed with mean length  $1/\mu$  bits. Assuming infinite buffers at each node and assuming that traffic patterns at different nodes are independent, the service discipline on each link can be modelled as an M/M/1 queue [5]. Let  $L_i$  denote the loading on link  $i$ ,  $\bar{L}$  the average loading on a link, and  $L_{\max}$  the maximum loading on a link. The total arrival rate of packets for link  $i$  is  $L_i\lambda$  and hence, the average queueing delay for a packet at link  $i$ ,  $f(L_i)$ , in seconds/packet is given by

$$f(L_i) = \frac{L_i}{\mu C - L_i\lambda}, \quad \text{for } 0 \leq \lambda \leq 1/L_i,$$

and the average queueing delay for a packet through the network,  $\bar{d}(\lambda)$ , in seconds/packet is given by [5]

$$\bar{d}(\lambda) = \frac{1}{N(N-1)} \sum_{i=1}^M \frac{L_i}{\mu C - L_i\lambda}, \quad \text{for } 0 \leq \lambda \leq 1/L_{\max}, \quad (1)$$

where  $N$  is the number of nodes and  $M$  is the number of links. For convenience, we shall use the *normalized delay*,  $\bar{d}\mu C$  and

<sup>1</sup> In this paper, we use the words *link* and *edge* interchangeably.

the normalized offered load per station  $(N-1)\lambda/(\mu C)$ . The *normalized throughput* per station  $\gamma$  is the least upper bound on the normalized offered load for which  $\bar{d}$  is finite. Hence,

$$\gamma = (N-1)/L_{\max}.$$

We define the length of a path to be the number of edges in that path and the *diameter* of any graph, denoted by  $D$ , as the maximum, over all pairs of nodes in the graph, of the length of the shortest path between a pair of nodes. The diameter of a graph whose nodes are identified with the stations in a network is the maximum number of hops required for any two stations in that network to communicate. Let  $n(i)$  denote the number of source-destination pairs in the graph for which the shortest path between the source and destination has  $i$  hops. (We define  $n(0) = N$ .) The average number of hops (for shortest-path routing) is then

$$\bar{H} = \frac{1}{N(N-1)} \sum_{i=0}^D i n(i). \quad (2)$$

Observe that for any graph and any routing scheme, the average link loading,  $\bar{L}$  satisfies,

$$M\bar{L} = N(N-1)\bar{H} = \sum_{i=0}^D i n(i). \quad (3)$$

#### A. General Bounds

**Theorem 1:** For any directed graph with  $N$  nodes and maximum out-degree  $\Delta$ , let

$$\bar{H}_{\min}(N, \Delta) \stackrel{\text{def}}{=} \begin{cases} \frac{\Delta - \Delta^{m+1} + Nm(\Delta-1)^2 + m(\Delta-1)}{(N-1)(\Delta-1)^2}, & \Delta \geq 2, \\ N/2, & \Delta = 1, \end{cases} \quad (4)$$

where  $m$  is the largest integer that satisfies

$$N \geq 1 + \Delta + \Delta^2 + \dots + \Delta^{m-1} = \frac{\Delta^m - 1}{\Delta - 1}. \quad (5)$$

Then,  $\bar{H} \geq \bar{H}_{\min}(N, \Delta)$ .

*Proof:* We prove the theorem only for  $\Delta \geq 2$ . The proof for  $\Delta = 1$  is similar.

Let the maximum distance between any pair of nodes in the graph be denoted by  $D$  (the diameter of the graph). For each node, the number of nodes with minimum distance  $\leq k$  is  $\leq 1 + \Delta + \Delta^2 + \dots + \Delta^k$ ; hence

$$\sum_{j=0}^k n(j) \leq N \frac{\Delta^{k+1} - 1}{\Delta - 1}, \quad 0 \leq k \leq D-1. \quad (6)$$

Since the graph has diameter  $D$  and there are a total of  $N^2$  source-destination pairs,

$$\sum_{j=0}^D n(j) = N^2. \quad (7)$$

Also,

$$\sum_{j=0}^k n(j) \leq N^2, \quad 0 \leq k \leq D-1. \quad (8)$$

Combining (6) and (8) yields

$$\sum_{j=0}^k n(j) \leq \min \left\{ N^2, N \frac{\Delta^{k+1} - 1}{\Delta - 1} \right\}, \quad 0 \leq k \leq D - 1$$

$$\leq \begin{cases} N \frac{\Delta^{k+1} - 1}{\Delta - 1}, & 0 \leq k \leq m - 1, \\ N^2, & m \leq k \leq D - 1, \end{cases} \quad (9)$$

where  $m$  is defined as in (5). (Note that  $m \leq D$ .) Subtracting (9) from (7), we get,

$$\sum_{j=k+1}^D n(j) \geq \begin{cases} N^2 - N \frac{\Delta^{k+1} - 1}{\Delta - 1}, & 0 \leq k \leq m - 1, \\ 0, & m \leq k \leq D - 1. \end{cases}$$

Summing from  $k = 0$  to  $D - 1$  and using (2), we get

$$\bar{H} \geq \frac{\Delta - \Delta^{m+1} + Nm(\Delta - 1)^2 + m(\Delta - 1)}{(N - 1)(\Delta - 1)^2}.$$

□

This bound is a variation of the Moore bound and a proof for the case of a regular directed graph with in- and out-degree  $\Delta$  appears in [3]. The directed-cycle graph on  $N$  vertices has  $\Delta = 1$  and satisfies  $\bar{H} = N/2$ .

Note that (4) can be inverted to yield an upper bound,  $N_{\text{up}}(\bar{H}, \Delta)$ , on the number of nodes in any directed graph with a given  $\bar{H}$  and maximum out-degree  $\Delta$ .

Since, by (3),  $M\bar{L} = N(N - 1)\bar{H}$ , we can obtain a lower bound on  $\bar{L}$  and hence on  $L_{\text{max}}$  from any lower bound on  $\bar{H}$ . Specifically, from Theorem 1 and  $M \leq N\Delta$ , we have that for any topology with  $N$  nodes and maximum out-degree  $\Delta$ , and any routing scheme,

$$L_{\text{max}} \geq \bar{L} \geq \frac{(N - 1)\bar{H}_{\min}}{\Delta}. \quad (10)$$

For  $N = 1024$  and  $\Delta = 4$ , this yields  $L_{\text{max}} \geq 1166.75$  and (normalized) throughput per station  $\gamma \leq 0.877$ .

The following theorem provides a lower bound on the delay versus offered traffic characteristic of a given topology.

