Optical Switching and Networking 11 (2014) 29–43

Contents lists available at ScienceDirect

Optical Switching and Networking

journal homepage: www.elsevier.com/locate/osn

On the design and optimization of a free space optical access network ☆

I.K. Son [a], Shiwen Mao [b,*], Sajal K. Das [c]

[a] *Defense Acquisition Program Administration (DAPA), Republic of Korea*
[b] *Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849-5201, USA*
[c] *Department of Computer Science and Engineering, University Texas at Arlington, Arlington, TX 76019, USA*

ARTICLE INFO

Article history:
Received 7 February 2013
Received in revised form
17 July 2013
Accepted 12 August 2013
Available online 5 September 2013

Keywords:
Algebraic connectivity
Clustering
Free space optical networks
Network planning
Wireless mesh networks

ABSTRACT

Although having high potential for broadband wireless access, wireless mesh networks are known to suffer from throughput and fairness problems, and are thus hard to scale to large size. To this end, hierarchical architectures provide a solution to this scalability problem. In this paper, we address the problem of design and optimization of a tiered wireless access network that exploits free space optical (FSO) communications. The lower tier consists of mesh routers that are clustered based on traffic demands and delay requirements. The cluster heads are equipped with wireless optical transceivers and form the upper tier FSO network. For topology design and optimization, we first present a *plane sweeping and clustering* (PSC) algorithm aiming to minimize the total number of clusters. PSC sweeps the network area and captures cluster members under delay and traffic load constraints. For the upper tier FSO network, we present an algebraic connectivity-based formulation for topology optimization. We then develop a *greedy edge-appending* (GEA) algorithm, as well as its distributed version, that iteratively inserts edges to maximize algebraic connectivity. The proposed algorithms are analyzed and evaluated via simulations, and are shown to be highly effective as compared to the performance bounds derived in this paper.

1. Introduction

The recent explosive increases in pervasive mobile devices and wireless data applications have greatly stressed the capacity of existing wireless access networks. The significant increase in wireless data volume will also have far-reaching impact on the design of future wireless access networks. To this end, wireless mesh networks (WMN) have emerged as a promising technology for providing ubiquitous broadband wireless access to mobile users [2]. Recent years have witnessed significant growth in WMN research and deployment. However, WMNs are also known to suffer from poor scalability. In [3], Jun and Sichitiu showed that the per node throughput decreases as $O(1/n)$, where n is the number of nodes. In [4], the authors demonstrated that starvation occurs even in the simple scenario where one-hop flows contend with two-hop flows for gateway access. This is largely due to the inefficiency and bi-stability of existing MAC protocols, as well as high penalty for multi-hop flows to re-capture system resources.

Historically, hierarchical network architectures have provided an effective solution to the scalability problem, as demonstrated in the Internet and wireless sensor networks [5]. In the case of WMNs, the authors in [6]

☆ Part of this work was conducted when In Keun Son was pursuing a doctoral degree at Auburn University. This work was presented in part at IEEE INFOCOM 2010, San Diego, CA, USA, March 2010 [1].

* Corresponding author. Tel.: +1 334 844 1845; fax: +1 334 844 1809.
E-mail addresses: soninkeun@gmail.com (I.K. Son), smao@ieee.org (S. Mao), das@cse.uta.edu (S.K. Das).

1573-4277/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.osn.2013.08.004

investigated a hybrid network architecture consisting of an underlying n-node wireless ad hoc network and a sparse overlay network of m base stations, which are connected with high-bandwidth wired links. The authors showed that the asymptotic throughput capacity of the hybrid network increases linearly with m if m grows faster than $\sqrt{n}$.

In this paper, we investigate the design and optimization of a tiered wireless access network, as illustrated in Fig. 1. The lower tier consists of a WMN with a large number of mesh routers providing wireless access to mobile users. To mitigate the scalability problem, we group mesh routers into clusters with bounded diameter. Traffic from/to the cluster members is aggregated and routed through the cluster head, which is equipped with free space optical (FSO) transceivers with multi-gigabit data rates and multi-kilometer ranges. With this architecture, resource contention mainly occurs within the cluster and end-to-end hop counts are greatly reduced due to the use of FSO links. Compared to the architecture in [6], FSO links are easier to deploy than wired links, and can be easily rearranged when traffic requirements change, or when links or nodes fail. The challenging task of QoS provisioning (e.g., delay and throughput) can be greatly simplified.

