WDM-Based Local Lightwave Networks

Part II: Multihop Systems

As an alternative to single-hop local lightwave networks, multihop systems have their own strengths and limitations.

---

Biswa Mukherjee

A lightwave network can be constructed by exploiting the capabilities of emerging optical technology, viz., dense wavelength division multiplexing (WDM) and tunable optical transceivers. (For a review of the basic concepts of this technology, see [11, 31, 40].) The vast optical bandwidth of a fiber (approximately 25 to 30 THz corresponding to the low-loss region in a single-mode optical fiber [40, Fig. 1]) is carved up into smaller-capacity channels, each of which can operate at peak electronic processing speeds (viz., over a small wavelength range) of, say, a few Gb/s. That is, a single fiber can accommodate up to 10^14 electronic-grade channels. Thus, innovative parallelism and concurrency mechanisms must be employed by end-users operating with electronic front-ends in order to tap into the huge optical bandwidth of a single fiber.

Typically, a node is equipped with a small number of optical transmitters (lasers) and optical receivers (filters), and some of these devices can be made tunable to operate on different channels as well. By tuning its transmitter(s) to one or more wavelength channels, a node can transmit into those channel(s); similarly, a node can tune its receiver(s) to receive from the appropriate channels. The system can be configured as a broadcast-and-select network in which all of the inputs from various nodes are combined in a WDM star coupler and the mixed optical information is broadcast to all outputs [11, 31] (Fig. 1a). In general, the physical topology, instead of being a star, can be a linear bus or a tree [40, Fig. 3]. Thus, given any physical network topology, the fact that the input lasers (transmitters), or the output filters (receivers), or both, can be made tunable opens up a multitude of possible virtual network configurations.

Generally, there are two classes of architectures that can be constructed for WDM local and metropolitan area networks—single-hop and multihop. If the system is based on a single-hop architecture, the nodes must communicate with one another in one hop. This requires a significant amount of dynamic coordination between the nodes since, for the packet transmission duration at least, one of the transmitters of the sending node and one of the receivers of the destination node must be tuned to the same wavelength channel. In order for the system to be efficient, the transceivers must be very agile (i.e., they must be rapidly tunable). Given that the tuning ranges of current optical transceivers are limited, and that their current tuning times are significantly long (compared to the packet transmission duration), the challenge is to design protocols for performing the proper transmission coordination in an efficient fashion.

Single-hop systems are reviewed in a companion article [40].

Conversely, in multihop systems, the channel to which a node's transmitter or receiver is to be tuned is relatively static, and this assignment normally is not expected to change except when a new global assignment of all transceivers is deemed to be more beneficial. It is unlikely that there will be a direct path between every node pair so that, in general, a packet from a source to a destination may have to hop through zero or more intermediate nodes. Different virtual structures will have different operational features (e.g., ease of routing) and different performance characteristics (e.g., minimal average packet delay, minimal number of hops, balancing of link flows, etc.).

An example multihop architecture is shown in Fig. 1. The physical topology is a star (Fig. 1a) while the embedded virtual topology is a 2 x 2 torus (Fig. 1b). Note that, in this example, node 1 can communicate with nodes 2 and 3 directly via wavelength channels 0 and 0, but in order to reach node 4, information from node 1 should multihop either through node 2 or node 3.

Between single-hop and multihop systems, there appears to be no clear winner, so the capabilities and characteristics of both categories of systems must be thoroughly examined. Specifically, this article reviews the characteristics of multihop lightwave network architectures; single-hop systems were surveyed in [40]. In developing these architectures, one must remember the fact that these approaches must not only be simple and implementable, i.e., they should be based on realistic assumptions on the prop-
properties of WDM optical components, but also that they be scalable to accommodate a large and expandable user population.

Although the transceiver tuning times play a vital role in determining the performance and characteristics of single-hop systems, they have little impact on multihop systems since the multihop virtual topology is essentially a static one. However, in designing a "good" multihop system, there are two other important issues which the system architect must address. First, the virtual structure chosen must be close to optimal in some sense, e.g., the structure's average (hop) distance between nodes must be small, or the average packet delay must be minimal, or the maximum flow on any link in the virtual structure must be minimal. Second, the nodal processing complexity also must be small because the high-speed environment allows very little processing time; consequently, simple routing mechanisms must be employed.

A routing-related subproblem is the buffering strategies at the intermediate nodes. A number of papers advocate the use of deflection routing under which, a packet, instead of being (electronically) buffered at an intermediate node, may be intentionally misrouted, but still reach its destination over a slightly longer (but all-optical) path.

Multihop structures can be either irregular or regular. Irregular multihop structures generally address the optimality criterion directly, but the routing complexity can be large since they lack a structural connectivity pattern. Studies dealing with topological optimization of multihop architectures have been reported [6, 7, 36]. Regular structures, because of their structured node-connectivity pattern, have simplified routing schemes; however, their regularity also constrains the set of solutions to solving the optimality problem, and the number of nodes in a complete regular structure usually forms a discrete set of integers, rather than an arbitrary integer. Regular structures which have received a significant amount of attention in the literature are the perfect shuffle (called ShuffleNet) [1, 2, 30], the de Bruijn graph [44], the toroid (Manhattan Street Network, MSN) [5, 38, 39], the hypercube [18, 20, 37], and the linear dual bus [8, 47]. A virtual tree structure also has been investigated [41]. Characteristics of alternative routing strategies, including deflection routing, in ShuffleNet also have been studied quite extensively [4, 32, 33, 52, 53].