**Theorem 2:** For a given topology with  $N$  nodes,  $M$  (directed) edges and in- and out-degree  $\leq \Delta$ , the average queueing delay under a given routing scheme satisfies,

$$\bar{d}(\lambda) \geq \frac{\bar{H}\Delta}{\mu C\Delta - (N - 1)\bar{H}\lambda}, \quad \text{for } 0 \leq \lambda \leq 1/L_{\text{max}}. \quad (11)$$

*Proof:* For  $0 \leq \lambda \leq 1/L_{\text{max}}$ ,

$$\begin{aligned} \bar{d}(\lambda) &= \frac{1}{N(N - 1)} \sum_{i=1}^M \frac{L_i}{\mu C - L_i \lambda} \quad (\text{using (1)}) \\ &\geq \frac{\sum L_i}{N(N - 1)(\mu C - \sum \frac{L_i}{M} \lambda)} \\ &\quad (\text{since } f(L) \text{ is convex } \cup \text{ in } L) \\ &\geq \frac{\bar{H}\Delta}{\mu C\Delta - (N - 1)\bar{H}\lambda} \\ &\quad (\text{using } M \leq N\Delta \text{ and } \sum L_i = N(N - 1)\bar{H}). \end{aligned}$$

**Corollary 1:** For a given topology and routing scheme, if  $\bar{H} \geq \bar{H}_{\text{l.b.}}$ ,

$$\bar{d}(\lambda) \geq \frac{\bar{H}_{\text{l.b.}}\Delta}{\mu C\Delta - (N - 1)\bar{H}_{\text{l.b.}}\lambda}.$$

*Proof:* This follows from Theorem 2 and the observation that the r.h.s. in (11) is a monotonically nondecreasing function of  $\bar{H}$ . □

For many topologies and routing schemes, it may be difficult to calculate  $\bar{H}$  exactly but we can still obtain a lower bound on the delay characteristic using Corollary 1 if we can lower-bound  $\bar{H}$ . We will apply this Corollary later to obtain a lower bound on the delay characteristic of shufflenets.

**Corollary 2:** For any topology with  $N$  nodes,  $M$  (directed) edges and in- and out-degree  $\leq \Delta$  and any routing scheme,

$$\bar{d}(\lambda) \geq \frac{\bar{H}_{\min}\Delta}{\mu C\Delta - (N - 1)\bar{H}_{\min}\lambda}.$$

where  $\bar{H}_{\min} = \bar{H}_{\min}(N, \Delta)$  is defined by (4).

*Proof:* From Theorem 1, for any topology with  $N$  nodes,  $M$  (directed) edges and in- and out-degree  $\leq \Delta$  and any routing scheme,  $\bar{H} \geq \bar{H}_{\min}$ . The result now follows from Corollary 1 with  $\bar{H}_{\text{l.b.}} = \bar{H}_{\min}$ . □

The results of this section serve as benchmarks for the performance of any logical topology and we will use them to compare the performance of de Bruijn graphs and shufflenets.

Results similar to Theorem 2 and its corollaries can be derived for any monotonically non-decreasing, convex  $\cup$  measure of the delay other than  $f(L)$  as well.

In the following sections we describe de Bruijn graphs, propose two routing algorithms, present results for the average number of hops, the average and maximum edge-loading, and the delay and throughput performance, and then compare their performance with that of shufflenets.

### B. de Bruijn Graphs

For any positive integers  $\Delta \geq 2$  and  $D \geq 1$ , the de Bruijn graph  $G(\Delta, D)$  [6] is the directed graph with the set of nodes  $\{0, 1, 2, \dots, \Delta - 1\}^D$  with an edge from node  $(a_1, a_2, \dots, a_D)$  to node  $(b_1, b_2, \dots, b_D)$  if and only if  $b_i = a_{i+1}$  for  $1 \leq i \leq D - 1$ .

The de Bruijn graph  $G(\Delta, D)$  has diameter  $D$ . The in-degree and out-degree of every node in  $G(\Delta, D)$  is  $\Delta$ . (Note that  $\Delta$  of the nodes have self-loops, which are present in graph but will not be present in the actual network.) The number of nodes and hence the number of stations, is given by

$$N = \Delta^D.$$

Consider a  $\Delta$ -ary shift register of length  $D$ . There is a one-to-one correspondence between all the possible states of the shift register and the nodes of the de Bruijn graph  $G(\Delta, D)$ . There is an edge joining node  $x_i$  to node  $x_j$  if node (state)  $x_j$  can be reached from state  $x_i$  with one shift (and a new input digit) [7]. In other words, the de Bruijn graph is the state transition diagram of the shift register. For example, the de Bruijn graph  $G(2, 3)$  is shown in Fig. 1.

□

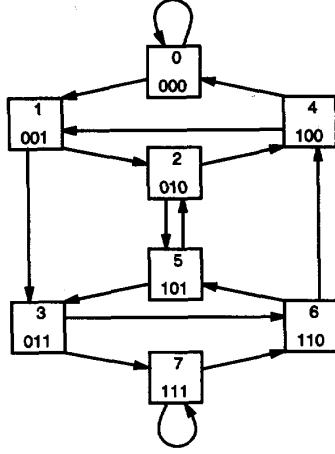


Fig. 1. The de Bruijn graph  $G(2,3)$  (degree = 2 and diameter = 3). Each node can be viewed as the state of a binary (in general  $\Delta$ -ary) shift register with  $D = 3$  stages. There is an edge from a node to all nodes that can be reached in one shift and a new input digit.

From the shift-register analogy, a node (or state) in the de Bruijn graph can be represented by a string (or sequence) of  $D$  digits. An edge from node  $A$  to node  $B$  can be represented by a string of  $D+1$  digits, the first  $D$  digits representing node  $A$  and the last  $D$  digits representing node  $B$ . Similarly, any path of length  $k$  hops can be represented by a string of  $D+k$  digits.

In any graph with maximum out-degree  $\Delta$ , there can be at most  $\Delta$  node-disjoint paths between any pair of nodes. In the de Bruijn graph  $G(\Delta, D)$ , there are  $\Delta - 1$  node-disjoint paths between every pair of nodes [8]. Therefore, networks based on de Bruijn graphs can tolerate up to  $\Delta - 2$  node failures without disruption in communication between any pair of nodes. It has further been shown that in the presence of  $\Delta - 2$  node faults, the diameter of the network increases by at most one hop [8].

**1. Shortest-Path Routing:** A shortest-path algorithm to route from node  $A = (a_1, a_2, \dots, a_D)$  to node  $B = (b_1, b_2, \dots, b_D)$  is as follows. Define the  $shift\_match(i, A, B)$ ,  $0 \leq i \leq D$  operation on the two strings  $A$  and  $B$  to be TRUE if and only if

$$(b_1, b_2, \dots, b_{D-i}) = (a_{i+1}, a_{i+2}, \dots, a_D)$$

and FALSE otherwise. Define  $merge(i, A, B)$ ,  $0 \leq i \leq D$  to be the string (or sequence) of length  $D+i$  given by  $(a_1, \dots, a_D, b_{D-i+1}, \dots, b_D)$ . The routing algorithm is given below.

```

number of hops
i = 0 while (shift_match(i, A, B) is FALSE)
    i = i + 1
end while
shortest_path = merge(i, A, B)

```

**Example:** Consider  $G(2,3)$  and let  $A = (0,0,1)$  and  $B = (1,0,1)$ . Here  $shift\_match(0, A, B)$  and  $shift\_match(1, A, B)$  are FALSE, and  $shift\_match(2, A, B)$  is TRUE. Then  $merge(2, A, B)$  yields  $(0,0,1,0,1)$ . Hence the shortest path is  $(0,0,1) \rightarrow (0,1,0) \rightarrow (1,0,1)$ .

