Three Dimensional Free Space Optical Network Reconfiguration Heuristics Using Active Link Removal

Xuemei Sun, Danny Luong, and G. Young
Computer Science Department, California State Polytechnic University, Pomona, CA 91768

Abstract

Free Space Optical (FSO) communication is a type of optical communication that relying on beams of light to transmit data over long distances wirelessly. With recent increase in popularity of the technology in commercial applications as well as military and scientific applications, it is important to study new models and algorithms to improve the performance of this technology. One of the most important aspects of FSO network is the line-of-sight (LOS) requirement to sustain an uninterrupted data flow. In order to ensure a steady connection, auto-aligning transceivers can be implemented to overcome the LOS issue. In this paper, we propose several re-alignment algorithms to reconfigure a FSO network when there is a loss of line of sight. The main idea of our experiment is to expand up on the three dimensional FSO network model proposed by Kosumo et al. in 2013, addressing cases that have not been considered.

Keywords: 3D Mesh Network, Routing, Reconfigurable Network, Free Space Optics, Wireless Network.

1. Introduction

Being a wireless technology, FSO has many advantages over its wired counterpart, fiber optic communication. Similar to fiber optic communication, FSO uses light sources and detectors to send and receive data. Nonetheless, instead of relying on extensive networks of fiber cables for transmission, FSO send data directly through the air, hence the name free-space. The main purpose for using FSO instead of fiber cable is to lower or even eliminate the costs of cable installation. FSO technology also does not require radio frequency spectrum licensing. This advantage allows FSO to be used more freely. Furthermore, it is possible to relocate an existing FSO network elsewhere, allowing the recycle of equipment. FSO technology may be the next prominent broadband network. High speed data rates, unlicensed spectrum, excellent security, low setup time, and inexpensive infrastructure are among its most attractive features [24].

Over the last few decades, FSO technology has been studied extensively. Different models and performance measurements have been proposed. There have also been many experiments done on this technology [7, 11, 14, 16-18, 20, 21, 26]. In 2013, Kosumo et al. proposed a three dimensional (3D) model for FSO network. In their paper, several different heuristic algorithms for link reconfiguration were introduced and discussed [12]. For the sake of convenience, we will refer to their model and heuristics as KLWY (Kosumo, Luong, Wong, and Young) throughout our paper.

KLWY’s 3D FSO network model provides a convenient way to study the effect that random broken links have on an auto-reconfigured FSO network over time. In their experiment, links between nodes in a simulated network are allowed to be randomly disconnected over time while the heuristic algorithms try to keep the network functional by reconnecting the nodes that have broken links. The experiment gives insight into how a network will transform after a period of time. It also shows that certain heuristics have better performance and are more effective than others in keeping the network connected and functional. However, the proposed heuristics do not consider the possibility of actively removing existing links in order to form new links [12].

In this paper, we propose several new link reconfiguration heuristics for 3D FSO networks. Our heuristics include the possibility of removing current working links in the network to form new links. We present analytical discussion and simulation results to determine the performance of the proposed heuristics. The overall performance of the heuristics is evaluated in terms of average node distance and network diameter.

The rest of the paper is organized as follows. Section 2 briefly discusses FSO technology and its different applications. Section 3 gives an overview of KLWY’s 3D FSO network model, heuristics, and experimental result. The detail of our work is described in section 4, and discussion about future works is covered in section 5.

2. FSO Technology and Applications

Telecommunication technology has been evolving dramatically over the last few decades. Both the volume and the speed of information exchange have increased greatly. Current trends in multimedia communications such as voice, video, data, and images, are creating a demand for flexible networks with extremely high capacities [19]. Optical fiber with its enormous potential has established the ability to satisfy this demand. Telecommunications companies have been increasing the reach of their fiber optic networks to their customers. Besides being highly reliable, optical fiber has unlimited growth potential. It offers a transmission medium with Terabit per second (Tbps) bandwidth. Nonetheless, even with this potential, optical fiber has been costly in
installations. Building infrastructure for an optical fiber network requires laying underground cable, which is usually very costly and time consuming. FSO technology has emerged because of this reason.