Attention must be paid to another piece of input to these designs, viz. the fact that the offered loads by the various nodes may not necessarily be symmetric, which is more pronounced with the proliferation of special-purpose networking equipment such as servers and gateways. Regular structures generally are amenable to uniform loading patterns, while irregular structures generally can be optimized for arbitrary workloads. The performance effect of nonuniform traffic and corresponding adaptive routing schemes to control congestion have started to receive attention [22, 32].

Finally, another pertinent issue which a multihop network architect must be mindful of is whether to employ dedicated or shared channels. Under the case of dedicated channels, each virtual link

Regular structures generally are amenable to uniform loading patterns, while irregular structures generally can be optimized for arbitrary workloads.

Figure 1. An example 2 x 2 (4 node) multihop network: (a) physical topology, (b) logical topology employs a dedicated wavelength channel. However, since internode traffic may be bursty, the traffic on an arbitrary link is expected to be bursty as well, as a result of which some of the links' utilizations may be low. Consequently, the shared channel mechanism advocates the use of two or more virtual links to share the same channel in order to improve the channel utilization. This, however, also introduces the need for a multiple access protocol on the channel, viz. an arbitration mechanism that governs access rights to the channel. Issues related to shared-channel strategies have been treated in [1, 6, 30].

Next, we will outline studies dealing with the topological optimization problem (resulting in optimal irregular structures). In a subsequent section, we review a number of alternative regular structures which have been investigated for multihop systems, and study their properties. Later, we focus on a subset of regular structures, viz. the linear dual bus, and the goal will be to study how nodes can be placed (near) optimally on such structures. Finally, we elaborate on other multihop approaches, viz. architectures employing the shared-channel approach, including one based on subcarrier multiplexing, which is another technique that can be used to share a wavelength channel in a lightwave network [16, 40, 42].

Topological Optimization Studies

Let us first review the construction of optimal structures based on minimizing the maximum link flow [34-36], followed by optimizations based on minimization of the mean networkwide packet delay [6, 7].
Flow-Based Optimization

Consider a network containing an arbitrary number of nodes \( N \), which are indexed \( 1, 2, \ldots, N \). Each node has \( T \) transmitters and \( T \) receivers. The capacity of each WDM channel is \( C \) units (say bits). The traffic matrix is given by \( \lambda_{sd} \) where \( \lambda_{sd} \) is the traffic flow from source node \( s \) to destination node \( d \) for \( s = 1, 2, \ldots, N \). The flow in link \( ij \) is denoted by \( f_{ij} \), while the fraction of the \( \lambda_{sd} \) traffic flowing through link \( ij \) is denoted by \( f_{ij}^{sd} \). Let \( Z_{ij} \) be the number of directed channels from node \( i \) to node \( j \). Then, the capacity of link \( ij \) equals \( C_{ij} = Z_{ij} \cdot C \). The fraction of the \( (i,j) \)-link capacity which is utilized equals \( f_{ij} / C_{ij} \). An arbitrary topology will have a link with maximum utilization given by

\[
\max_{(ij)} \left( \frac{f_{ij}}{C_{ij}} \right)
\]

Among various alternative topologies that are possible, the one that minimizes the above quantity is chosen to be the optimal interconnection pattern [36].

Formally, the aforementioned flow and wavelength assignment (FWA) problem can be set up as a mixed integer optimization problem with a min-max objective function subject to a set of linear constraints [36]. The main characteristic of this problem formulation is that it allows the traffic matrix to scale up by the maximum amount before its most heavily loaded link saturates. Another important characteristic is that only the node-to-node traffic intensities need to be known, and the solution is independent of the traffic type, which could be either circuit-or packet-switched (which, in turn, could be datagram-based or virtual-circuit-based).

Unfortunately, the search space for the connectivity diagram grows rapidly with increasing \( N \). Hence, the work in [36] proposes a suboptimal and iterative algorithm which first determines a heuristic initial solution and then applies branch-and-exchange operations iteratively to improve the solution. The initial solution, in turn, consists of a connectivity problem, which heuristically tries to maximize the one-hop path traffic (i.e., it attempts to connect nodes with more traffic between them in one hop), and this can be solved by a special version of the simplex algorithm. The second part of the initial solution is the routing problem, which can be formulated as a multicommodity flow problem with a nonlinear, nondifferentiable, convex objective function [36], and it can be solved by using the flow deviation method [23]. Iterative improvement is performed by considering a number of least-utilized branches (say \( K \)) at a time. A branch-exchange operation is performed by (1) swapping the transmitters (or receivers) of the two least-utilized branches, (2) resolving the routing problem on the new connectivity diagram, and (3) accepting the swap if the new topology leads to a lower networkwide maximum link utilization. This procedure is repeated until no improvement is obtained.

Results obtained in [36] via the above algorithm for \( N = 8 \) and \( T = 2 \) are encouraging. The connectivity diagrams and the corresponding link flows for uniform traffic, ring-type traffic, disconnected-type traffic, and centralized traffic do match with intuition. An improvement of the problem formulation also can accommodate the finite tuning range of the transceivers [35].

Delay-Based Optimization

In designing an optimal virtual topology, an alternative objective may be to minimize the mean networkwide packet delay. The packet delay has two components. The first is due to the propagation delays encountered by the packet as it hops from the source through intermediate nodes to the final destination. The second is due to queuing at the intermediate nodes. In a high-speed environment where the channel capacity \( C \) is large and the link utilizations are expected to be in the light to moderate range, the queuing delay component can be ignorable compared to the propagation delay component which is directly dependent on the link distance between the nodes [6]. Thus, this optimization also requires knowledge of the distance matrix \( (d_{ij}) \) where \( d_{ij} \) is the distance between node \( i \) and node \( j \) per the underlying physical topology. Therefore, the mean networkwide packet delay can be written as

\[
\bar{D} = \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{f_{ij} d_{ij}}{v} + \Delta
\]

where \( v \) = velocity of light in a fiber, \( f_{ij} \) is the flow through link \( ij \), \( \gamma = \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_{sd} \) = total offered load to the network, and \( \Delta \) is the nodal queuing delay component (whose derivation can be found in [6]).