FSO communications provide cost-effective, license-free, and high-bandwidth links for the upper tier [7]. FSO links require line-of-sight (LOS) and are point-to-point connections. They are immune to electromagnetic interference and are secure due to point-to-point connection with narrow beam divergence. However, FSO links are subject to impairments in the open-air transmission medium, such as attenuation, atmospheric turbulence, obstacles, and beam misalignment. It is important to design topologies with rich connectivity to cope with transmission impairments.

We address the challenging problem of network planning for the tiered wireless access network. The objective is to jointly determine the optimal partition for the lower tier WMN as well as the optimal topology for the upper tier FSO network. However, such a topology design problem is highly complex due to its combinatorial nature. To make the problem tractable, we take a divide-and-conquer approach to break it down into two sub-problems. The first sub-problem is cluster formation in the underlying WMN. In the two-tier architecture, each cluster head serves as an FSO node in the upper tier and is equipped with FSO transceivers. The more the clusters, the more the FSO transceivers that are required in the upper tier. Further, it would be desirable to include more mesh nodes in a cluster (as long as the aggregate traffic condition is satisfied, see Section 3.1) to fully utilize the high data rates an FSO link can offer. Therefore, the objective is to minimize the number of clusters (i.e., cost), while satisfying the delay and traffic load requirements. The second sub-problem is topology design and optimization for the upper tier FSO network to achieve maximum connectivity for a given number of edges. The two sub-problems are coupled with the common objectives of minimizing the cost and maximizing the reliability of the tiered system, while satisfying the QoS requirements.

The formulated sub-problems belong to the class of NP-hard problems [8].[1] To provide effective solutions, we take a graph theoretic approach to develop heuristic algorithms. First, we develop a *plane sweeping and clustering* (PSC) algorithm that sweeps the network area and captures cluster members one after another under delay and traffic load constraints. PSC chooses cluster members by manipulating the adjacency matrix and hop-count matrix of the underlying graph. We derive a lower bound on the number of clusters, which can be used as a benchmark for performance evaluation, and investigate effective schemes to reduce the computational complexity of PSC. Second, we present an algebraic connectivity-based formulation for FSO network topology optimization. We then develop a *greedy edge-appending* (GEA) algorithm, as well as its distributed version, that iteratively inserts edges to maximize algebraic connectivity. The proposed algorithms are analyzed with regard to complexity and performance bounds, and evaluated via simulations. They are shown highly effective for solving the network design and optimization problem as compared to the performance bounds developed in this paper.

The rest of this paper is organized as follows. We present the system model in Section 2. The design and optimization problems are formulated in Section 3. We present PSC and GEA, and analyze their performance in Sections 4 and 5, respectively. Simulation results are presented in Section 6, and related work is discussed in Section 7. Section 8 concludes the paper.

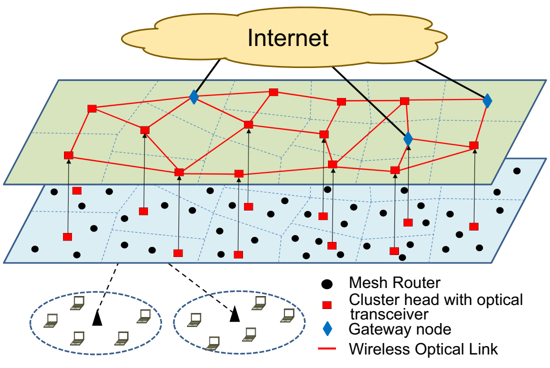


Fig. 1. Reference architecture for the tiered wireless access network.

2. System model

2.1. Tiered access network model

As shown in Fig. 1, the lower tier of the wireless access network is a WMN consisting of n mesh routers that provide access to mobile users [2]. The mesh routers use broadcast radio transceivers and each covers a small

[1] Aoun, et al. in [9] showed that their gateway placement problem is NP-hard in order to minimize the number of gateways, which is analogous to the number of clusters in the cluster formation subproblem. The FSO topology optimization problem is also NP-hard, since in [10] it was proven that maximizing algebraic connectivity with a given number of edges is NP-hard.

service area. Without loss of generality, we assume that the mesh routers form a connected network. A small fraction of the mesh routers are *gateway nodes* with connection to the Internet. Traffic to/from the Internet is routed via the gateways, while traffic between a pair of mesh routers is relayed through multi-hop routes. The traffic demand is characterized by a given *traffic matrix* **F**, where element $[\mathbf{F}]_{uv}$ is the required traffic load from mesh router u to v.

