

# Large Scale, All-Fiber Optical Cross-Connect Switches for Automated Patch-Panels

Anthony S. Kewitsch, *Member, IEEE*

**Abstract**—Transparent, all-optical cross-connect switches provide non-blocking, one-to-one reconfigurable interconnections between any of  $N$  input fibers and any of  $N$  output fibers. Cross-connects based on *all-fiber* rather than free-space interconnections have the potential for low opto-mechanical complexity, miniaturization, high optical power handling and ideal optical performance. In this paper, we propose and analyze novel architectures for large-scale, all-fiber cross-connects based on the application of Theory of Knots and Braids to fiber switching and compare them on the basis of scalability, modularity and reconfiguration time. A prototype 140 input by 140 output, automated all-fiber cross-connect is developed to demonstrate these concepts.

**Index Terms**—Fiber-optic switch, cross-connect, automated patch panel, switch topology, theory of knots and braids.

## I. INTRODUCTION

**A**  $N \times N$  optical cross-connect switch interconnects any of  $N$  optical inputs to any of  $N$  optical outputs in a one-to-one, optically transparent and arbitrary fashion. Today, such functionality is achieved by manually connecting fiber optic patchcords between input connector terminals and output terminals arrayed on a patch-panel. The configuration and inventory management of these physical connections is one of the few remaining aspects of the network that is still a manual process.

Automation has already been successfully applied to layers 0 to 6 of today’s telecommunications networks. The only layer not yet automated is layer –1, which includes the vast number of physical interconnections made between transceivers, reconfigurable add/drop multiplexers, wavelength division multiplexers, electronic cross-connects and routers, fiber amplifiers and test equipment [1].

Previous work in the field of mechanical cross-connects includes the robotic approach described by S. Sjolinder [2], in which opposing fiber connectors independently translate along separate, orthogonal, linear tracks in two separate planes so that any input optical fiber and any output fiber can be coaxially aligned at an array of  $N^2$  insertion points formed between the two planes. In addition, various implementations of cross-connects in which fiber optic connections are configured by a robotic fiber handler have been described in the literature [2]–[7]. Typically, these approaches utilize an  $N \times N$  internal honeycomb matrix where reconfigurable fiber connections are made.

Manuscript received June 26, 2008; revised August 18, 2008, September 02, 2008. First published April 21, 2009; current version published July 09, 2009.

A. S. Kewitsch is with Telescent Inc., Santa Monica, CA 90403 USA (e-mail: kewitsch@telescent.com).

Digital Object Identifier 10.1109/JLT.2008.2006280

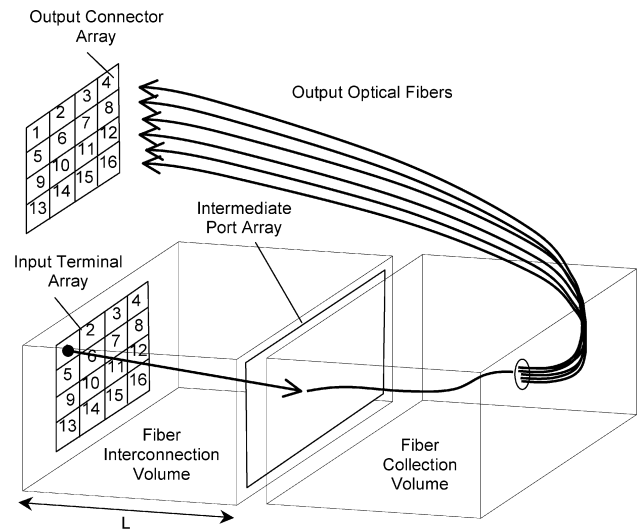


Fig. 1. Generic strand interconnection geometry.

In this paper, we propose and analyze a new type of all-fiber cross-connect exhibiting unique characteristics such as modularity, scalability to thousands of ports and mechanical simplicity. We apply the Theory of Knots and Braids [8], [9], prevalent in such diverse fields as particle physics, quantum gravity, DNA and polymer physics, to develop novel switch architectures and computational algorithms to guide the reconfiguration of dense systems of intermixed fiber interconnects, for unlimited numbers of strands and permutations. In this system, the arbitrary, non-blocking reconfiguration of any fiber strand can be achieved based on knowledge of the configuration of all other fibers within the interconnect.

## II. FIBER OPTIC CROSS-CONNECT SYSTEM ARCHITECTURE

This all-fiber cross-connect consists of a fiber interconnection volume bounded at opposite ends by parallel planes separated by a distance  $L$  (Fig. 1). Within this volume, a large number (100’s to 1000’s) of optical fiber strands linking inputs to outputs intermix. The first plane coincides with an input terminal array, where the connectorized optical fiber strands internal to the cross-connect are interfaced with external fiber patchcords, and the second plane coincides with an intermediate port array internal to the cross-connect, through which these same optical fiber strands pass to the fiber collection volume and ultimately terminate at the output connector array. Only those internal fiber connections made at the input terminal array are reconfigurable. The number of input terminals and intermediate ports are taken to be identical; however, in general they can be different. *Terminals* are here defined as reconfigurable points where the fiber

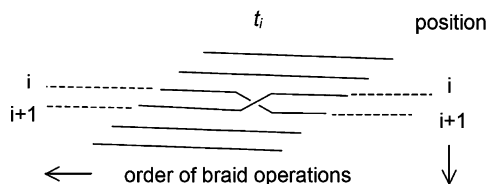


Fig. 2. This braid diagram illustrates the function of an elementary braid generator on individual strands.

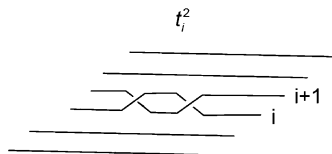


Fig. 3. Diagram of a knotted braid with two crossing points.

strands are connectorized and attached to mating adapters, and *ports* are defined as fixed points through which contiguous fiber strands pass through as they enter the fiber collection volume and ultimately terminate in outputs.

One would like to develop an interconnection topology and reconfiguration algorithm enabling one end of a strand within the interior of the input array to be maneuvered so that this strand passes through the interconnect volume without entangling other strands. This requires that any fiber optic strand can be arbitrarily reconfigured within this volume regardless of the configuration of the surrounding strands. To determine the required reconfiguration trajectories, it is of particular value to develop representations of strands based on the mathematics of topology; namely, the Theory of Knots and Braids [8], [9].