Formally, the optimization can be stated as follows [6]:

Given:
- Traffic matrix
- Distance matrix

Objective:
- Minimize mean packet delay

Design variables:
- Virtual topology
- Link flows

Constraints:
- Flow conservation
- Nodal connectivity (including the number of transmitters and receivers per node)

The work in [6] reports that algorithms based on simulated annealing can be employed to solve both the dedicated-channel and the shared-channel cases, where time-division multiple access (TDMA) is employed for channel sharing. A faster solution to the shared-channel topology has also been obtained in [6] by using genetic algorithms.

Regular Structures

Regular topologies that have been studied as candidates for multihop lightwave networks include the perfect shuffle [1, 2, 30], the de Bruijn graph [44], the toroid [5, 38, 39], and the hypercube [18, 20, 37]. We study their characteristics in the following subsections. (For general results and bound on regular multihop structures, see [30].)
ShuffleNet
A \((p, k)\) ShuffleNet can be constructed out of \(N = kp^k\) nodes which are arranged in \(k\) columns of \(p^k\) nodes each (where \(p, k = 1, 2, 3, \ldots\)), and the \(k\)th column is wrapped around to the first in a cylindrical fashion [2]. The nodal connectivity between adjacent columns is a \(p\)-shuffle, which is analogous to the shuffling of \(p\) decks of cards. This interconnection pattern can be defined more precisely as follows: (a) number the nodes in a column from top to bottom as 0 through \(p^k - 1\), and (b) direct \(p\) arcs from node \(i\) to nodes \(i + j\) \(1, \ldots, j + p - 1\) in the next column where \(j = (i \mod p^k) \times p\). A \((2, 2)\) ShuffleNet is shown in Fig. 2.

An important performance metric of this structure is the mean hop distance between any two randomly chosen nodes. From any tagged node in any column (say, the first column), note that \(p\) nodes can be reached in one hop, another \(p^2\) nodes in two hops, and so on, until all remaining \(p^k - 1\) nodes in the first column are visited. Call this the first pass. In the second pass, all nodes which were not visited in the first pass can now be visited (although there can be multiple routes for doing so). For example, in the \((2, 2)\) ShuffleNet in Fig. 2, node \(3\) can be reached from node \(0\) (either via the path \(0 \rightarrow 3\) or the path \(0 \rightarrow 4 \rightarrow 6\)). A preferred routing algorithm (developed in [2]) will be outlined later. Thus, the number of nodes which are \(h\) hops away from a tagged node can be written as

\[
n_h = p^h - \sum_{h=1}^{k-1} x_{h,k} \cdot p^h = \begin{cases} p^h & \text{for } h = 1, 2, \ldots, k-1 \hfill \\
p^h - p^{h+1} & \text{for } h = k, k+1, \ldots, 2k-1 \hfill 
\end{cases}
\]

Then, the average number of hops between any two randomly selected nodes, given by

\[
\bar{h} = \frac{\sum_{h=1}^{2k-1} x_{h} n_h / (N-1)}{N-1}
\]

can be obtained as

\[
\bar{h} = \frac{k p^k (p^k - 1) (3k - 1) - 2k (p^k - 1)}{2(p^k - 1)} = \frac{k p^{k+1}}{n_h}
\]

Note that the ShuffleNet structure's diameter, which for any multihop structure is defined to be the maximum hop distance between any two nodes, equals \(2k - 1\).

Due to multihopping, note that only a fraction of a link's capacity is actually being utilized for carrying direct traffic between the two specific nodes connected by a link, while the remaining link capacity is used for forwarding of multihop traffic. In a symmetric \((p, k)\) ShuffleNet in which the routing algorithm uniformly loads all the links, the above utilization of any link is given by \(1/n_k\). Since the network has \(N = kp^k\) links, the total network capacity equals [2]

\[
C = \frac{k p^{k+1}}{n_h}
\]

while the per-user throughput equals \(C/N = p/n_h\). The above throughput for different \((p, k)\) combinations can be found in [2]. Note that the per-user throughput may be increased by choosing a smaller \(k\) and a larger \(p\), so that the mean hop distance between nodes is reduced.

The case of \(p = 2\) is treated in [1]. Treatment of the shared-channel ShuffleNet, i.e., the assignments of channels between nodes for sharing purposes and the corresponding throughput results can be found in [30], and is discussed later. In the following paragraphs, we concentrate on alternative routing strategies that can be employed in this network.

Simple Routing in ShuffleNet - A number of papers have appeared in the literature dealing with the routing problem in ShuffleNet [2, 4, 32, 33, 52, 53]. A simple addressing and fixed routing scheme is outlined in [2]. A node in a \((p, k)\) ShuffleNet is assigned the address \((c, r)\) where \(c \in \{0, 1, \ldots, k - 1\}\) is the node's column coordinate (labeled 0 from left to right) and \(r \in \{0, 1, 2, \ldots, p^k - 1\}\) is the node's row coordinate (labeled 0 from top to bottom, using base-\(p\) digits).