**Theorem 3:** For de Bruijn graphs, the average number of hops for shortest-path routing,  $\bar{H}$  satisfies

$$D \frac{N}{N-1} - \frac{\Delta}{(\Delta-1)^2} + \frac{D}{(\Delta^D-1)(\Delta-1)} \leq \bar{H} \leq D \frac{N}{N-1} - \frac{1}{\Delta-1}, \quad (\Delta \geq 2, D \geq 1).$$

**Proof:** For the de Bruijn graph  $G(\Delta, D)$  the largest integer  $m$  such that

$$N \geq \frac{\Delta^m - 1}{\Delta - 1}$$

is  $m = D$ . Therefore, from Theorem 1, together with  $N = \Delta^D$ , we get the lower bound for  $\bar{H}$ .

The number of source-destination pairs with distance  $k$  is  $N\Delta^k$  for  $0 \leq k \leq D-1$ , since from each node we can reach  $\Delta^k$  nodes in  $k$  hops. Note that some of these source-destination pairs may have a minimum distance less than  $k$ . Therefore (recalling that  $n(i)$  denotes the number of source-destination pairs in the graph for which the shortest path between the source and destination has  $i$  hops),

$$\sum_{j=0}^k n(j) \geq N\Delta^k, \quad 0 \leq k \leq D-1. \quad (12)$$

Since the de Bruijn graph has diameter  $D$  and there are a total of  $N^2$  source-destination pairs,

$$\sum_{j=0}^D n(j) = N^2. \quad (13)$$

Subtracting (12) from (13) yields,

$$\sum_{j=k+1}^D n(j) \leq N^2 - N\Delta^k, \quad 0 \leq k \leq D-1.$$

Summing from  $k = 0$  to  $D-1$  and using (2), we get the upper bound for  $\bar{H}$ .  $\square$

Note that the difference between the upper and lower bounds for  $\bar{H}$  is  $O(1/\Delta^2)$  for large  $\Delta$ . Therefore,  $\bar{H}$  for de Bruijn graphs is arbitrarily close to the minimum achievable  $\bar{H}$  for any graph with the same number of nodes and maximum out-degree  $\Delta$  for sufficiently large  $\Delta$ . In the appendix, we give a recursive technique for calculating  $\bar{H}$  exactly for these graphs.

Table I shows the number of stations and the average number of hops for de Bruijn graphs of different degrees and diameters. Also shown in each row is the upper bound,  $N_{up}$ , on the number of nodes that can be supported in any directed graph with the same average number of hops. Table I shows that even for small, practically feasible values of  $\Delta$ , de Bruijn graphs are fairly close to the optimal topology.

We next determine the edge loading for the shortest-path routing scheme. On the average, a transmission uses  $\bar{H}$  edges, and there are  $N(N-1)$  possible combinations of source and destination nodes, all of which are assumed to be equally likely. The number of edges in the network is  $\Delta N - \Delta$  (excluding the edges from a node to itself). Therefore, the average edge-loading

$$\bar{L} = \frac{\bar{H}N(N-1)}{\Delta N - \Delta} = \frac{\bar{H}}{\Delta} \Delta^D. \quad (14)$$

TABLE I  
NUMBER OF STATIONS  $N$ , AND AVERAGE NUMBER OF HOPS  $\bar{H}$  FOR THE DE BRUIJN GRAPH AS A FUNCTION OF THE DEGREE  $\Delta$  AND THE DIAMETER  $D$ .  $N_{up}$  IS AN UPPER BOUND ON  $N$  FOR THE SAME AVERAGE NUMBER OF HOPS.  $L_{max}$  IS THE MAXIMUM EDGE LOADING AND IS SHOWN HERE FOR SHORTEST-PATH ROUTING (SP), AND LONGEST-PATH ROUTING (LP)

de Bruijn Graph						
$\Delta, D$	$\bar{H}$	$N$	$N_{up}$	$N/N_{up}$	$L_{max}(SP)$	$L_{max}(LP)$
2, 2	1.5000	4	5	0.8000	3	4
2, 3	2.1071	8	9	0.8889	11	12
2, 4	2.8333	16	19	0.8421	29	32
2, 5	3.6492	32	39	0.8205	81	80
3, 2	1.6667	9	10	0.9000	7	6
3, 3	2.4786	27	29	0.9310	31	21
3, 4	3.3861	81	88	0.9205	138	108
3, 5	4.3440	243	266	0.9135	535	405
4, 2	1.7500	16	17	0.9412	9	8
4, 3	2.6399	64	67	0.9552	57	48
4, 4	3.5985	256	269	0.9517	313	256
4, 5	4.5844	1024	1079	0.9490	1589	1280
5, 2	1.8000	25	26	0.9615	11	10
5, 3	2.7277	125	129	0.9690	86	75
5, 4	3.7059	625	647	0.9660	586	500
5, 5	4.7000	3125	3234	0.9663	3711	3125
6, 2	1.8333	36	37	0.9730	13	12
6, 3	2.7823	216	221	0.9774	121	108
6, 4	3.7694	1296	1327	0.9766	985	864
6, 5	4.7665	7776	7966	0.9761	7465	6480

An upper bound on the maximum edge loading for de Bruijn graphs using the shortest-path routing strategy<sup>2</sup> can be obtained by determining the maximum number of paths of length  $\leq D$  that use an edge. We first find the maximum number of paths of length  $k$ ,  $P_k$ , that use a given edge. As noted earlier, an edge can be represented by a string of length  $D+1$  and a path of length  $k$  by a string of length  $D+k$ . So the problem is transformed to that of determining how many strings of length  $D+k$  contain a given string of length  $D+1$ . The string of length  $D+1$  can be positioned at one of  $k$  positions in the string of length  $D+k$  and there are  $\Delta$  choices for each of the remaining  $k-1$  digits in the string of length  $D+k$ . Therefore,

$$P_k \leq k\Delta^{k-1}, \quad 1 \leq k \leq D.$$

Therefore the maximum number of paths of all lengths  $\leq D$  that use an edge is

$$L_{max} = \sum_{k=1}^D P_k \leq \frac{D\Delta^{D+1} - (D+1)\Delta^D + 1}{(\Delta-1)^2} \quad (15)$$

$$\approx \frac{D}{\Delta} \Delta^D \quad \text{for large } \Delta. \quad (16)$$

If  $D \leq \Delta - 1$  we leave it to the reader to verify that the edge  $(0, 1, 2, \dots, D)$  will have a loading equal to  $L_{max}$  with shortest-path routing, proving that the bound is tight in this case.