To develop a generalized reconfiguration algorithm, the set of  $N$  interconnections describing a particular switch state is represented mathematically by an  $N$ -stranded Braid Group. A Braid Generator  $t_i$  is defined over the Braid Group to represent the physical crossing of a strand in position  $i$  over a strand in position  $i + 1$  (Fig. 2). For example, the  $N$ -stranded Braid Group may be denoted as  $B_N = (t_1 \dots, t_N)$ . A particular braid element within this group is described by the product of Braid Generators, where the terms in the product are ordered from right-to-left, corresponding to crossings positioned from right-to-left along each strand. Note that the subscript  $i$  does not refer to a physical strand, but rather to its particular position within the braid.

Two crossings commute only if they do not operate on adjacent positions within the strand; that is,  $t_i t_j = t_j t_i$  if  $|i - j| > 1$  (the usual notion of commutativity under multiplication does not apply to the braid generator). A knot occurs when one strand fully wraps around another and is represented by a generator  $t_i^2$ , for example (Fig. 3). Note that a strand does not create a knot if it simply passes over another strand (represented by  $t_i$ ), because for such an arrangement, strands can be thought of as lying in different layers, one layer on top of another. Therefore, a knotted strand would include the generator  $t_i^x$ , where  $x > 1$ . The layered structure is of practical value because it allows groups of strands to be split into two parts, so that a strand can slide or cross through a braid without knotting. This will be a key feature enabling the arbitrary reconfiguration of strands.

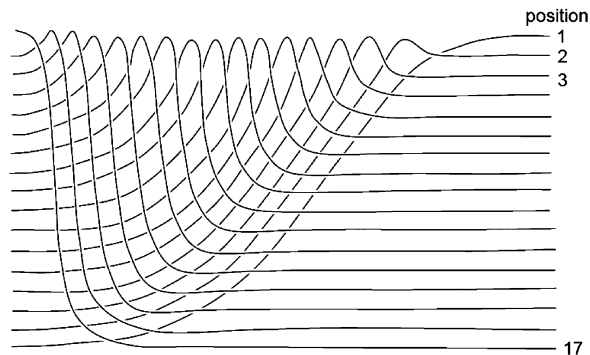


Fig. 4. Braid diagram of a non-repeating negative braid in which the strands are inverted.

It is clearly a requirement of all-fiber cross-connect systems that interconnections remain knot-free during an unlimited number of reconfigurations. Such configurations correspond mathematically to a braid composed of strands with  $|x| \leq 1$ , conventionally called positive ( $x > 0$ ) or negative ( $x < 0$ ) *non-repeating* braids (positive if the braid only has positive crossings, that is, the front strands have a positive slope). For positive braids, the strand at position  $i$  passes over the strand at position  $i + l$  and for negative braids, the strand at position  $i$  passes under the strand at position  $i + l$ . Therefore, to prevent physical entanglement within the switch interconnect volume, it is necessary that each strand be composed of generators  $t_i^x$  with  $|x| \leq 1$ .

A particular example of a negative, non-repeating braid is given by  $(t_1^{-1} t_2^{-1} \dots t_{16}^{-1})(t_1^{-1} t_2^{-1} \dots t_{15}^{-1}) \dots (t_1^{-1} t_2^{-1}) t_1^{-1}$  and illustrated in Fig. 4. While this braid includes the product of number of generators  $t_i^{-1}$  on the same position  $i$ , it does not include terms  $t_i^x$  with  $|x| > 1$  because of non-commutativity. Each individual strand within this non-repeating braid can be thought of as residing in its own layer, which can be individually peeled away from other layers without entanglement.

In general, there are several classes of fiber interconnection systems. In the discussion that follows, we compare three classes of interconnections: those in which strands link (1) a two-dimensional array of input terminals and two-dimensional array of intermediate ports, (2) a two-dimensional array of input terminals and one-dimensional array of intermediate ports, and (3) a one-dimensional array of input terminals and one-dimensional array of intermediate ports.

#### A. Interconnections Between a 2-D Array to 2-D Array of Points

A first class of interconnection system corresponds to a collection of fiber strands linking a two-dimensional input array and a two-dimensional intermediate array. Assume there are  $a$  columns by  $b$  rows of terminals at both the input and intermediate arrays. In general, the number of rows and columns can be dissimilar; in fact, it may be advantageous for some applications that the input array has a larger number of terminals than the intermediate array.

Fiber interconnects represented by an  $N$ -stranded Braid Group can generate an infinite number of braid configurations. If the interconnect paths are not straight, as would be the case

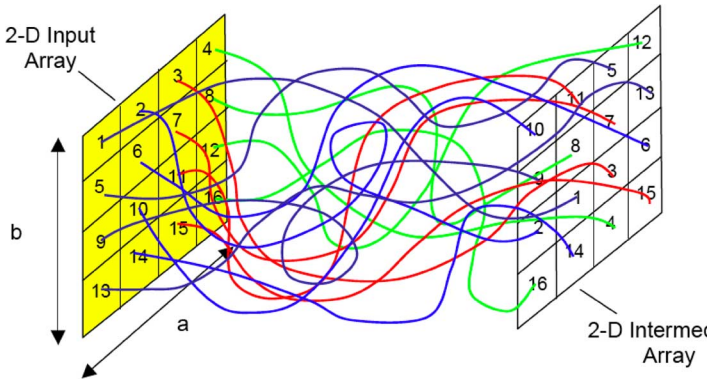


Fig. 5. General system of strands having a 2-D to 2-D array of endpoint.

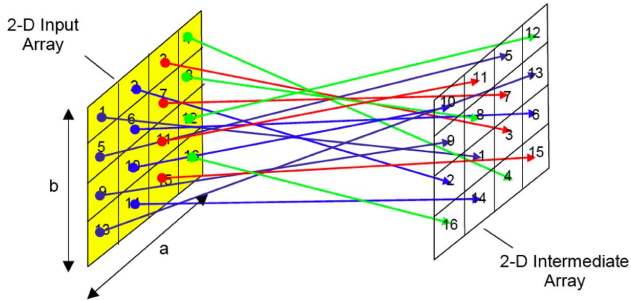


Fig. 6. 2-D to 2-D geometry with straight-line (and shortest distance) interconnections.