Thus, one may write \(r = r_0 r_1 r_2 \ldots r_d r_{d-1}\). This addressing scheme, along with the \(p\)-shuffle interconnection pattern, has the property that, from any node \((c, r)\) where \(r = r_0 r_1 r_2 \ldots r_d r_{d-1}\), the row addresses of all the nodes reachable in the next column have the same first \(k - 1\) \(p\)-ary digits (given by \(r_0 r_1 \ldots r_d r_{d-1}\)), and they differ in only the least significant digit. For routing purposes, it is required that the destination address \((c', r')\) be included in every packet. When such a packet arrives at an arbitrary node \((\tilde{c}, \tilde{r})\) then, it is removed from the network if \((c, r) = (\tilde{c}, \tilde{r})\) (i.e., the packet has reached its destination). Otherwise, node \((c, r)\) determines the column distance \(X\) between itself and the packet's destination \((c', r')\) to be

\[
X = \begin{cases}
(k + c' \mod c) & \text{if } c' \neq c \\
0 & \text{if } c' = c
\end{cases}
\]

Out of the \(p\) nodes in the next column to which node \((c, r)\) may forward the current packet, it chooses the one whose least-significant digit is given by \(c' \mod X\) (which is part of the destination node's address obtainable from the packet header). In particular, the packet is routed to the node with the identity \((c + c' \mod X, c' \mod X)\).

Note that this routing scheme follows the single shortest path between nodes \((c, r)\) and \((c', r')\) if the number of hops between them equals \(k\) or less; otherwise, it chooses one among several possible shortest paths. Also, note that the routing decision made at node \((c, r)\) is independent of the packet's original source.

Effect of Nonuniform Traffic on ShuffleNet Performance - If the traffic between all node pairs is the same, then all of the ShuffleNet links are uni-
Deflections occur due to contention, and contentions may be resolved based on priority mechanisms.

![Figure 3. A (2,3) de Bruijn graph.](image)

![Figure 4. Nonuniform link flows in a 1024-node (4,5)-de Bruijn graph.](image)

formally loaded. In reality, however, the offered loading is expected to fluctuate and be nonuniform; so, the effect of nonuniform traffic on the network's traffic handling capability is of paramount importance [22].

The work in [22] employs two alternative approaches to study the effect of traffic imbalance. The first approach, referred to as extreme-value analysis, is based on the assumption that, even though the load distributions are uniform, the intensity of traffic sourced by individual nodes follows a normal distribution. Then, given the average and the variance of the aggregate per-channel load intensities, the probability that a channel is overloaded can be determined. The second approach, referred to as random load generation method, assumes that even though the intensities of loads generated by all nodes are the same, the pattern of destination reference is random. In particular, all of the traffic generated at a node is assumed to be destined to exactly one of the other network nodes, determined randomly. It is found that, for realistic load patterns, the uniform load model predicts that the nodal throughput is reduced by a de-loading factor of approximately 0.5 for small user populations to approximately 0.3 for large user populations.

Adaptive and Deflection Routing Strategies in ShuffleNet - An adaptive routing scheme for ShuffleNet (in order to deal with nonuniform traffic) has also been recently reported [32]. The objective of this scheme is to ensure that packets avoid congestion or hot spots in the network. Basically, when a packet is more than k hops away from its destination in a (p,k) ShuffleNet, the packet is routed on the outgoing link with the least number of queued packets. If more than one such link exists, one is chosen at random. In addition, sometimes, even if a packet is less than k hops away from its destination (i.e., a single shortest path to the destination exists), the packet may be routed to one of the remaining and least-congested p − 1 outgoing links if the number of packets queued for the preferred link exceeds a certain threshold, while the queue size on the least-congested link is below a different and much smaller threshold. Thus, although the packet is now bumped and must take a longer path to its destination, it may still reach its destination faster since it can avoid the congested link(s) in the network.

In a multihop network, when two packets arrive at an intermediate node, and contend for the same preferred outgoing link, one of them usually is allowed access to the link (possibly based on some priority mechanism). The other packet may be electronically buffered at the node (i.e., the normal store and forward mechanism may be employed), but this will require opto-electronic and electro-optic conversions, which can slow down the data rates on the optical channels if they are to be operated at higher than electronic speeds. To avoid this conversion (and buffering), the intermediate node may choose an alternate strategy, viz., it can deflect or intentionally misroute the packet(s) which has just lost its contention(s) along its other (free) outgoing paths with the hope that the packet will eventually find its way back to its destination (over a slightly longer all-optical path while avoiding congested parts of the network). Deflection routing also has been studied for the Manhattan Street Network (MSN), a torus network [38, 39].

The advantage of employing deflection routing in a (p,k) ShuffleNet is that, if a packet at an intermediate node is more than p hops away from its destination, then multiple shortest paths exist between the intermediate node and the destination. The intermediate node can therefore be considered a "don’t-care" node, and all outgoing links are equally suitable [4]. A large network (with a large p) has a large number of "don’t-care" nodes for an arbitrary path, and hence there are fewer contentions for preferred paths.

The works in [4, 33, 53] assume p = 2, but some of their models appear to be generalizable. In [4], deflections at intermediate nodes are treated on a probabilistic basis, and it is found that deflection routing can reduce the mean number of hops, but it can also result in lower aggregate capacity, as compared to store-and-forward routing.

Deflections occur due to contention, and contentions may be resolved based on priority mechanisms. The works in [33, 53] consider two metrics among contending packets—one of them is the (remaining) distance to the destination, and the other is the age (the number of deflections already suffered by the contending packet). Age-distance priority is treated in [33], where packets which have suffered more deflections are given higher priority, and if there is a tie, the packet which
An undesirable characteristic of the de Bruijn graph is that, even if the offered traffic to the network is fully symmetric, the link loadings can be unbalanced.
the number of alternate paths available to reach a node, thereby providing increased fault tolerance. The method requires that (1) the links of the network graph be labeled by dummy variables, (2) the transfer function of the resulting signal flow graph be solved, and (3) the transfer function coefficients be expanded into a Taylor series. These coefficients then correspond to the number of paths of a given length between a source node and a destination node.