FSO uses beams of light to provide optical connections that can transmit images, videos, voice, and data. FSO has found applications in many different industries from commercial to scientific and military. For private corporate networks, wireless optics systems eliminate the recurring cost of leased lines while still providing a very high bandwidth link between sites. For temporary network connectivity needs, such as at exhibitions, conventions, sporting events, or disaster recovery, high bandwidth links can be easily and quickly provided using portable FSO systems. Furthermore, wireless optics systems can also be used as high-speed wireless backup for fiber optic cable and as "Last Mile" solutions, connecting customer sites to fiber backbones [2, 19].

Recent progress in space communication technology has proven the undisputable future of FSO. The Lunar Laser Communication Demonstration (LLCD) mission conducted by NASA in 2013 has revealed the possibility of expanding broadband capabilities in space using laser communication. LLCD demonstrated a record-breaking data download and upload speed to the moon at 622 Megabit per second (Mbps) and 20 Mbps respectively. NASA’s next mission for laser communication in space will be the Laser Communications Relay Demonstration (LRCD). LRCD is expected to demonstrate the ability to relay data at the rate of over one billion bits per second between two Earth stations using a satellite in geosynchronous orbit [1].

Despite the strengths and progress mentioned previously, FSO also has some weaknesses. FSO is essentially a LOS technology using free space as its medium. Because of the nature of its transmission medium, an FSO connection has potential disturbances such as rain, fog, physical obstructions, scintillation, beam wander, building’s movement, and seismic activities [7, 20, 26, 23]. In space communication and other long distance connections, the challenge involves pointing a very narrow laser beam accurately to the receiving device and keeping the LOS free of any physical obstruction [13]. When there is a physical obstruction in the path of the laser beam, the connection is completely lost.

3. Related Works

In 2013, Kosumo et al. proposed an approach to address link failures in a reconfigurable 3D FSO network, where transceivers are capable of realignment to create new connections with other transceivers in the network. This capability allows the network to be much more flexible. When there is a link failure in the network, transceivers can realign themselves to form new connections, keeping the network functional and connected. Following is an overview of KLWY’s 3D FSO network model and their reconfiguration heuristics.

3.1. KLWY’s 3D FSO Network Model

KLWY’s model is designed based on the well-studied mesh network topology model [5, 6, 10]. It is assumed that the network is made up of different nodes that are linked together. All the nodes are FSO devices. Thus the network model is classified as a homogeneous n x n x n mesh network topology. Each node in the network is limited to 6 connections, i.e. each node can connect up to 6 other nodes in the network.

![Figure 3.1](image1.png)

Figure 3.1 a node can have up to 6 connections [12]

The reconfiguration strategy for 2D n x n mesh networks was studied by Lee and Young in 2004 [17, 18]. In order to study the proposed 3D model, Kosumo et al. divided possible links into three different categories, type I, II, and III links. Type I links connect all the transceivers in the 3D FSO network before any reconfiguration take place. A node can connect with other node one hop away on any one of its axes through type I link. There can be up to 6 different type I links for a single node.

![Figure 3.2](image2.png)

Figure 3.2 Type I link [12]

Type II link can be formed by connecting two diagonal nodes on a plane. In other words, a type II link connects a node with another node that is one hop away on two of its axes. A node has 12 different possible type II links.

![Figure 3.3](image3.png)

Figure 3.3 Type II link [12]

The last type is type III link. Type III link is a resulting diagonal link formed by two nodes that are located on different planes diagonally. Type III link connects a node with another node that is one hop away on all of its three axes. A node has 8 different possible type III links.
3.2. KLWY’s Heuristics and Experimental Results

3.2.1. Link Reconfiguration Heuristics

Kosumo et al. offered 7 different possible heuristics to reconfigure their network after a link failure. Based on the model described in the previous section, when there is a link failure, a certain type of link is reformed based on the heuristic algorithm used. Every time a link is broken, there are two more nodes that have less than the maximum number of connection allowed. The heuristics then will try to reconnect either one or both of these nodes to other free nodes in the network. The 7 reconfiguration heuristics are summarized below.