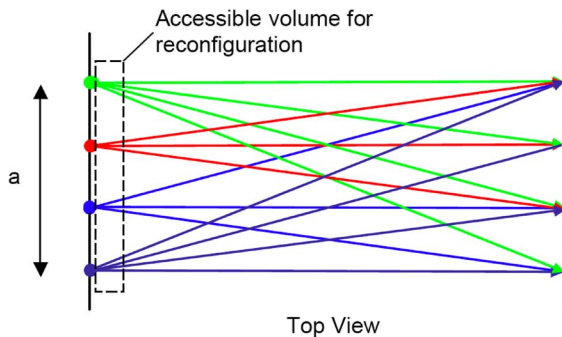


Fig. 7. Top view of straight-line interconnections with 2-D arrays of endpoints.

when the length of the strand is greater than the straight-line path between input and intermediate planes, knotting of strands is possible. Fig. 5 illustrates this situation for a  $16 \times 16$  cross-connect. Excess strand lengths within the interconnect volume result in indeterminate crossing points and excessively sharp bending of strands, leading to potential knotting, high insertion loss and even damage to the optical fibers.

A first requirement to prevent entanglement is therefore that the interconnect system maintains least-path, *straight-line* strands. Each interconnect state is then described by a braid comprised of linear strands bridging the input and intermediate arrays, as shown in Figs. 6 and 7. We assume initially that the strands have infinitesimal thicknesses so that deviations from linear paths at potential crossing points have a negligible effect on the interconnection trajectory.

By maintaining straight-line paths, any fiber optic strand connected to the front input terminal array may be reconfigured by

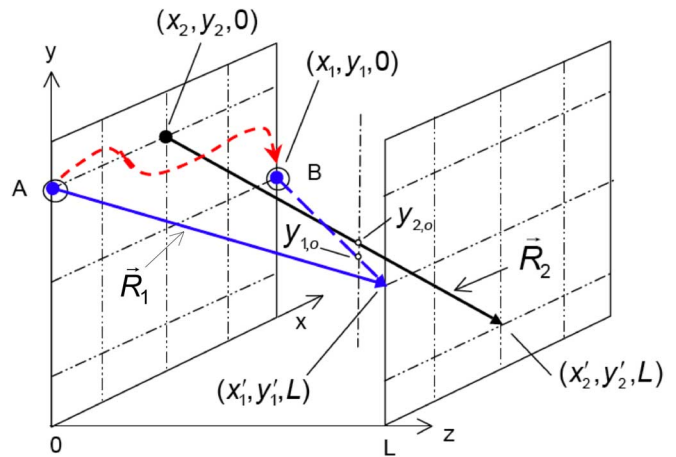


Fig. 8. Trajectory to avoid entanglement for an arrangement of two strands.

physically translating its endpoint within the interstitial regions immediately between the arrayed interconnections at the input terminals. This endpoint should remain close to the plane of the input array during translation because interstitial gaps between columns of interconnects exist here and allow physical access for an actuator to reconfigure endpoints. On the other hand, the interconnects' rear endpoints extend back to the intermediate array through ports whose positions here remain fixed at all times. There is generally not an unobstructed path for an actuator to move through the intervening mixed interconnect volume back to the intermediate array; thus, the reconfiguration trajectory should be limited to the interstitial regions.

Mathematically, we can ensure that all braid configurations will belong to the group of positive (or negative) non-repeating braids by maintaining consistency in the strand ordering convention during all subsequent reconfigurations. Assuming each interconnect follows a straight-line path, there is a deterministic algorithm to move the endpoint of one strand through the interconnect volume to a new state, so that this strand and all others trace straight-line paths in the final state. An example reconfiguration from an initial port A to a final port B, following a path shown as a dotted line in the vicinity of the input array, is illustrated in Fig. 8. To simplify this discussion, only one intervening strand is shown.

To ensure that the strand corresponding to vector  $\vec{R}_1$  follows a straight-line path after translating to the destination port B, its moveable end must pass above or below intervening interconnections in a deterministic fashion through the shared volume. The resulting trajectory is computed by taking account of the locations of all other strands. The endpoints of an intervening strand joining the point  $(x_2, y_2, 0)$  within the input plane and the point  $(x'_2, y'_2, L)$  of the intermediate plane define a vector  $\vec{R}_2$ , the equations of which, parameterized here in  $s$ , are given by:

$$\vec{R}_2 = \langle X_2, Y_2, Z_2 \rangle \tag{1}$$

$$X_2 = x_2 + s(x_2 - x'_2) \tag{2}$$

$$Y_2 = y_2 + s(y_2 - y'_2) \tag{3}$$

$$Z_2 = -sL. \tag{4}$$

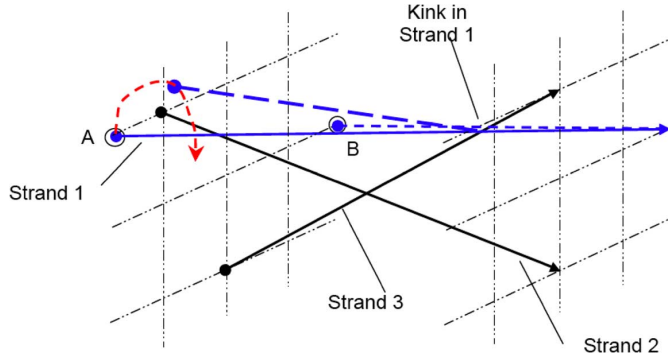


Fig. 9. A reconfiguration trajectory in which the intermediate state does not follow a straight line.