For the example 64-node (2,4)-ShuffleNet in Fig. 7, consider the transfer function from node 0 to node 16 or node 17. Both of these transfer functions are given by [5]

\[ T(D) = D + D^3 + 15D^5 + 225D^7 + 3375D^9 + \ldots \]

where \( D \) is a dummy variable representing one hop. The above transfer function indicates that, in going from node 0 to node 16, there is one 1-hop path, one 5-hop path, 15 9-hop paths, 225 13-hop paths, and so on.

Using such path enumeration methods, the 64-node MSN and 64-node ShuffleNet have been compared in [5]. It is found that, for the 64-node case, (1) in ShuffleNet, there are more paths to reach a node from any other node in a given number of hops, (2) ShuffleNet has a larger number of distinct nodes reachable from a given node in a given number of hops, (3) ShuffleNet has more paths to reach the distinct nodes in a given number of hops, and (4) ShuffleNet has a smaller hop distance.

**All-Optical Switch for MSN Node** - The investigation of an all-optical switch architecture for MSN [12] is motivated by the fact that, instead of simply deflecting a packet when it contends with another packet for the same preferred outgoing path, the optical switching performance can be improved by providing some form of intermediate node buffering without the electronic penalty of opto-electronic conversion, electronic buffering, and electro-optic conversion. Specifically, the use of optical delay lines as optical buffers is proposed in [12], and a number of alternative architectures incorporating such delay lines is investigated. It is shown that the proposed optical switch architectures can provide switching efficiencies comparable to those of electronic switches, but at optical transmission speeds. Finally, even though these architectures are studied in the MSN context, the concepts are generalizable to other multihop networks as well.

**Token Grid**

The token grid can be considered to be a mechanism employing the MSN connectivity pattern, except that nodes communicate with one another by employing a number of tokens circulating around the network [50]. Alternatively, the token grid can be considered to be a multidimensional extension of the token ring, so that nodes can share access to a mesh-connected topology by a number of overlapping rings. Nodes achieve transmission rights on the network by acquiring one of the several circulating tokens. Simple, distributed mechanisms are employed to enable rings to couple and decouple with one another in a dynamic fashion. For additional details on the token grid architecture, see [50]. Additional related work, referred to as the MultiMesh architecture, can be found in [51].
Hypercube

The hypercube interconnection pattern has been actively investigated for multiprocessor architectures, and it is starting to receive attention as a virtual topology for multihop lightwave networks as well [18, 20, 37]. First we will discuss the binary hypercube, followed by the generalized hypercube.

Binary Hypercube - The simplest form of the hypercube interconnection pattern is the binary hypercube [37]. A p-dimensional binary hypercube has \( N = 2^p \) nodes, each of which have \( p \) neighbors. A node requires \( p \) transmitters and \( p \) receivers, and it employs one transmitter-receiver pair to communicate directly and bidirectionally with each of its \( p \) neighbors. Any node \( i \) with an arbitrary binary address will have as its neighbors those nodes whose binary address differ from node \( i \)’s address in exactly one bit position. An eight-node binary hypercube is shown in Fig. 8.

The merits of this structure are its small diameter \((\log_{2}N)\) and short average hop distance \( (N(\log_{2}N)/2(N-1)) \). Its disadvantage is that the nodal degree increases logarithmically with \( N \). For other properties of this network, see [37].

Generalized Hypercube - The radix in the nodal address notation can be generalized to arbitrary integers, thereby resulting in the generalized hypercube structure [18, 20, 37]. This structure can employ a mixed radix system to represent the node addresses. Let the number of nodes be given by \( N = \Pi_{i=1}^{p} n_{i} \) where the \( n_{i} \) are positive integers. A node’s address \( P (0 \leq P \leq N-1) \) is represented by the \( p \)-tuple \((m_{1}, m_{2}, \ldots, m_{p})\) where \( 0 \leq m_{i} < n_{i} \) and \( 1 \). Thus, we have \( P = \sum_{i=1}^{p} m_{i} \cdot w_{i} \), where \( w_{i} = \Pi_{j=1}^{i-1} n_{j} \).

The generalized hypercube shares similar merits and demerits as its binary version, except that it is more flexible in accommodating different numbers of nodes and their interconnection patterns. For more information on its characteristics, see [37].

Near-Optimal Node Placement

Given the flexibility of nodal interconnection patterns, one can construct an optimal regular structure which not only preserves a regular structure’s simplified routing property, but also satisfies an optimality criterion such as minimum networkwide mean packet delay. Such studies have been reported for the linear dual-bus structure [8, 47], while work on other structures is in progress [10]. In this section, various algorithms for placing nodes in a near-optimal fashion on a linear dual bus are reviewed.

Motivation for optimally structuring a linear bus is due to recent major development in data networking, viz. the standardization of the distributed queue dual bus (DQDB) as the medium access control protocol for the IEEE 802.6 metropolitan area network (MAN) [17]. This network structure consists of two linear unidirectional buses. The incorporation of slot reuse techniques, which enable spatial reuse of the channel bandwidth, has also been proposed for DQDB [9, 24, 43]. Thus, the network nodes may be considered to be connected via direct point-to-point links to form a linear multihop network, as shown in Fig. 9. The specific optimization problem may be stated as follows: Given that the network nodes must be connected linearly and that the node positions in the linear network may be adjusted by properly tuning their (optical) transmitters and receivers, what is the best pattern for interconnecting them?