For  $G(4, 5)$  with 1024 nodes and  $\bar{H} = 4.5844$ ,  $\bar{L} = 1174$  from (14) and  $L_{max} \leq 1593$  from (15). (Exact calculation yields  $L_{max} = 1589$ .)

<sup>2</sup>This bound is also valid for any routing strategy that does not use routes of length  $> D$ .

**2. Longest-Path Routing:** Shortest-path routing minimizes  $\bar{H}$  and hence  $\bar{d}$  at low loads. However, as we saw in the previous section, with shortest-path routing, the loading on some edges is significantly higher than others. In order to increase the throughput, we consider another routing scheme that lowers the maximum edge-loading in most cases, at the cost of increasing the average number of hops.

The routing rule is as follows. In the de Bruijn graph  $G(\Delta, D)$ , to route from a node  $A = (a_1, a_2, \dots, a_D)$ , to a node  $B = (b_1, b_2, \dots, b_D)$ , use the unique  $D$ -hop path  $(a_1, a_2, \dots, a_D, b_1, b_2, \dots, b_D)$ . If this path has any circuits, they are removed from the route. Note that if these circuits are not removed, all routes are of length  $D$  hops. We call this longest-path routing because, before circuits are removed, this routing scheme uses the longest path between a pair of nodes subject to the constraint that its length is  $\leq D$ . With the circuits removed, there may exist other circuit-free routes between nodes that are longer but of length  $\leq D$  hops and hence our algorithm is not strictly a longest-path algorithm.

We now prove that the maximum loading on an edge is  $L_{max} = D\Delta^{D-1}$  for the longest-path routing algorithm. To show that  $L_{max} \leq D\Delta^{D-1}$ , consider the longest-path algorithm without removing circuits. We have to determine the number of paths of length  $D$  that contain a given edge. An edge can be represented by a string of length  $D+1$  and a path of length  $D$  by a string of length  $2D$ . So the problem is transformed to that of determining how many strings of length  $2D$  contain a given string of length  $D+1$ . The string of length  $D+1$  can be positioned at one of  $D$  positions in the string of length  $2D$  and there are  $\Delta$  choices for each of the remaining  $D-1$  digits in the string of length  $2D$ . Therefore,

$$L_{max} \leq D\Delta^{D-1}.$$

We leave it to the reader to verify that the edge  $(1, 0, 0, \dots, 0)$  will have a loading equal to  $D\Delta^{D-1}$  with longest-path routing.

Table I compares the average number of hops and the maximum edge-loading for the two routing schemes. In all cases except  $\Delta = 2, D \leq 4$ , the longest-path scheme has a lower maximum edge-loading and hence a higher throughput. For  $G(4, 5)$  with 1024 nodes, with this scheme, we get  $\bar{H} = 4.9829$  and  $L_{max} = 1280$  while  $L_{max}$  for the shortest-path scheme is 1589.

**3. Delay and Throughput Performance:** Fig. 2 shows the normalized average queueing delay  $\bar{d}\mu C$  plotted against the normalized offered load per station  $(N-1)\lambda/\mu C$  for the 1024-node de Bruijn graph  $G(4, 5)$  for both the shortest-path and longest-path routing schemes. Also shown is the lower bound on the average queueing delay for any topology and routing scheme with  $N = 1024$  and  $\Delta = 4$  computed using Corollary 2. (The fourth curve corresponds to the shufflenet and will be discussed in the following section.) Observe that at low offered loads, the normalized queueing delay is essentially equal to the average number of hops, which corresponds to the transmission time of the packet. The normalized network throughput per station,  $\gamma$ , is 0.779 for longest-path routing and 0.644 for shortest-path routing.

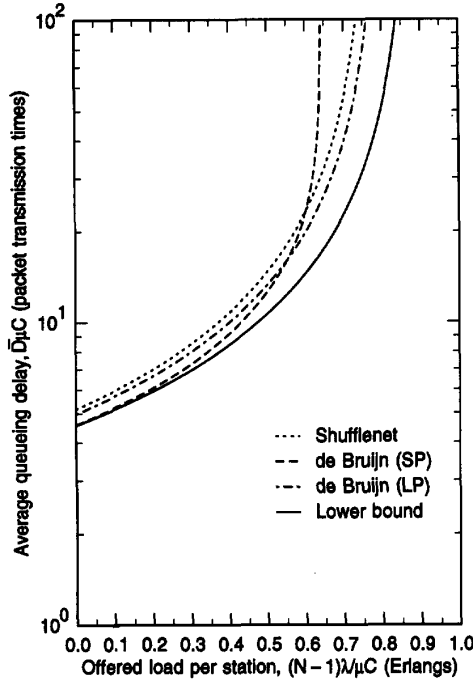


Fig. 2. Normalized average queuing delay versus normalized offered load for the 1024-node de Bruijn graph and shufflenet, both with  $\Delta = 4$ . Both shortest (SP) and longest (LP)-path routing are shown for the de Bruijn graph. A lower bound on the delay for any 1024-node topology with  $\Delta = 4$  is also shown.

At low offered loads, we can use shortest-path routing so that the delay is minimized and at high offered loads, we can obtain a higher throughput by switching to longest-path routing.

### C. Comparison with Shufflenets

The  $(\Delta, k)$ -shufflenet consists of  $k\Delta^k$  stations arranged in  $k$  columns with  $\Delta^k$  stations per column [4]. Adjacent columns are connected in a perfect  $\Delta$ -shuffle [4]. The  $k^{th}$  column is connected back to the first column, also in a perfect  $\Delta$ -shuffle. The  $(2, 2)$ -shufflenet is shown in Fig. 3. The in- and out-degree of the  $(\Delta, k)$ -shufflenet are both  $\Delta$  and the diameter is  $D = 2k - 1$ . Thus, the number of stations that can be supported by the  $(\Delta, k)$ -shufflenet is

$$N = k\Delta^k = \frac{D+1}{2} \Delta^{\frac{D+1}{2}}.$$

The average number of hops between nodes is [3]

$$\bar{H} = \frac{k\Delta^k(\Delta-1)(3k-1) - 2k(\Delta^k-1)}{2(\Delta-1)(k\Delta^k-1)}. \quad (17)$$

Table II shows the number of stations that can be supported by shufflenets along with the average number of hops, for different degrees and diameters. Also shown is the upper bound,  $N_{up}$ , on the number of stations that can be supported by any topology for the same average number of hops. Shufflenets perform well when the diameter is small (and the number of stations is small). A comparison with Table I shows that logical

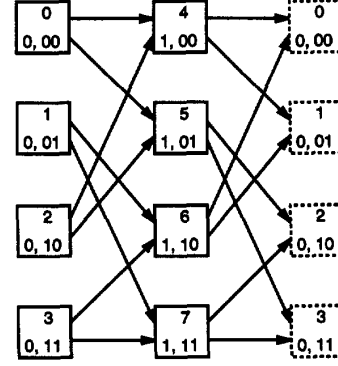


Fig. 3. The  $k\Delta^k = 8$ -node shufflenet consisting of  $k = 2$  columns with  $\Delta^k = 4$  nodes per column arranged in a perfect shuffle.