Similarly, the endpoints  $(x_1, y_1, 0)$  and  $(x'_1, y'_1, L)$  define a vector  $\vec{R}_1$  corresponding to strand 1:

$$\vec{R}_1 = \langle X_1, Y_1, Z_1 \rangle \quad (5)$$

$$X_1 = x_1 + s(x_1 - x'_1) \quad (6)$$

$$Y_1 = y_1 + s(y_1 - y'_1) \quad (7)$$

$$Z_1 = -sL. \quad (8)$$

Strand 1 must follow a path through the volume such that its final state also follows a straight-line, disentangled from other strands. This path is determined by solving for  $(Y_{1,0}, Y_{2,0})$ , the relative elevations of strands 1 and 2 respectively, when strand 1 is at its destination port and when the vertical (y) projection of strands 1 and 2 cross, so that  $X_1 = X_2$  and  $Z_1 = Z_2$ . When the X-positions are equal, the parameter  $s_o$  is given by:

$$s_o = \frac{(x_1 - x_2)}{(x_2 - x_1 - x'_2 + x'_1)}, \quad (9)$$

so that the elevations of each strand at this location are given by:

$$y_{1,o} = y_1 + s_o(y_1 - y'_1), \quad (10)$$

$$y_{2,o} = y_2 + s_o(y_2 - y'_2). \quad (11)$$

The requirements to prevent entanglement are described by the following relations: if  $y_{1,o} > y_{2,o}$ , strand 1 must pass above strand 2, and if  $y_{1,o} < y_{2,o}$ , strand 1 must pass below strand 2. If  $y_{1,o} = y_{2,o}$ , then relative positions are established using an ordering convention based on the strand endpoint positions  $(x_1, y_1)$  or  $(x'_1, y'_1)$ . This ensures that after reconfiguration and for strands of infinitesimal thickness, strand 1 follows a straight-line path unaffected by strand 2. By induction, applying this requirement to all other intervening strands in a similar fashion to strand 2 would ensure that strand 1 follows a straight-line path throughout the entire interconnect volume, for any number of intervening strands.

Fig. 9 illustrates a reconfiguration example with two rather than one intervening strand. In this situation, the input end of strand 1 must move above strand 2 to satisfy the above condition and maintain a straight line at the destination port. While this algorithm ensures that the final state is a straight line, it does not guarantee that the strand follows a straight line for intermediate

states passed through during reconfiguration. Physical interference with intermediate strands such as 3 can occur during the intermediate state, and as a result, the location of intermediate strands will be perturbed. Strand 1 may “kink” due to interference with strand 3 when passing over strand 2. For large numbers of strands, the number and degree of kinks will be significant. The amount of perturbation depends on the force balance between interacting strands. Therefore, the intermediate configuration of strands must be solved self-consistently for each intermediate state based on the knowledge of the tension with all strands.

In practice, tensioning can be provided by spring-loaded take-up spools [10] attached to the fiber strands after passing out of the open interconnect volume through the intermediate port array to the storage volume. These spools retain excess fiber strand lengths outside of the interconnect volume and keep the strands straight under slight tension. Each strand may experience different tension depending on its extended length and dissimilarities in friction. Moreover, the non-straight-line paths during intermediate reconfiguration states may require a non-negligible and non-deterministic additional length.

Moreover, there is a configuration (i.e.,  $y_{i,o} = y_{j,o}, x_{i,o} = x_{j,o}$  for all  $i, j$  strands) in which the strands all cross at the same central point within the interconnect volume. Because of the strands' finite thicknesses, a single dense cluster of crossing points is formed. The physical thickness of the crossover point depends on the physical diameter of the strands. The diameter of this cluster is approximated by:

$$D = 2d_o \sqrt{\frac{N}{\pi}}, \quad (12)$$

where  $N$  is the total number of strands, each of diameter  $d_o$ . For an  $N = 1000$  stranded cross-connect with 0.6 mm diameter strands, the cluster diameter  $D$  is 21.4 mm.

A strand undergoing reconfiguration must maneuver through this cluster freely; however, it is in physical contact with a large number of surrounding strands that inhibit its motion. This limits the ability of the 2-D to 2-D system of interconnects to be arbitrarily reconfigured.

The 2-D to 1-D system described next separates this 3-D crossing point into several physically separate and independent crossing points. The layered structure of these crossing points will then enable the cluster to be split into two parts and therefore provides an unobstructed path for the strand undergoing reconfiguration to pass through.

## B. 2-D Array to 1-D Array

In this class of fiber interconnection, fiber strands link a two-dimensional input array and a one-dimensional intermediate array. The input array consists of  $a$  columns by  $b$  rows and the intermediate array consists of 1 column and  $N = a \cdot b$  rows. The resulting braid of strands exhibits a high degree of “order” within the interconnect volume. This order is manifested in the formation of a series of physically independent, orthogonal zones or “subbraids” within the interconnect volume, each zone associated with a particular column of the input terminal array. Arbitrary reconfigurations across any number of columns are achieved by crossing through each independent, orthogonal

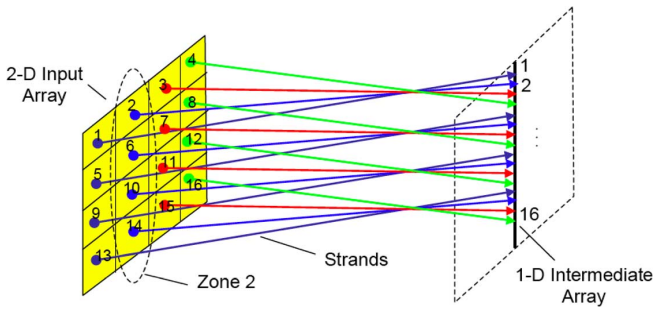


Fig. 10. System of strands joining a 2-D to 1-D array of endpoints.

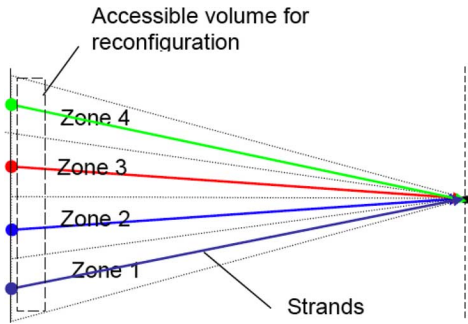


Fig. 11. 2-D to 1-D array of interconnections, top view.

zone, one at a time in a sequential fashion. Straight-line interconnections are maintained even during reconfiguration.

For example, the interconnection strands between a  $4 \times 4$  input terminal array and the  $16 \times 1$  intermediate port array are shown in Figs. 10 and 11. The mapping of the 2-D input array to a 1-D intermediate array as in Fig. 10 performs the mathematical equivalent of “combing” the interconnection braid into independent “subbraids”.

This interconnection geometry results in a deterministic arrangement that eliminates the potential for circuit interference during reconfiguration. The braid group  $B_n = (t_1 \dots, t_n)$  reduces to the Subbraid Group  $B_a = (t_1 \dots, t_a)$ , where  $a$  is the number of rows of the input array. The reconfiguration trajectory separates into a series of independent reconfiguration trajectories, one per subbraid. Since the strands within each subbraid lie approximately within a plane, the computation of these trajectories is considerably simplified.