Generally, there are \((N!/2)\) different ways in which \( N \) nodes may be arranged in a linear fashion. From among these \((N!/2)\) structures, identifying the optimal structure(s) is a computationally intensive problem. Therefore, the works in [8, 47] investigate fast heuristic algorithms for constructing near-optimal structures. These algorithms can be classified into two categories: flow- and delay-based heuristics. The flow-based heuristics are concerned with minimizing the maximum flow in any link (as in [36]), given that the network’s traffic matrix is known. The delay-based heuristics require the knowledge of not only the traffic matrix but also the distance matrix, viz. the vector of distances between nodes and the hub. The goal of these algorithms is to find the node order which will minimize the networkwide mean packet delay. Our treatment of the heuristics first will consider static traffic patterns, but dynamic algorithms allowing traffic changes also can be developed [8].

Flow-Based Heuristics

These heuristics require that, for a given traffic matrix, the flow through the most heavily congested link in the network be minimized. The nodes are connected via full-duplex links and active interfaces, and traffic from the source node is relayed by intermediate nodes toward its destination where it is absorbed (Fig. 9). The average traffic between nodes in such an \( N \)-node network may be represented by an \( N \times N \) matrix \( F \), where \( f_{ij} \) represents the average traffic from node \( i \) to node \( j \).
In the photonic bus network (PBNets) study [47], the algorithms build up the network around a partially formed PBNets by adding nodes to it one by one. Among the various heuristic algorithms studied in [47], the Min-Max algorithm is found to perform the best and its outline follows.

Min-Max - The optimality criteria of this algorithm is to locally minimize the maximum flow in any link in the PBNets as it builds up starting with a single network fragment (viz. a partial PBNets). New (unadded) nodes are selected on the basis of the maximum traffic flow between them and the super node consisting of all nodes currently belonging to the partial PBNets. The new node is added to the side of the partial PBNets such that the maximum utilization (on any link) is minimized.

In [8], instead of adding nodes to a partially formed network fragment as in [47], the algorithms search for parallel techniques as well. Specifically, the algorithms build up the bits and pieces of the network in multiple fragments instead of around a single fragment.

SORTed First-fit - In this algorithm, first the elements of the traffic matrix are sorted in nondecreasing order. Then the algorithm steps through this sorted list to select candidate chains (of connected nodes) to be joined. Let \( N \) be the next highest element in the sorted list. Then, if both nodes \( i \) and \( j \) are end nodes of two chains, a larger chain is formed by joining these two ends; otherwise the next highest element is considered. The time complexity of this algorithm is \( O(N \log N) \).

First-fit SUPERnodes - This algorithm is operated in \((N - 1)\) steps, and at each step, two chains of nodes are connected together and the size of the effective traffic matrix is reduced by one. If the two chains \( k \) and \( j \) are to be connected to form a longer chain, then the two end nodes of these two chains are connected to the traffic matrix between chains is updated.

First-fit on Binary TREE - This algorithm is based on a bottom-up design technique. Initially, the algorithm constructs chains of length two. Then, pairs of these chains are linked to obtain chains of length four (and possibly of length three as well, if \( N \) is odd). This process of doubling the length of the chains and halving their number is continued until a single chain is formed. The time complexities of this, the previous, and the Min-Max algorithms are \( O(N^2) \).

Divide and Minimize Link Flow (DMP) - In this algorithm, first the \( N \) nodes are partitioned into two groups \( G_1 \) and \( G_2 \) consisting of \([N/2]\) and \([N/2]\) nodes, respectively. This partitioning attempts to minimize the flow through the link connecting the two groups \( G_1 \) and \( G_2 \) and is carried out as follows. Initially, one of the two nodes \( i \) and \( j \) with the minimum \( u \) is placed in \( G_1 \) and the other one is placed in \( G_2 \). Then, from among the remaining nodes, node \( k \) is chosen such that the differential in traffic between node \( k \) and all nodes in \( G_1 \) and between node \( k \) and all nodes in \( G_2 \) is the maximum, and it is added to \( G_2 \). This process is repeated alternately for the two groups until all nodes are placed. Then, the nodes within each of these two groups are ordered as follows. First, a node from \( G_2 \) is chosen such that if this node were removed from \( G_2 \) and added to \( G_1 \), then the flow from the new \( G_2 \) to the new \( G_1 \) would be minimum (over all possible choices of \( k \)). This node is placed at the \( (\lfloor N/2 \rfloor + 1) \) position in the linear topology being constructed. Using a similar approach, the other nodes in \( G_2 \) as well as the nodes in \( G_1 \), are arranged. Performance of this algorithm is generally superior to those of the previous algorithms, and its time complexity can be shown to be \( O(N^2) \).

Iterative Approach - This approach is based on finding a Hamiltonian chain that optimizes a certain cost function. Consider an \( N \)-dimensional surface, composed of points representing the values of the cost function for all different permutations of \( [1, 2, \ldots, N] \). Then, this surface will have several local minima and one or few global minima. The iterative algorithm starts by picking randomly one point \( s \) on this surface. Then, at each iteration, \( s \) is placed at the node \( s \), \((r < k; r = 1, \ldots, N - 1; k = r + 1, \ldots, N) \) if the maximum link-flow in the new sequence is lower than that in the previous one. If the maximum link-flow remains the same, then the new sequence is retained if the total link-flow is reduced. By successive execution of this operation, a point on the surface is reached when no further minimization is possible. This point is either one of the local minima or the global one. By starting from different initial points, chances of hitting the global minimum can be arbitrarily increased. However, for each iteration, this algorithm takes \( O(N^2) \) time.

For additional work on flow-based heuristics and algorithms applicable to the photonic bus network (PBNets), especially on optimal mechanisms by which some nodes can be equipped with additional receivers, see [47-49].