TABLE II  
NUMBER OF STATIONS  $N$ , AND AVERAGE NUMBER OF HOPS  $\bar{H}$  FOR SHUFFLENETS AS A FUNCTION OF THE DEGREE  $\Delta$  AND THE DIAMETER  $D$ .  $N_{up}$  IS AN UPPER BOUND ON  $N$  FOR THE SAME AVERAGE NUMBER OF HOPS.  $\bar{L}$  IS THE AVERAGE LOADING ON AN EDGE

$\Delta, D$	Shufflenet			
	$\bar{H}$	$N$	$N_{up}$	$N/N_{up}$
2, 3	2.0000	8	9	0.8889
2, 5	3.2609	24	30	0.8000
2, 7	4.6349	64	84	0.7619
2, 9	6.0692	160	256	0.6250
3, 3	2.1765	18	19	0.9474
3, 5	3.5625	81	122	0.6639
3, 7	5.0217	324	549	0.5902
3, 9	6.5074	1215	3289	0.3694
4, 3	2.2581	32	33	0.9697
4, 5	3.6911	192	343	0.5598
4, 7	5.1730	1024	2192	0.4672
4, 9	6.6683	5120	21864	0.2342
5, 3	2.3061	50	51	0.9804
5, 5	3.7620	375	784	0.4783
5, 7	5.2525	2500	6522	0.3833
5, 9	6.7505	15625	97688	0.1599
6, 3	2.3380	72	73	0.9863
6, 5	3.8068	648	1559	0.4157
6, 7	5.3012	5184	16014	0.3237
6, 9	6.8002	38880	335971	0.1157

topologies based on de Bruijn graphs support more stations than shufflenets for the same average number of hops.

Shufflenets and de Bruijn graphs are related. The de Bruijn graph  $G(\Delta, k)$  can be viewed as two columns, each column containing the same  $\Delta^k$  nodes. The two columns are connected in a perfect  $\Delta$ -shuffle.

Fig. 2 compares the delay performance of the 1024-node shufflenet and the 1024-node de Bruijn graph, both with  $\Delta = 4$ . For the shufflenet, we use the lower bound on the delay for any routing scheme from Theorem 2.

The maximum loading on a link in the shufflenet is at least 1323 (using the fact that  $L_{max} \geq \bar{L}$ , (3), and (17)) and the maximum loading on a link in the de Bruijn graph using longest-path routing is 1280. Therefore, the normalized throughput per station is at most 0.773 for the shufflenet and at least 0.779 for the de Bruijn graph. Thus the shufflenet has a lower throughput than the de Bruijn graph. With longest-

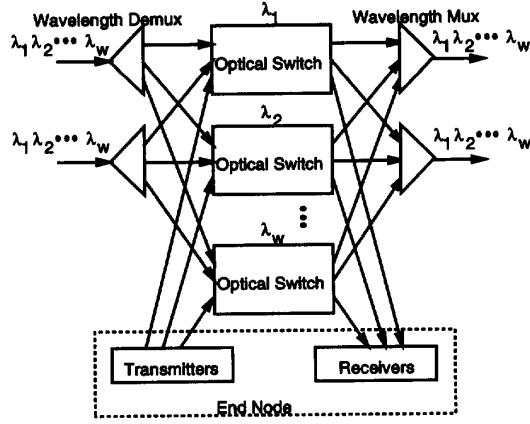


Fig. 4. Architecture of a  $3 \times 3$  optical routing node. The network consists of many of these nodes (stations) interconnected to form a particular topology, for example, a de Bruijn graph.

path routing, the de Bruijn graph has a lower delay than the shufflenet at all offered loads.

### III. PHYSICAL TOPOLOGIES

We now consider a network architecture consisting of all-optical wavelength-routing nodes interconnected by point-to-point links. A routing node of this type is described in [9] and is shown in Fig. 4. The node is capable of routing a wavelength at an input port to an output port independent of the remaining wavelengths, subject to the constraint that the same two wavelengths at two different input ports cannot be routed to a single output port. The routing pattern can be changed by reconfiguring the switches.

#### A. Network Topology

Our aim is to support the maximum number of end-stations given a certain end-to-end power loss requirement, or equivalently, to minimize the end-to-end power loss requirement given a certain number of stations to be supported. At the same time, we desire a topology where addressing and routing are simple.

Given the maximum out-degree of a node, say  $\Delta$  (we assume in-degree = out-degree), and the loss parameters of the switches and the gratings, the power loss through a node can be determined as follows. The maximum switch size is  $(\Delta + 1)$ . Assuming that each  $(\Delta + 1) \times (\Delta + 1)$  switch is built as a crossbar consisting of  $(\Delta + 1)^2 2 \times 2$  switches, the maximum loss incurred by the signal in going through a single  $(\Delta + 1) \times (\Delta + 1)$  switch is  $(2\Delta + 1)L + 2W$  dB where  $L$  is the loss (in dB) incurred in each  $2 \times 2$  switch and  $W$  is the fiber-to-switch coupling loss [10]. Since there can be up to  $D$  hops ( $D$  is the diameter) in a transmission path, if  $\alpha$  is the maximum link loss (in dB) per hop, the maximum loss is

$$(D + 1)((2\Delta + 1)L + 2W) + D\alpha \text{ dB}$$

which is proportional to the diameter  $D$ . In practice, there is a limit on the maximum allowable loss because the transmitter power is limited and at the receiver, a certain power is needed

TABLE III

NUMBER OF STATIONS THAT CAN BE SUPPORTED IN DE BRUIJN GRAPHS FOR DIFFERENT MAXIMUM OUT-DEGREES  $\Delta$  AND DIAMETERS  $D$ . THE NUMBER  $N_{\text{Moore}}$  IS AN UPPER BOUND ON THE MAXIMUM NUMBER OF STATIONS THAT BE SUPPORTED IN ANY GRAPH FOR THE SAME DEGREE AND DIAMETER. FOR ODD VALUES OF  $D$ , THE NUMBER OF STATIONS SUPPORTED BY THE CORRESPONDING SHUFFLENETS (FOR THE SAME  $\Delta$ ) IS ALSO SHOWN

$\Delta$	$D$	$N_{\text{deBr}}$	$N_{\text{Moore}}$	$N_{\text{deBr}}/N_{\text{Moore}}$	$N_{\text{shuf}}$	$N_{\text{shuf}}/N_{\text{Moore}}$
2	2	4	7	0.571429		
2	3	8	15	0.533333	8	0.533333
2	4	16	31	0.516129		
2	5	32	63	0.507937	24	0.380952
2	6	64	127	0.503937		
3	2	9	13	0.692308		
3	3	27	40	0.675000	18	0.450000
3	4	81	121	0.669421		
3	5	243	364	0.667582	81	0.222527
3	6	729	1093	0.666972		
4	2	16	21	0.761905		
4	3	64	85	0.752941	32	0.376471
4	4	256	341	0.750733		
4	5	1024	1365	0.750183	192	0.140659
4	6	4096	5461	0.750046		
5	2	25	31	0.806452		
5	3	125	156	0.801282	50	0.320513
5	4	625	781	0.800256		
5	5	3125	3906	0.800051	375	0.096006
6	2	36	43	0.837209		
6	3	216	259	0.833977	72	0.277992
6	4	1296	1555	0.833441		

in order to achieve a required bit-error-rate performance. This limits the number of stations that can be supported in the network.