The strands are inserted into the subbraid during reconfiguration and maintained in the proper order such that each subbraid is strictly positive or negative and non-repeating. Any two strands of a non-repeating braid cross at most once. The strands of the non-repeating braid are overlaid back to front without intertwining and effectively lie within separate layers, eliminating the tendency to tangle.

This interconnect geometry results in several unique advantages. At any time during the reconfiguration process, strands do not span more than one zone, regardless of their configuration. This eliminates the indeterminism afflicting the 2-D to 2-D system of interconnections. The algorithms to re-arrange any strand in a non-blocking fashion requires knowledge of each interconnect’s intermediate array row  $m$  and the sign of the braid. To move a strand  $n$  within column  $i$  to a column  $j$ , the subset

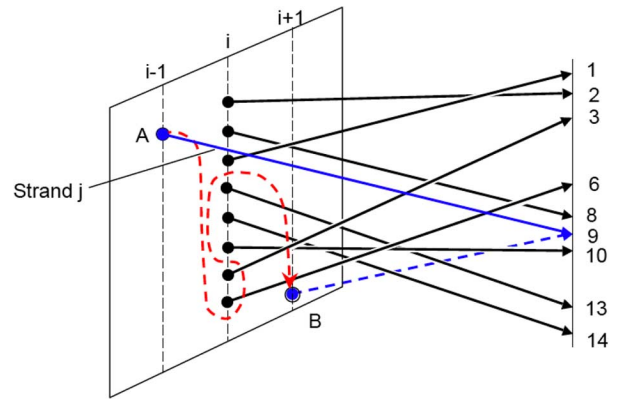


Fig. 12. Example trajectory to reconfigure a strand in the 2-D to 1-D geometry.

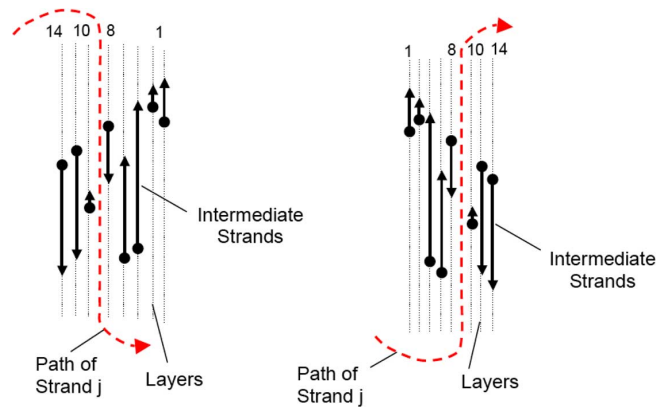


Fig. 13. End-on view of a non-repeating braid, each strand occupying a separate layer (positive braid ordering, left; negative braid ordering, right).

of strands in column  $i$  lying between strand  $n$  and strand  $j$  must be identified. The endpoint of strand  $n$  must then trace out a continuous path at the interior to the input terminal array when crossing each intermediate subbraid, passing below the subset of strands with  $m > n$  and above the subset of strands with  $m < n$ .

In the particular example illustrated in Fig. 12, a strand passes from column  $i - 1$  to column  $i + 1$  by traversing an intermediate subbraid  $i$  in a manner that avoids entanglement. If the strand were reconfigured by moving the end of strand  $i$  from a port A to a port B along a direct straight-line path in the plane parallel to input array, the strand would become physically entangled with other strands of subbraid  $i$ . Entanglement prevents subsequent reconfiguration through the knotted region. A proper path of the strand endpoint is represented by the dotted line in Fig. 12, where the strand passes below those strands originating from a higher level at the switch backbone and above those strands originating from a lower level. In this example, it is clear that the straight-line path would lead to entanglement.

Since the strands within any column are configured to maintain a non-repeating braid structure regardless of the history of prior reconfigurations, each strand occupies its own layer that can be individually peeled back from the other strands within the same subbraid. Fig. 13 illustrates an end-on view of strands within the zone  $i$  for (A) positive ordering and (B) negative ordering. In this representation, the strand  $j$  passes between the

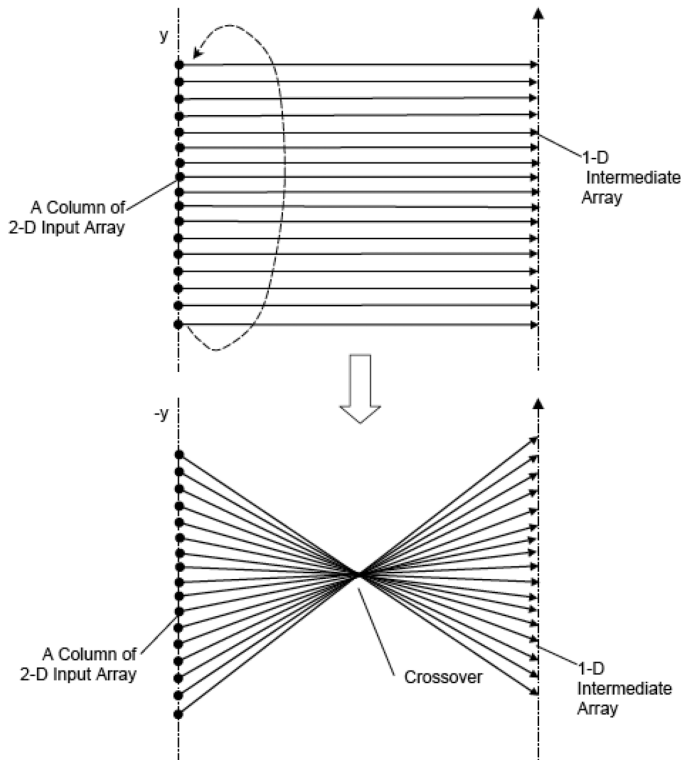


Fig. 14. Identity braid (top) and the identity braid in which the input column has been inverted in  $y$  (bottom) to create a twisted braid.

separated layers corresponding to strands  $m = 8$  and 10. The subbraid is separated at this location to provide a path for strand  $j$  that does not cross nor entangle adjacent strands.