- **H0**: No reconnection after link failures for both nodes
- **H1**: Attempt to reconnect only one node with a type II link
- **H2**: Attempt to reconnect each node with a type II link
- **H3**: Attempt to reconnect only one node with a type III link
- **H4**: Attempt to reconnect each node with a type III link
- **H5**: Attempt to reconnect only one node with a type II or type III link
- **H6**: Attempt to reconnect each node with a type II or type III link
- **H7**: Attempt to reconnect one node with a Type II link, and the other node with a type III link

For link reconfiguration using one node (H1, H3, and H5), the procedure checks all the neighboring nodes of the first node of the pair with broken link. If there is a diagonal node with less than maximum number of connections, then a new diagonal link is established. If there is no neighboring node with free connection, then the procedure checks the second node of the pair with broken link to see if any of its neighbors have less than maximum number of connections. Again if there is a node with free connection, a new link is established [12].

For link reconfiguration using two nodes (H2, H4, H6, and H7), both nodes of the pair with broken link are checked. If any of their neighbors have less than maximum number of connections, a new diagonal link is formed [12].

3.2.2. Experimental Results

Kosumo et al. measured the performance of their heuristics based on the network diameter and the average node distance of the network. Data gather from their experiment are tabulated in the following graphs.

The main conclusion that they drew is both the average node distance and network diameter decrease over time as more diagonal links are introduced into the network to replace the broken links. However, the type of diagonal link used to reconnect free nodes does provide different outcomes. As we can observe from the graphs, heuristics H1 and H2, which use type II links only, outperformed the other heuristics. H7, which uses both type of links at the same time, had the best performance. H7 was able to keep the whole network connected while decreasing the average node distance and network diameter as the number of link failures increased. Heuristics using type III link, on the other hand, did not perform well. H3 and H4 managed to decrease both the average node distance and the network diameter, but they allowed the network to become disconnected very early. In summary, the factors that contribute to the difference in performance between heuristics are the types of link used to reform broken links and the degree of connectivity of each node [12].
4. Reconfiguration Heuristics Using Active Link Removal

In this paper, we extend the work of Kosumo et al., which was done last year [12]. They introduced the possibility of reconfiguring FSO network after links failures, focused on a 3 dimensional \( n \times n \times n \) mesh network. Our research utilizes the same 3D FSO network model but focus on active link removal. Through the study of different shortest path algorithms and reconfigurable network models [4-6, 8, 9, 15, 22, 25], we come up with a different set of heuristics that take into account the possibility of removing existing links that are still working in the network to form new links.

4.1. Reconfiguration Heuristics

First, we introduce a notation system for our heuristic algorithms to it more convenience referring to them.

- \( p_i \): Reconfiguration using pattern \( i \) \((1 \leq i \leq 6)\)
- \( N \): No removal of any existing links, use free transceivers only.
- \( R \): Remove an existing link for reconfiguration of certain pattern.

Based on this notation, the definition and description of the heuristic algorithms are given:

- \( H_0 \): No reconnection after link failures.
- \( H_{Np1}, H_{Np2}, H_{Np3}, H_{Np4}, H_{Np5}, H_{Np6} \): No removal of any existing links, using Pattern 1, 2, 3, 4, 5, 6 for reconfiguration.
- \( H_{Rp1}, H_{Rp2}, H_{Rp3}, H_{Rp4}, H_{Rp5}, H_{Rp6} \): Removing an existing link if necessary, using Pattern 1, 2, 3, 4, 5, 6 for reconfiguration.

These 6 different patterns show which link is removed based on the heuristics we use. In order to guarantee that both black nodes will be reconnected via a type I, II, or III link, we use these patterns to pick a suitable link for removal.

Heuristic \( H_0 \) is the simplest among all above algorithms, and it also offers the worst results. Since \( H_0 \) does nothing after link failures, the network very quickly becomes disconnected after a certain number of link failures. The only reason it is included in our study is because we can use it as a control group to compare the results of other reconfiguration algorithms.

We believe that with stronger constraints on the link patterns, better performance can be achieved. An important part of our study is to better understand and compare the impact of the proposed heuristic algorithms through real data. In our experiment, a multi-threaded simulation harness is designed and implemented. In the simulation a relatively large network \((n = 10)\) is used and a given number of links are randomly chosen in the network to fail. All the algorithms described previously are implemented and applied to the network. The resultant average node distance and network diameter for each algorithm are recorded and compared.