For given values of  $\Delta$  and  $D$  (and hence a given value of the maximum loss), the best network topology is the one that maximizes the number of stations in the network. The network topology problem can be formulated as follows: Given the maximum in- and out-degree and the diameter, find the graph with the maximum number of nodes. The corresponding problem for undirected graphs is known to be hard, but the family of de Bruijn graphs are good though not necessarily optimal [6]. We consider directed de Bruijn graphs. In this section, when we refer to a de Bruijn graph, we mean the physical topology based on that de Bruijn graph.

Table III shows the number of nodes that can be supported in de Bruijn graphs of different degrees and diameters. Also shown is an upper bound on the maximum number of stations that can be supported in any graph (the Moore bound [6]) given by

$$N_{\text{Moore}} = 1 + \Delta + \Delta^2 + \dots + \Delta^D \\ = \frac{\Delta^{D+1} - 1}{\Delta - 1}.$$

This bound is obtained by observing that from any node, there can be  $\Delta$  nodes at distance 1,  $\Delta^2$  nodes at distance 2, and so on, and hence  $1 + \Delta + \Delta^2 + \dots + \Delta^D$  nodes at distance  $\leq D$ .

For moderate values of  $\Delta$ , de Bruijn graphs are close to this bound. On the same table, the number of stations that can be supported by different shufflenets is shown. Originally, shufflenets were proposed as logical topologies in [2]. However, recently, shufflenets have also been considered as physical

topologies although their loss properties were not analyzed [11], [12]. As can be seen, shufflenets are inferior to de Bruijn graphs as physical topologies.

### B. Wavelength Assignment

A network of  $N$  stations built using a star coupler that broadcasts all transmissions to all nodes requires  $N$  wavelengths to support one connection per station simultaneously. If a logical topology with each node having an in- and out-degree of  $\Delta$  is superimposed on this,  $N\Delta$  wavelengths are required. In the case of the wavelength routing topology, we expect that the number of wavelengths required will be less than  $N$  because wavelengths can be reused spatially in the network. The wavelength assignment problem can be stated as follows:

Given (a) the (physical) topology of the network, (b) the maximum number of duplex (i.e., if node A is transmitting to node B, node B is also transmitting to node A) connections,  $m$ , per station, and (c) the maximum blocking probability, find the number of wavelengths required, with the constraint that none of the connections that have been set up may change their wavelengths.

The determination of the number of wavelengths required to support a given set of connections is equivalent to solving a certain graph-coloring problem [9]. If we are interested in finding the number of wavelengths required to support  $m$  duplex connections per station (with no blocking), we have to find the maximum of the chromatic numbers of the graphs corresponding to all possible sets of  $m$  duplex connections. Moreover, even if this number of wavelengths were made available, it may not suffice to set up any combination of  $m$  duplex connections per station if we are not permitted to change the wavelengths of existing connections.

The analytical solution of the wavelength assignment problem, even for regular topologies like de Bruijn graphs, appears to be difficult. However, we can easily estimate the number of wavelengths required for networks of moderate sizes (a few thousand), for a small number of duplex connections per station ( $m \leq 5$ ) by Monte Carlo simulation methods.

In our simulation, for the case of one duplex connection per station, the procedure used for each trial is as follows: (1) Pick a pair of nodes A and B from the nodes that are not busy, at random. (2) Set up a connection from node A to node B using the first available wavelength on the shortest path between A and B. Set up the reverse connection from node B to node A also on the shortest path using the first available wavelength. (The wavelengths are preordered in an arbitrary fashion.) If there is no wavelength available to set up either of these connections, the duplex connection is considered blocked. Otherwise the connection is considered successful and the nodes are labelled as busy. Already existing connections are not disturbed.

In order to set up  $m > 1$  duplex connections per station, the procedure above is repeated  $m$  times. Each time, one duplex connection is set up for each station.

It may be possible to obtain a lower number of blocked connections by changing the method by which the wavelengths are

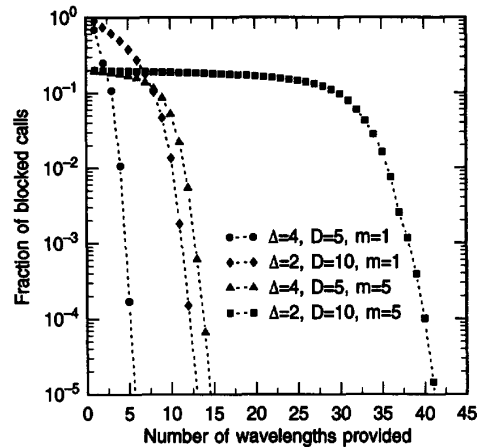


Fig. 5. Fraction of blocked connections versus number of wavelengths provided for networks with 1024 nodes and a distributed-switch physical topology based on two different de Bruijn graphs with 1024 nodes viz.  $G(2, 10)$  and  $G(4, 5)$ . The number of connections per station is denoted by  $m$ .

selected, and in the case of multiple connections per station, the order in which the connections are set up. For example, one might select the most- or least-used wavelength among the available wavelengths when setting up a connection, and in the case of multiple connections per station, one can set up all  $m$  duplex connections per station before proceeding to another station.

The results of our simulation for the physical topologies corresponding to the de Bruijn graphs  $G(2, 10)$  and  $G(4, 5)$ , both of which have  $N = 1024$  nodes, are shown in Fig. 5. It can be seen that six wavelengths for  $m = 1$ , and 15 wavelengths for  $m = 5$  suffice to limit the fraction of blocked connections to less than  $10^{-5}$  if the topology is  $G(4, 5)$ . On the other hand, with  $G(2, 10)$ , 14 wavelengths for  $m = 1$ , and 42 wavelengths for  $m = 5$  are required. Recall that the number of wavelengths is equal to the number of switches in each node and the size of each switch is  $(\Delta + 1) \times (\Delta + 1)$ . Increasing  $\Delta$  increases the size of the switches but reduces the number of wavelengths and hence the number of such switches that are needed. The choice of the particular de Bruijn graph that is used as the topology in this architecture will be determined by this trade-off between the size and number of switches as well as the loss properties of the switches.