Fig. 14 illustrates a particular subbraid type relevant to the issue of interconnect density. The upper Figure illustrates the identity subbraid  $E$ . Inversion of the  $y$  axis of the input column generates the twisted subbraid shown in the lower diagram. Physically, this subbraid would represent a cross-connect zone or column with highly mixed interconnections. In fact, this subbraid is topologically similar to the non-repeating braid represented in Fig. 4, in which the crossing points have been translated along the strands to reveal the characteristic layering of the non-repeating (non-tangled) braid.

The physical thickness of the strands becomes relevant to determine the ultimate density limit for interconnects and the maximum number of cross-connect terminals achievable within a given volume. The subbraid in the lower portion of Fig. 14 produces the highest strand density within a localized volume about the crossover point.

The thickness of the crossover point depends on the diameter of the strands and can be stretched out as wide as  $b \cdot d_o$ , where  $d_o$  is the diameter of a strand and  $b$  is the number of rows of the input array [Fig. 15(A)]. However, since each strand is under tension, the strands will arrange into a bundle to achieve a least-path configuration as shown in Fig. 15(B). The outer-most strands collapse onto the central axis to reduce their lengths. Assuming the strands form a loosely packed bunch with a circular perimeter, the diameter of the crossover region is equal to:

$$D = 2d_o \sqrt{\frac{b}{\pi}}. \quad (13)$$

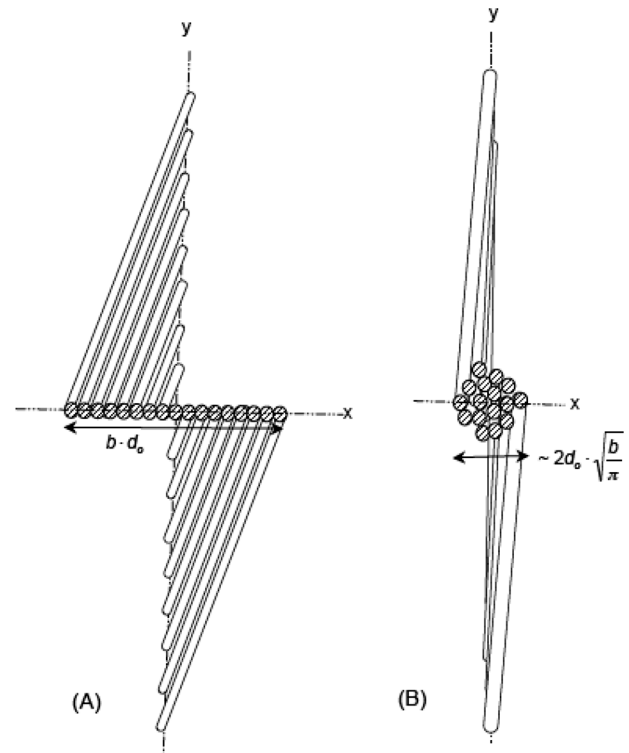


Fig. 15. Finite width strands form a cluster at crossover point.

Note that this diameter is significantly smaller than that of (12), because each subbraid only has  $b$  strands while the braid in the 2-D to 2-D array has  $N$  strands.

If all subbraids are similarly ordered, the crossover points within each zone will be in alignment along a common line parallel to the  $x$  axis. Therefore, the center-to-center spacings between each subbraid at the crossover point is established by the input array column spacing and must be greater than or equal to  $D$  to prevent interference between the subbraids. For typical values used in our experiments ( $D = 12.5$  mm,  $d_o = 0.6$  mm), there may be up to  $b = 340$  strands per braid.

In practice, one of these  $b$  strands is reserved for the additional strand that must pass through the subbraid during the reconfiguration process. In contrast to the single cluster formed in the 2-D to 2-D geometry of the previous example, the cluster formed here can be split into two clusters during the reconfiguration process, enabling an individual strand to pass freely through the subbraid within the gap formed between clusters.

Patch-panels may have 12 to 18 columns of connectors per rack, so the cross-connect has the potential to scale up to about  $6120 \times 6120$  ports based on this geometry or greater than  $10\,000 \times 10\,000$  ports for smaller diameter fiber, or for patch-panels of greater width.

The maximum time to reconfigure a port for this 2-D to 1-D array of interconnections is proportional to the length of the reconfiguration trajectory  $\sim a \cdot b \cdot dy + a \cdot dx$ , where  $dx$  is the horizontal terminal spacing and  $dy$  is the vertical terminal spacing of the input array. Examples of reconfiguration trajectories (for positive and negative braids) to move a strand from terminal A to B for a 140-terminal array with randomized interconnections are illustrated in Figs. 16(a) and (b), respectively. The numbers

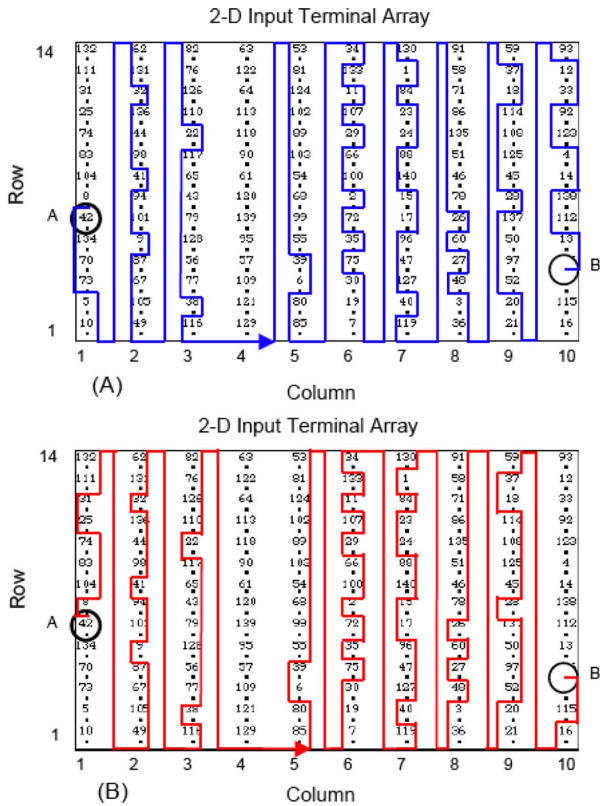


Fig. 16. Input terminal mapping and strand reconfiguration path for positive braid ordering (A) and for negative braid ordering (B).

associated with the 2-D array of input terminals in Fig. 16 correspond to the addresses of each strand traced back to the 1-D intermediate plane. The reconfiguration algorithm implemented in Fig. 16 computes a minimum length reconfiguration path, skipping those columns or strands that do not require unweaving, such as column 4.