Delay-Based Heuristics

Load Over Distance - A set of algorithms for minimizing the average delay can be obtained by applying some of the flow-based algorithms to a transformed traffic matrix. Generally, one would like to place two nodes close to each other if their combined distance from the hub is small. Also, two nodes which have a lot of traffic between them should be placed close to each other, so that this heavy traffic may travel through node or few intermediate nodes and thus may encounter a lower delay. Following these general guidelines, the traffic matrix is transformed by dividing each of its elements by the sum of the distances of the two corresponding nodes, and flow-based algorithms are applied to the transformed matrix.

Divide and Minimize Delay and Iterative Approach - Both of these approaches are similar to their flow-based counterparts.

Dynamic Load Balancing

In [8], the case where the prevailing traffic conditions may change is treated so that the node sequence may need to be readjusted in order to maintain optimality. Specifically, the characteristics of a mechanism by which nodes can dynamically perform load balancing, and thereby reduce the maximum link-
flow as well as the total link-flow of the system are investigated. The operations to achieve a better network configuration are performed by the nodes in a localized and distributed fashion using only local information available to them. See the technical report [8] for additional details.

**Shared-Channel Multihop Systems**

Channel sharing was introduced in [1] with the goal that the utilization of a multihop link can be improved if more than one transmitter-receiver pair is allowed to access the same wavelength channel. The work in [1], as well as most other work on channel sharing [6, 25, 30], advocates the use of time division multiplexing (TDM) as the multiple access mechanism for sharing a common channel. However, other channel arbitration strategies are studied in [18, 20] in connection with a shared-channel hypercube architecture. We will discuss channel sharing strategies proposed for ShuffleNet, MSN, and hypercube, followed by another channel-sharing concept called subcarrier multiplexing.

**Channel Sharing in ShuffleNet**

In the original ShuffleNet work in [1], (p,k) ShuffleNets with p = 2 are considered, and the following channel sharing strategy is used. Recall that there are k columns of nodes and p nodes per column in a (p,k) ShuffleNet. All nodes in the same row share their transmissions on p (p = 2) channels. That is, although node i (0 ≤ i ≤ p−1) in an arbitrary column transmits to nodes j, j + 1, ..., j + p − 1 in the next column (where j = (i mod p^k) * p) via p distinct channels, all k of the n^k transmitters (n = 0, 1, ..., p) from the i^k nodes in all columns must share the same channel.

Due to channel sharing, a packet can reach its destination in fewer hops, on the average. For p = 2, an upper bound on the expected number of hops is obtained to be [1]

\[
R^* ≤ \frac{1}{2^k} \left[ 2 + (k - 1)2^k \right]
\]

The above result is not directly comparable to the mean hop distance for the dedicated channel case in equation (1) since (2) includes self-hopping along which was not included in (1); however, this factor can easily be corrected.

Channel sharing in generalized ShuffleNets is treated more rigorously in [30], but a different shared-channel allocation mechanism is employed. Specifically, it is required that, for i = 0, 1, ..., p, nodes i, i + p, i + 2p, ..., i + (k − 1)p in a column transmit on a shared channel which, in turn, is received by nodes j, j + 1, j + 2, ..., j + p − 1 in the next column where j = (mod p^k) * p. For other properties of shared-channel ShuffleNets, see [30].

**Channel Sharing in the MSN**

The performance of the Manhattan Street Network (MSN) and two of its shared-channel variants are compared in [25]. Note that a node in the MSN can be considered to be flanked by two imaginary rows and two imaginary columns. Each such imaginary row and column is treated as a separate wavelength channel in [25]. In shared-channel variant 1 in [25], called broadcast-wavelength structure 1 (BW1), each node is equipped with four transmitters and four receivers, and it can transmit on and receive from all four wavelengths flanking itself.

In the other variant, called broadcast-wavelength structure 2 (BW2), each node is equipped with two transmitters and two receivers, and it can transmit on and receive from one row channel and one column channel. For channel sharing, TDMA is used. Channel sharing reduces the network diameter, the number of wavelengths required, and the average hop distance between nodes.

It also is shown in [25] that, for the uniform traffic case, both BW1 and BW2 can support much higher aggregate network throughput than the original MSN, and that BW1 can support more traffic than BW2 because nodes in BW1 have twice the number of transmitters and receivers than nodes in BW2 do.

**Channel Sharing in the Generalized Hypercube**

Recall that the number of nodes in a generalized hypercube is given by \( N = \Pi_{i=1}^{p} n_i \) where \( p \) is the number of dimensions as well as the nodal degree and \( n_i \) is the number of nodes in the \( i \)-th dimension. Channel sharing is performed as follows. There are \( N \Pi_{i=1}^{p} p_i \) channels spanning the \( p \)-dimension, and the identity of these channels is given by \( c_0, c_p, ..., c_{n-1} \). Each channel spans dimension \( i \), where \( 0 ≤ i ≤ n_i - 1 \) and an X in position \( i \) denotes that the channel is a shared channel. A node \((m_0, m_1, ..., m_{p-1})\) is connected to an \( i \)-dimensional channel \( c_0, c_p, ..., c_{n-1} \). Note that any node is connected to \( p \) different channels, all spanning different dimensions.

For channel arbitration, three control strategies are used. The first is random access with a control subchannel—in particular, a slotted ALOHA control channel and ALOHA-based data channels as in [28] is employed. The second approach is a random access protocol without a control subchannel; specifically, a slotted ALOHA approach is used to access the data channels as in [19]. The third approach is a statically allocated media access protocol, viz. a multichannel TDMA mechanism [15]. As expected, simulation results in [18] show that the random access approaches perform better at light loads, resulting in lower delay, but the TDMA approach can support higher throughputs. Additional details can be found in [20].