### IV. CONCLUSION

We found that for the same maximum degree and average number of hops, logical topologies based on de Bruijn graphs can support a much larger number of stations than shufflenets, while retaining the simple addressing and self-routing properties of shufflenets. In the 1024-node example considered, the network based on the de Bruijn graph has a higher throughput and a lower average delay at all offered loads compared to the shufflenet. In general, except for small networks, compared to shufflenets, networks based on de



Brujn graphs have lower average delays at low offered loads and comparable throughputs.

We also found that de Bruijn graphs are good topologies for wavelength-routing networks consisting of all-optical routing nodes. These graphs can support a large number of stations for a given degree and diameter (and hence loss budget), and are regular with a simple routing rule. The number of wavelengths that are required to support a certain number of connections  $C$  is much smaller than  $C$ , unlike in a star network where  $C$  wavelengths are required, because wavelengths are reused spatially and switched independently of one another at the routing nodes.

## V. APPENDIX

In Theorem 3, we stated lower and upper bounds on the average number of hops,  $\overline{H}$ , for de Bruijn graphs with shortest-path routing. We now proceed to compute it exactly. A small change in notation is required. Denote  $n(\cdot)$  for the de Bruijn graph  $G(\Delta, D)$  by  $n_D(\cdot)$ . Let  $s_D(i)$  denote the number of nodes for which the shortest circuit containing that node has length  $i$ . We then have the following theorem.

*Theorem 4:*

$$n_D(i) + s_D(i) = n_{D-1}(i-1)\Delta^2, \quad 1 \leq i \leq D, D \geq 2.$$

*Proof:* If  $(a_1, \dots, a_{D-1})$  is the source node and  $(b_1, \dots, b_{D-1})$  is the destination node of a source-destination pair with minimum distance  $i-1$  in  $G(\Delta, D-1)$ , then we can find  $\Delta^2$  source-destination pairs in  $G(\Delta, D)$  with distance  $i$  namely, all source-destination pairs with source nodes of the form  $(a, a_1, \dots, a_{D-1})$  and destination nodes of the form  $(b_1, \dots, b_{D-1}, b)$ , where we have  $\Delta$  choices for each of  $a$  and  $b$ .  $i$  must be the minimum distance for each of these source-destination pairs; otherwise we can find a path of distance less than  $i-1$  between  $(a_1, \dots, a_{D-1})$  and  $(b_1, \dots, b_{D-1})$ . By definition,  $s_D(i)$  of these  $\Delta^2$  pairs in  $G(\Delta, D)$  have the same source and destination nodes. Therefore,

$$n_D(i) + s_D(i) \geq n_{D-1}(i-1)\Delta^2, \quad 1 \leq i \leq D, D \geq 2.$$

The proof of the reverse inequality is similar.  $\square$

We define  $n_D(0) = \Delta^D$  for  $D \geq 0$  so that the above recurrence holds for  $D \geq 1$ . In order to compute the  $n_D(\cdot)$  for  $D \geq 1$  from the recurrence relation above, it only remains to find  $s_D(\cdot)$ . We have not been able to find any "simple" expression for the  $s_D(\cdot)$  but we have the following partial results.

*Theorem 5:*

$$s_D(k) = \sum_{d|k} \mu(d) \Delta^{k/d}, \quad D \geq 2k-2, k \geq 1$$

where the summation is over all the divisors  $d$  of  $k$  and  $\mu(n)$  is the Möbius function which is defined as follows ([13, p. 10]): Let

$$n = p_1^{e_1} p_2^{e_2} \dots p_r^{e_r}$$

be the unique factorization of  $n > 1$  as a product of prime powers. Then,

$$\begin{aligned} \mu(1) &= 1, \\ \mu(n) &= 0, \quad \text{if any of the } e_i > 1, \\ \mu(n) &= (-1)^r, \quad \text{if } e_1 = e_2 = \dots = e_r = 1. \end{aligned}$$

In order to prove this theorem, we need the following definitions and lemmas.

*Definition:* A string  $(a_1, a_2, \dots, a_D)$  of length  $D$  has a period of length  $k$  if

$$a_i = a_{i+k}, \quad 1 \leq i \leq D-k$$

and the sequence  $(a_1, a_2, \dots, a_k)$  of length  $k$  is called its period.

Note that the nodes in  $G(\Delta, D)$  can be identified with strings (or sequences) of length  $D$ . With this identification observe that if a node is contained in a circuit of length  $k$ , the sequence (or string) corresponding to that node has a period of length  $k$ .

*Lemma 1 ([14]):* If  $D \geq m+n-\gcd(m, n)$  and a sequence of length  $D$  has a period of length  $m$  and a period of length  $n$ , it has a period of length  $\gcd(m, n)$ .

*Lemma 2:* If  $D \geq 2k-2$  and a sequence of length  $D$  has a period of length  $k$ , the length of the shortest period of that sequence is a divisor of  $k$ .

*Proof:* Consider any sequence of length  $D \geq 2k-2$  with a period of length  $k$ . If the length of the shortest period of this sequence is  $k$ , there is nothing to prove. Otherwise let  $p \leq k-1$  be the length of the shortest period of this sequence. Suppose  $p$  is not a divisor of  $k$ . Since  $\gcd(k, p) \geq 1$ , we have  $k+p-\gcd(k, p) \leq k+(k-1)-1 \leq D$ . Therefore, by Lemma 1, the sequence has a period of length  $\gcd(k, p) < p$ , which is a contradiction.  $\square$

*Proof of Theorem 5:* Consider the  $\Delta^k$  sequences of length  $D$  which have a period of length  $k$ . (Note that the number of sequences of length  $D$  which have a period of length  $k$  is  $\Delta^k$ , since each sequence of length  $k$  occurs exactly once as a period of a sequence of length  $D \geq k$ .) Among these sequences with a period of length  $k$ , some of them have a shortest period of length  $k$  while the others have shortest periods whose lengths are divisors of  $k$  (from Lemma 2). Moreover, all sequences which have a period whose length is a divisor of  $k$  also have a period of length  $k$ . Hence,

$$\sum_{d|k} s_D(d) = \Delta^k, \quad D \geq 2k-2, k \geq 1.$$

The theorem follows by applying Möbius Inversion [13, theorem 2.1.1., p. 11] to the equation above.  $\square$

Using Theorem 5, we get,

$$\begin{aligned} s_D(1) &= \Delta, & D \geq 1, \\ s_D(2) &= \Delta^2 - \Delta, & D \geq 2, \\ s_D(3) &= \Delta^3 - \Delta, & D \geq 4, \\ s_D(4) &= \Delta^4 - \Delta^2, & D \geq 6, \end{aligned}$$

and so on. All the values of  $s_D(\cdot)$  required to calculate  $n_D(i)$  for  $D \leq 5$  are provided by Theorem 5 together with

$$\sum_{k=1}^D s_D(k) = \Delta^D,$$

with the exception of  $s_5(4)$ , which is given by

$$s_5(4) = \Delta^4 - 2\Delta^2 + \Delta.$$

This is computed as follows. The general sequence of length 5 with a period of length 4 is of the form  $(a, b, c, d, a)$  and there are  $\Delta^4$  such sequences. Among these sequences, the length of the shortest period of the  $\Delta$  sequences which satisfy  $a = b = c = d$  is 1, the  $\Delta(\Delta - 1)$  sequences which satisfy  $a = c \neq b = d$  is 2 and the  $\Delta(\Delta - 1)$  sequences which satisfy  $a = b = d \neq c$  is 3. Therefore,

$$s_5(4) = \Delta^4 - \Delta - \Delta(\Delta - 1) - \Delta(\Delta - 1) = \Delta^4 - 2\Delta^2 + \Delta.$$

The other values of  $s_D(\cdot)$  may be calculated similarly but the calculations become progressively more complex as  $D$  increases.