The volume of this switch system scales gracefully as  $N$  and the width and height of the input array scales as  $a \cdot dx$  and  $b \cdot dy$ , respectively. In contrast, the vertical spacing of the  $N$  strands at the intermediate plane is small since the strands pass contiguously through this array of ports or guides and connectors are not used here. As a result, the vertical size of this switch implementation is typically limited by the size and spacing of the input array, which is equivalent to manual patch-panels.

In this 2-D to 1-D example, the strands are constrained to follow straight-line paths whose individual lengths  $l$  are bounded by:

$$L^2 \leq l^2 \leq \left(\frac{a}{2}dx\right)^2 + (bdy)^2 + L^2. \quad (14)$$

The interconnect structure for this switch geometry also provides modularity, enabling interconnections to be added and/or replaced as needed. The system incorporates a series of tensioning modules (corresponding to a row of terminals and interconnects) that can be stacked. The interconnect strands corresponding to a block of rows at the intermediate port array can be removed, even from the middle of the stack, without disrupting other interconnections. The resulting gap left after removing the

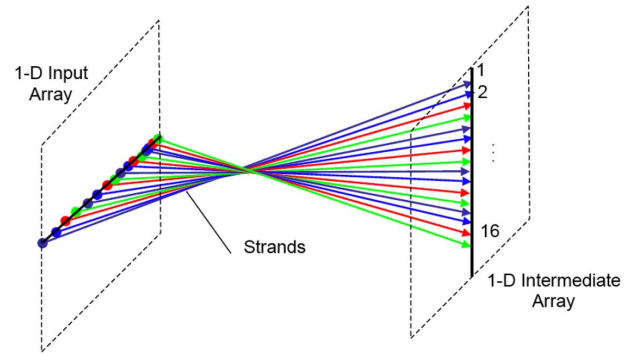


Fig. 17. 1-D to 1-D array of input and intermediate ports, perpendicular orientation.

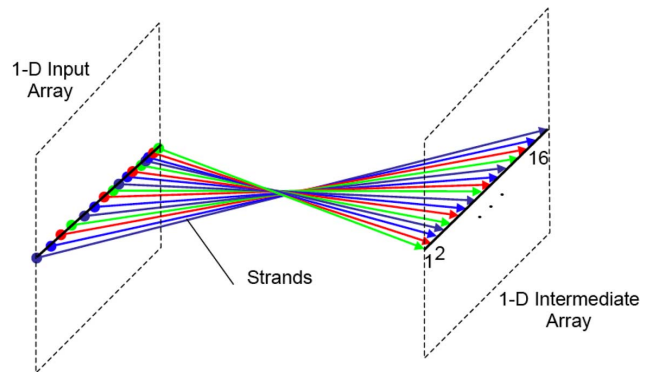


Fig. 18. 1-D to 1-D array of input and intermediate ports, parallel orientation.

block is collapsed vertically and a replacement block of interconnects can be inserted at the top of the stack. This architecture provides complete modularity and serviceability.

### C. 1-D Array to 1-D Array

In a final example, the all-fiber cross-connect is configured so that interconnect strands bridge a one-dimensional array with  $N$  rows at the input plane and a one-dimensional array with  $N$  columns at the intermediate plane. This configuration retains the desirable disentangled characteristics of the previous example, with each interconnect following a deterministic, straight-line path during and after reconfiguration. The one-dimensional arrays are oriented perpendicular (Fig. 17) or parallel (Fig. 18) to one another.

The reconfiguration algorithm for this geometry is similar to that of the previous example. For the arrangement in Fig. 17, interconnects are reconfigured in accordance with their elevations at the intermediate array, passing above those interconnects originating from a lower level and below those interconnects originating from a higher level. An example reconfiguration is illustrated in Fig. 19, where the dotted line indicates the path followed by the endpoint of a strand.

Similarly, the two 1-D arrays can be oriented parallel to one another, with the strands ordered according to their locations at the intermediate 1-D array. The interconnects will have a tendency to form a thick cluster at the center of the interconnection region, similar to that illustrated in Fig. 15. This is a disadvantage compared to the 2-D to 1-D array example, where the

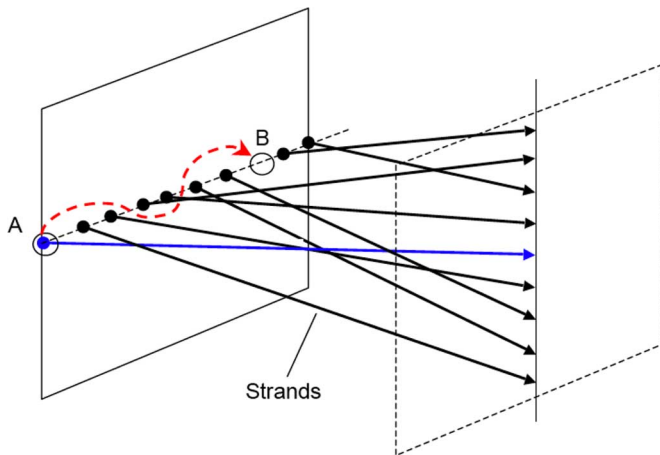


Fig. 19. Reconfiguration path for 1-D input and intermediate arrays.

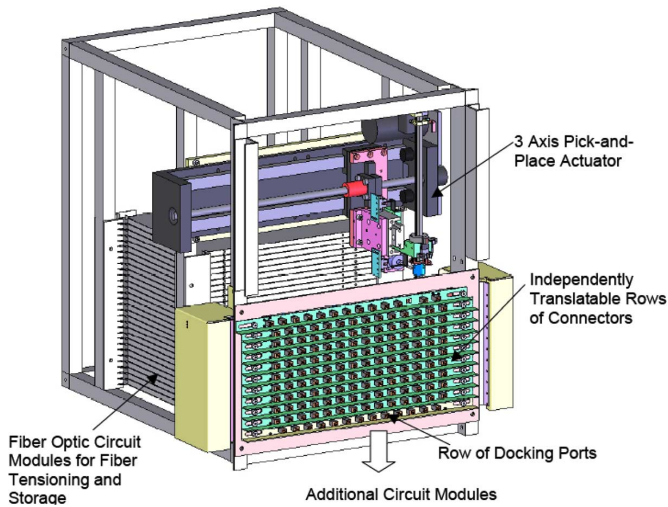


Fig. 20. Drawing of the  $140 \times 140$  port optical cross-connect switch prototype.

crossover clusters are more evenly distributed. The time to reconfigure a strand in this example is proportional to  $N \cdot dx$ .