4.2. Simulation Environment and Parameters

4.2.1. Simulation Environment

- OS: Window 7 Ultimate
- Processor: Intel Core i5-3210M CPU @ 2.5GHz
- Programming language: Java
- IDE: Eclipse Java EE IDE, Juno Release
- Threads: Multi-threaded

4.2.2. Simulation parameters:

- Mesh size \((n = 10)\): \(10 \times 10 \times 10\)
- Number of nodes \( V = n^3; 1000 \)
- Initial number of links \( E = 3n^2(n - 1) \)
Statistical interval: 200 links failures
Network performance parameters: average node distance and network diameter

4.3. Simulation Results

The average node distance matrix for all algorithms and various numbers of failed links is shown in Table 4.1 and the network diameter in Table 4.2. The algorithm H7 in [12] and the baseline without any reconfiguration are also run for the sake of comparison. The overall best algorithms from the two groups of proposed algorithms are portrayed side by side with H0 and H7 in Figure 4.5 and Figure 4.6.

From Table 4.2, it can be observed that with increasing number of failed links, the network soon become unconnected, whereas several algorithms can maintain the connectivity throughout the whole simulation. In Table 4.1 and Table 4.2, we notice some non-removal algorithms. Although reduce the average node distance with a small fraction of broken links, these algorithms tend to lead to the network being disconnected when the number of link failures is higher. On the contrary, the removal-based algorithms, even with the same patterns, can defer the appearance of disconnection in the network.

The algorithms HNp1 and HRp1 are selected for their overall better performance and drawn against H0 and H7 in Figure 4.5 and Figure 4.6. In the figures, it is easy to observe the intersection of HNp1 and HRp1. In other words, it shows that as we have predicted, the removal based algorithms perform better when the link failures are relatively sparse. This can also be verified with the data of the other algorithms in the tables. In the same figures, it is also shown that the proposed algorithms outperform the existing algorithm H7 in terms of the average node distance.

<table>
<thead>
<tr>
<th>Failure</th>
<th>$H_0$</th>
<th>$H_7$</th>
<th>H_Np1</th>
<th>H_Np2</th>
<th>H_Np3</th>
<th>H_Np4</th>
<th>H_Np5</th>
<th>H_Np6</th>
<th>H_Rp1</th>
<th>H_Rp2</th>
<th>H_Rp3</th>
<th>H_Rp4</th>
<th>H_Rp6</th>
<th>H_Rp6</th>
</tr>
</thead>
</table>

Table 4.1 Comparison of Average Node Distances under Link Failures for Different Heuristics

<table>
<thead>
<tr>
<th>Failure</th>
<th>$H_0$</th>
<th>$H_7$</th>
<th>H_Np1</th>
<th>H_Np2</th>
<th>H_Np3</th>
<th>H_Np4</th>
<th>H_Np5</th>
<th>H_Np6</th>
<th>H_Rp1</th>
<th>H_Rp2</th>
<th>H_Rp3</th>
<th>H_Rp4</th>
<th>H_Rp6</th>
<th>H_Rp6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1900</td>
<td>20.828</td>
<td>20.799</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>23.213</td>
<td>24.042</td>
<td>--</td>
<td>22.385</td>
<td>23.517</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2100</td>
<td>19.924</td>
<td>21.971</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>22.799</td>
<td>23.799</td>
<td>--</td>
<td>22.213</td>
<td>--</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2300</td>
<td>20.364</td>
<td>21.799</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>22.213</td>
<td>23.627</td>
<td>--</td>
<td>22.799</td>
<td>--</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2500</td>
<td>20.560</td>
<td>20.385</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>22.627</td>
<td>25.042</td>
<td>--</td>
<td>23.627</td>
<td>--</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2700</td>
<td>20.292</td>
<td>20.799</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>23.627</td>
<td>--</td>
<td>--</td>
<td>23.627</td>
<td>--</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2 Comparison of Network Diameters under Link Failures for Different Heuristics
rose link failures data may not readily available to a network. The configuration algorithm for the network. This group of algorithm for the estimation of the number of link failures. Existing patterns can be used to form new link. This group of algorithms does not remove existing links even when it is necessary in order to form new links, i.e., there are not enough free nodes to actually form new connections. The other group of reconfiguration algorithms, however, is more aggressive in forming new links for the network. This group of algorithm actively removes existing link in the network in order to form new links and maximize the total number of connection in the network. A simulated environment was used to test the effectiveness of the proposed algorithms.