*Example:* Let us compute  $\bar{H}$  for  $G(\Delta, 2)$ . We have

$$\begin{aligned} n_2(1) &= n_1(0)\Delta^2 - s_2(1) \\ &= \Delta\Delta^2 - \Delta \\ &= \Delta^3 - \Delta. \end{aligned}$$

$$\begin{aligned} n_2(2) &= n_1(1)\Delta^2 - s_2(2) \\ &= (\Delta^2 - \Delta)\Delta^2 - (\Delta^2 - \Delta) \\ &= \Delta^4 - \Delta^3 - \Delta^2 + \Delta. \end{aligned}$$

Then,

$$\begin{aligned} \bar{H} &= \frac{2n_2(2) + 1n_1(1)}{\Delta^2(\Delta^2 - 1)} \\ &= 2 - \frac{1}{\Delta}. \end{aligned}$$

Similarly, we can calculate that for  $G(\Delta, 3)$ ,

$$\bar{H} = \frac{3\Delta^4 + 2\Delta^3 - 2\Delta - 1}{\Delta^2(\Delta^2 + \Delta + 1)},$$

for  $G(\Delta, 4)$ ,

$$\bar{H} = \frac{4\Delta^5 + 3\Delta^4 + \Delta^3 - \Delta^2 - 4\Delta - 2}{\Delta^2(\Delta^3 + \Delta^2 + \Delta + 1)},$$

and for  $G(\Delta, 5)$ ,

$$\bar{H} = \frac{5\Delta^7 + 4\Delta^6 + 2\Delta^5 - 3\Delta^3 - 6\Delta^2 - 3\Delta - 1}{\Delta^3(\Delta^4 + \Delta^3 + \Delta^2 + \Delta + 1)}.$$

#### ACKNOWLEDGMENT

The authors wish to thank Pierre Humblet for help with the computation of the bounds of Theorem 3, Paul Green for his comments on the manuscript, and Anujan Varma and Harold Stone for helpful discussions.

#### REFERENCES

- [1] I. Chlamtac, A. Ganz, and G. Karmi, "Transport optimization in broadband networks," in *Proc. IEEE INFOCOM '91*, 1991, pp. 49–58.
- [2] A. S. Acampora, "A multi-channel multihop local lightwave network," in *Proc. IEEE GLOBECOM '87*, 1987, pp. 37.5.1–9.
- [3] M. G. Hluchyj and M. J. Karol, "Shufflenet: An application of generalized perfect shuffles to multihop lightwave networks," *IEEE/OSA J. Lightwave Tech.*, vol. 9, pp. 1386–1397, 1991.
- [4] A. S. Acampora and M. J. Karol, "An overview of lightwave packet networks," *IEEE Network Mag.*, pp. 29–41, Jan. 1989.
- [5] A. Tanenbaum, *Computer Networks*. Englewood Cliffs, NJ: Prentice Hall, 1988.
- [6] B. Bollobás, *Extremal Graph Theory with Emphasis on Probabilistic Methods*. American Math. Soc., 1986.
- [7] S. W. Golomb, *Shift Register Sequences*. Aegean Park Press, 1982.
- [8] M. A. Sridhar and C. S. Raghavendra, "Fault-tolerant networks based on the de Bruijn graph," *IEEE Trans. Computers*, vol. 40, pp. 1167–1174, Oct. 1991.
- [9] I. Chlamtac, A. Ganz, and G. Karmi, "Purely optical networks for terabit communication," in *Proc. IEEE INFOCOM '89*, 1989, pp. 887–896.
- [10] R. A. Spanke, "Architectures for guided-wave optical space switching systems," *IEEE Commun. Mag.*, vol. 25, pp. 42–48, May 1987.
- [11] Z. Zhang and A. S. Acampora, "Analysis of multihop lightwave networks," in *Proc. IEEE Globecom '90*, 1990, pp. 1873–1879.
- [12] A. S. Acampora and S. Ali-Shah, "Multihop lightwave networks: A comparison of store-and-forward and hot-potato routing," *IEEE Trans. Commun.*, vol. 40, pp. 1082–1090, 1992.
- [13] M. Hall, Jr., *Combinatorial Theory*. Reading, MA: Wiley, 1986.
- [14] D. E. Knuth, J. J. H. Morris, and V. R. Pratt, "Fast pattern matching in strings," *SIAM J. Comput.*, vol. 6, no. 2, pp. 323–350, 1977.



**Kumar N. Sivarajan** (S'88–M'90) received the B. Tech. degree in electrical engineering (electronics) from the Indian Institute of Technology, Madras, in 1987, and the M. S. and Ph. D. degrees in electrical engineering from the California Institute of Technology, Pasadena, in 1988 and 1990, respectively.

From July 1990 to March 1992, he held a post-doctoral position with the Advanced Optical Networking department at the IBM T. J. Watson Research Center in New York. Since then he has been a Scientific Staff Member, first with the High Performance Computing and Communications department and since October 1993, with the Advanced Networking Laboratory, in the T. J. Watson Research Center. His research interests include mobile, optical and high-speed networks, and discrete mathematics.

Dr. Sivarajan was the recipient of the IEEE Charles LeGeyt Fortescue fellowship for the academic year 1987–1988.

**Rajiv Ramaswami** (S'88–M'90) received the B. Tech. degree in electrical engineering (electronics) from the Indian Institute of Technology, Madras, in 1986, and the M. S. and Ph. D. degrees in electrical engineering and computer science from the University of California, Berkeley in 1988 and 1990, respectively.

Since Jan. 1989, he has been with the optical networking group at the IBM T. J. Watson Research Center, Yorktown Heights, and is presently the project leader of a group prototyping optical networks and implementing high-speed transport protocols. His research interests are in the areas of optical networks, high-speed protocols, and network control algorithms.

Dr. Ramaswami is currently an Editor for the IEEE/ACM TRANSACTIONS ON NETWORKING.