Terminals can be added incrementally by concatenating interconnects at the ends of the array. However, since the minimum spacing between each strand in the input array is typically 5 mm to allow adequate space for a connector body, then within the footprint of manual patch—panels, this approach is practically limited to less than about 400 ports and the port density is significantly lower than the previous 2-D to 1-D example.

### III. EXPERIMENTAL RESULTS

We have implemented the reconfiguration algorithms and system architecture of the optimal 2-D to 1-D all-fiber cross-connect system in a fully automatic prototype with 140 input and 140 output fibers and full software control. In this demonstration system, the input terminals are MU simplex adapters and the flexible circuits utilize single-mode bend-insensitive fiber. To manage bend radius while the strands are tensioned, each strand consists of an optical fiber in combination with a stainless steel stiffening wire. The outer dimensions of the cross-connect unit are 58 cm wide by 61 cm tall.

Reconfiguration at the input terminal array is achieved by a robotic system utilizing a three-axis pick-and-place system with a gripper that clips onto the internal MU connectors to remove or insert them at the front panel array. Typical fiber optic connectors include internal structures to guide the connector into the mating receptacle during insertion, so positioning tolerances during strand reconfigurations are not precise (about  $\pm 1$  mm). Independent tensioning and buffering of each flexible circuit is achieved by flexible circuit modules located to the rear of the intermediate port array.

Fig. 20 illustrates a three-dimensional model of this cross-connect system, with the fiber strands not shown for clarity. Note that the fiber strands attached to the front input terminal array originate from a one-dimensional array interior to the cross-connect, where the fiber tensioning and buffering modules also reside.

Insertion loss of this system is identical to a standard patch-panel, 0.25 dB typical, and the return loss is consistent with UPC polished single mode connectors ( $\sim 50$  dB). During the  $\sim 1$  minute reconfiguration time, the robotic unit consumes about 25 W on average. No electrical power is required between reconfigurations. Scaling of the cross-connect to higher port counts is readily achieved by vertically stacking additional circuit modules above or below existing modules.

In practice, a clean fiber endface is required to achieve repeatable, low loss connections. In addition to external contaminants, typical connector housings can shed particulates that can deposit on the fiber endface. A process of cleaning the endface between reconfigurations, for example, by contacting with a moving ribbon of cleaning fabric, is necessary.

### IV. CONCLUSION

The reconfigurable, all-fiber interconnections described in this paper exhibit ideal optical performance, scalability and modularity. The system of fiber strands linking a 2-D input array to a 1-D intermediate array provides optimal functionality from among the various interconnect geometries. We have developed an automated cross-connect prototype based on this architecture that demonstrates reconfigurability in a completely non-blocking fashion using Braid Theory-based algorithms. The use of all-fiber interconnections makes this approach broadly applicable to all optical fiber types, including single mode and bend-insensitive fiber, multimode fiber, holey fiber, plastic optical fiber, and high power handling, large mode area fiber. Moreover, the favorable scaling characteristics make this cross-connect architecture ideal for both small scale ( $< 100$  input and output fibers) and large scale ( $> 1000$  input and output fibers) optical switches.

### ACKNOWLEDGMENT

The author thanks Prof. A. Yariv and P. Hudson for their insightful comments and valuable contributions.

### REFERENCES

- [1] T. J. Xia and G. Wellbrock, NXCComm 2008, Thur. 11:20 am-12:15 pm Session. Las Vegas, NV, June 19, 2008.
- [2] S. Sjolinder, "Mechanical optical fibre cross connect," in *Proc. Photon. Switching, Paper PFA4*, Salt Lake City, UT, Mar. 1995.

- [3] M. Mizukami, M. Makihara, S. Imagaki, and K. Sasakura, "200 × 200 automated optical fiber cross-connect equipment using a fiber-handling robot for optical cabling systems," in *Optical Fiber Communications Conf. 2005*, Anaheim, CA, 2005, Paper OFP5.
- [4] K. Goossen, "Robotic Optical Cross-Connect," U.S. Patent No. 6 307 983, Oct. 23, 2001.
- [5] K. Saito, M. Nishimura, T. Yamanishi, H. Koboyashi, T. Katagiri, and M. Tachikura, "Optical Fiber Switching Device Having One of a Robot Mechanism and an Optical Fiber Length Adjustment Unit," U.S. Patent No. 5 613 021, Mar. 18, 1995.
- [6] N. Tamaru, Y. Nishida, T. Kanai, J. Yamaguchi, and T. Shoji, "Optical Fiber Cross Connection Apparatus and Method," U.S. Patent No. 5 784 515, Jul. 21, 1998.
- [7] J. Arol and Z. Ganor, "Self Aligning Opto-Mechanical Crossbar Switch," U.S. Patent No. 6 859 575, Feb. 22, 2005.
- [8] D. A. Epstein, *Word Processing in Groups*. New York: Jones and Bartlett, 1992, ch. 9.
- [9] S. Moran, *The Mathematical Theory of Knots and Braids an Introduction*. The Netherlands: North Holland, 1983.
- [10] A. S. Kewitsch, "Fiber Optic Rotary Coupling and Devices," U.S. Patent No. 7 315 681, Jan. 1, 2007.

**Anthony S. Kewitsch** (M'92) received the B.S. degree with Distinction in electrical engineering from Stanford University, Stanford, CA, in 1991, where he was a Kodak Fellow, and the M.S. and Ph.D. degree in applied physics from the California Institute of Technology, Pasadena, in 1993 and 1995, where he was an NSF, ONR, Tau Beta Pi and Caltech Special Institute Fellow.

He is presently President/CEO of Telescent Inc. in Santa Monica, CA, a developer of automated fiber optic cross-connect systems. Current research interests include optical switching, network management, optical network design and microwave photonics.

Dr. Kewitsch is a member of IEEE, OSA, and the Advanced Fiber Connectivity and Switching (AFCS) Forum.