We collected the simulated results and computed the impact of different link reconfigurations on the overall network performance in terms of average node distance and network diameter. When the link breakage is expected to be sparse in the network, the removal-based algorithms in general work better than the non-removal ones. From the simulation result, HRp1 should be adopted in the long run, since it has the best over all performance. However, when the ratio of link failures exceeds a critical point, non-removal algorithm such as HNp1 might work better. In practice, it is FSO networks may operate on a large scale, and due to its distributed nature, the global link failures data may not readily available to a central control module. Thus it is not possible to pick the optimal algorithm for link reconfiguration. In this case, it would be desirable to first adopt a proper algorithm based on estimation of network variance. Perhaps when the network is in operation, local nodes can maintain a history of link failures. This data can then be aggregated and analyzed for a probability distribution function of number of link failures over time. The most suitable reconfiguration algorithm for the specific network condition can then be decided by using the knowledge from our simulated results.

Further study can be done on this topic. Under our current assumption of the network model, it is possible to add new patterns to the link removal algorithms. Existing patterns can also be combined, possibly forming a sequence of preferential patterns based on the number of free/available neighbor nodes. Instead of measuring performance of different heuristics using average node distance and network diameter, it is also feasible to use another parameter. Our current simulation model does not take into account the cost of each link in the network. All links in our network have their costs associated with the achievable 3D FSO network. The average node distance is a measure of connectivity and network diameter is a measure of scalability. A lower average node distance indicates a more connected network, while a smaller network diameter indicates a more scalable network. In this study, we proposed two groups of 3D FSO network reconfiguration algorithms based on patterns. One group of algorithms is opportunistic in a sense that they try to establish the predefined link pattern only if there are free nodes, which are available to form new link. This group of algorithms does not remove existing links even when it is necessary in order to form new links, i.e., there are not enough free nodes to actually form new connections. The other group of reconfiguration algorithms, however, is more aggressive in forming new links for the network. This group of algorithm actively removes existing link in the network in order to form new links and maximize the total number of connection in the network. A simulated environment was used to test the effectiveness of the proposed algorithms.

We collected the simulated results and computed the impact of different link reconfigurations on the overall network performance in terms of average node distance and network diameter. When the link breakage is expected to be sparse in the network, the removal-based algorithms in general work better than the non-removal ones. From the simulation result, HRp1 should be adopted in the long run, since it has the best over all performance. However, when the ratio of link failures exceeds a critical point, non-removal algorithm such as HNp1 might work better. In practice, it is FSO networks may operate on a large scale, and due to its distributed nature, the global link failures data may not readily available to a central control module. Thus it is not possible to pick the optimal algorithm for link reconfiguration. In this case, it would be desirable to first adopt a proper algorithm based on estimation of network variance. Perhaps when the network is in operation, local nodes can maintain a history of link failures. This data can then be aggregated and analyzed for a probability distribution function of number of link failures over time. The most suitable reconfiguration algorithm for the specific network condition can then be decided by using the knowledge from our simulated results.

Further study can be done on this topic. Under our current assumption of the network model, it is possible to add new patterns to the link removal algorithms. Existing patterns can also be combined, possibly forming a sequence of preferential patterns based on the number of free/available neighbor nodes. Instead of measuring performance of different heuristics using average node distance and network diameter, it is also feasible to use another parameter. Our current simulation model does not take into account the cost of each link in the network. All links in our network have their costs associated with the
distances between nodes. Different costs for each link might be implemented in the model, and another performance measurement might be used.

Another potential direction for future researches is to analyze the performance intersection point of non-removal algorithms and removal-based algorithms. This intersection point, if can be found, may be independent of the network size (i.e. not a function of n) and can serve as an indication of sparseness of the link failure of the network. It is not only useful for selecting reconfiguration algorithms but can also provide directions on the design of the network topology in the first place. The case where links in the network fail progressively can also be investigated. If a global reconfiguration can be performed in a network, more comprehensive algorithms should be available to attain overall optimal performance. The ultimate goal would be to achieve some theoretical analysis of the network topology change under link failures and various reconfiguration algorithms. The analysis can be started on smaller networks and then be scaled up by composition.

REFERENCES