**Multihop Systems Based on Subcarrier Multiplexing**

The concept of subcarrier multiplexing was introduced in [16]. It is based on the observation that, although rapid tuning between wavelength channels may not be feasible today, systems can be built which employ rapid tuning between subcarriers within the same wavelength. A multihop architecture exploiting the above concept has been proposed in [42].

Under this scheme, each wavelength channel is partitioned into a number of nonoverlapping subcarrier frequencies (subchannels). A node transmits on a fixed wavelength channel, but it can transmit on any subcarrier within the channel. Note that two or more nodes can transmit conflict-free information on the same channel if they do so on different subcarrier subchannels.
A significant amount of further research on multihop systems is expected, and new architectures for multihop systems also are expected to be investigated.

![Figure 10. Classification of multihop network architectures.](image)

Free information on the same channel if they do so on different subcarrier subchannels. Since a node can tune rapidly between subcarriers on its transmit wavelength, it can transmit packets in quick succession to other nodes which receive on various subcarriers within the transmitting node's wavelength channel. Information destined for nodes which do not receive on the wavelength of the transmitting node must be routed through intermediate nodes. The work in [42] demonstrates how multihop topologies such as ShuffleNet and de Brujin graph can be realized by using the above subcarrier multiplexing approach.

Finally, TDM-based channel sharing is also briefly discussed in the minimum-mean-packet-delay-based topological optimization work in [6]. A genetic algorithm was employed to solve the shared-channel optimization problem.

**Conclusion**

Wavelength division multiplexing (WDM)-based local lightwave networks employing the multihop approach were surveyed in this article. The general characteristics of multihop systems were discussed, and various multihop approaches were reviewed. Figure 10 provides a summarized classification of these multihop systems.

A significant amount of further research on multihop systems is expected. New architectures for multihop systems also are expected to be investigated, e.g., variants of the basic structures studied here in order to reduce the structure's mean hop distance, e.g., the work in [21] studies a variation of the hypercube with reduced diameter. Optimal node placement algorithms in rings, ShuffleNet, and other regular structures in order to counteract the nonuniformities in the offered traffic are expected to be developed. The modularity of regular structures, viz., how can one add nodes to or delete nodes from existing regular structures, is an important issue. One can refer to such structures as injured regular structures. How can these multihop structures reorganize their nodal connectivity pattern to restore optimality when the traffic pattern changes is another question which needs to be answered.

Although this article was based on local lightwave networks employing the broadcast-and-select mechanism, work on optical WDM wide area networks (WANs) [13, 14, 29, 46] also has been initiated based on photonic wavelength switching at intermediate nodes [11]. The latter category of networks does not need a centralized hub as local lightwave networks do. Hence, WDM WANs can employ spatial reuse of wavelengths in different parts of the network, thereby increasing the amount of concurrency in the network. Some work on embedding a ring [41] and a hypercube [13, 14] on WDM WANs have been reported, and more activity in this topic is anticipated.

An experimental prototype of a multihop network employing WDM and subcarrier multiplexing, and referred to as TeraNet, also has been reported [26].
Acknowledgments

I thank Subrata Banerjee for his assistance with several of the illustrations. I also thank the following individuals for pointing out several references and for their comments on an earlier draft of the paper—Imrich Chlamtac, Patrick Dowd,Aura Ganz, Nader Mehravari, Terry Todd, and Zhenheng Zhang.

References


The modularity of regular structures, viz., how can one add nodes to or delete nodes from existing regular structures, is an important issue.
On p. 18, left column, first full paragraph, line 4, references should be "[8, 11]."
On p. 16, left column, first full paragraph, line 7 should read "... and pre- transmission-coordination protocols are discussed. While ..."
On p. 16, third full paragraph, line 9 should read "... spaced by 2.2 GHz."
On p. 16, third full paragraph, line 10 should read "... 5-GHz-spaced..."
On p. 16, paragraph on LAMBEANET, both references to "[431]" should be replaced by "[18]."
On p. 18, second paragraph on the RAINBOW subsection, second to last line should be "... should support 300 packet-switched nodes..."
On p. 17 and 18, the three tables should have column headings "1+1" and "1+2."
On p. 17, left column, first line after the table should read "... in slot 1..."
On p. 17, right column, lines 5 and 6, i should be subscripts of t and z.
On p. 17, table in right column, one of the two entries in the last row and column should be "[2, 11]" instead of "[2, 1]."
On p. 18, table in left column, one of the two entries in second row and third column should be "[2, 11]" instead of "[1, 1]."
On p. 18, table in left column, one of the two entries in third row and fourth column should be "[2, 11]" instead of "[2, 0]."
On p. 18, right column, line 3-3 should read "... and time is synchronized across..."
On p. 17, right column, line 5, should read "\q - 1 to \q + 1."

On p. 19, right column, first full paragraph, line 6 should read "... these schemes..."
On p. 21, Fig. 8 caption should read "The dotted ALOHA protocol in 1981..."
On p. 23, left column, lines 16 and 22, 2 and 4 should be superscripts in the expressions "[QNY]" and "[QNY]."
On p. 23, right column, line 23, "[QNY]" is supposed to represent the "floor" operation.

Reference


Biography

Biswanath Mukherjee (M'84) received the B. Tech. and Ph.D. degrees from IIT Kharagpur (India) and the University of Washington, Seattle, respectively. Since 1987, he has been at UC Davis where he is currently associate professor of computer science. He is co-winner of the 1991 National Computer Security Conference Outstanding Paper Award.