WMNs are known to suffer from throughput and fairness problems [4]. They are hard to scale to large sizes, resulting in the so-called “a-few-hop” wireless networks. Although increasing the link data rate helps, it is not practical to have FSO transceivers at each mesh router, due to the cost constraint and the mismatch on data rate and range between radio and FSO links. We form clusters in the WMN. The mesh router traffic is aggregated at the cluster heads, which are connected with high speed FSO links to provide fast lanes for aggregate traffic.

The upper tier network is designated to cover wide range with high data rate. FSO provides license-free and high-bandwidth links for this purpose. We assume that each cluster head can be equipped with multiple FSO transceivers for rich connectivity, and the links are full duplex. The link capacity from cluster head i to j is denoted by C_{ij}. Due to LOS and narrow beam divergence, we assume that $C_{ij}=C_{ji}$, for all $i \neq j$. The main design considerations for FSO networks are cost and reliability. Available FSO systems cost from $10,000 USD to $25,000 USD for medium to long link ranges [11], more expensive than mesh routers. Therefore it is important to minimize the number of FSO transceivers (or, links) to reduce the deployment cost. In addition, FSO links are also susceptible to atmospheric turbulence, unexpected tiny obstacles, and beam misalignment. It is important to achieve rich connectivity for the FSO topology.

We assume three operation modes for the tiered access network: (i) Mode I: inter-cluster or Internet traffic is aggregated at cluster heads and forwarded through FSO links; (ii) Mode II: when an FSO link is temporarily blocked (i.e., by a bird flying through the LOS path), a multipath routing protocol will redirect traffic through alternative paths in the upper tier FSO network; and (iii) Mode III: during severe weather conditions such as heavy snow, many FSO links may be blocked (like satellite links that could be blocked by a heavy thunder storm). The WMN will backoff to the normal mesh networking mode with radio communications and multi-hop relays. Experimental results in three U.K. cities from January 1975 to December 1983 show that atmospheric attenuation was constantly low over 99% of the time [12]. So the tiered access network will work in Modes I and II most of the time, exploiting the high-speed FSO links for high throughput/low delay performance.

2.2. FSO channel model

We assume that point-to-point wireless optical transceivers are used for the cluster heads in the outdoor field. Practical FSO communication systems usually adopt *intensity modulation with direct detection* (IM/DD) with *On-Off Keying* (OOK) in the PHY. There are several impairing factors through an LOS path, such as attenuation, atmospheric turbulence, unexpected tiny obstacles, and beam misalignment. In this paper, we consider attenuation and atmospheric turbulence as characterized by the *refractive-index structure parameter*, which affect the FSO channel at the relatively large timescale relevant to network planning.

Several irradiance distributions have been used for modeling a light beam propagating through turbulent air (i.e., the distribution of the received light intensity). The *log-normal distribution* is shown to be highly accurate for weak weather turbulences [13]. The marginal distribution[2] of received light intensity can be modeled as

$$f_I(I)=\frac{1}{2\sigma_X I}\frac{1}{\sqrt{2\pi}}\exp\left\{-\frac{(\ln(I)-\ln(I_0))^2}{8\sigma_X^2}\right\},$$

where I_0 is the average received intensity and σ_X^2 is the variance of the log-intensity fluctuation. The variance has the form

$$\sigma_X^2=0.30545\left(\frac{2\pi}{\lambda}\right)^{7/6}C_n^2(\eta)z^{11/6},$$

where λ is the wavelength, $C_n^2(\eta)$ is the *index of refraction structure parameter* with absolute altitude η, and z is the transmission distance.

The received optical intensity of fixed FSO links mainly depends on $C_n^2(\eta)$. For atmospheric channels near the ground, e.g., $\eta < 18.5$ m, C_n^2 ranges from $10^{-13}\ \mathrm{m}^{-2/3}$ to $10^{-17}\ \mathrm{m}^{-2/3}$ for strong to weak atmospheric turbulences, with typical value $10^{-15}\ \mathrm{m}^{-2/3}$ [13]. Under log-normal fading, the reliability of an FSO link can be computed as

$$\Gamma_{ij}=\Pr\{I \geq I_{th}\}=\frac{1}{2}-\frac{1}{2}\,\mathrm{erf}\left(\frac{\ln(I_{th})-\ln(I_0)}{2\sigma_X\sqrt{2}}\right), \tag{1}$$

where I_{th} is the threshold of received signal intensity. With a suitable threshold Γ_{th}, we define the *weight* of an FSO link as

$$\omega_{ij}=\begin{cases}\Gamma_{ij} & \text{if } \Gamma_{ij}\geq\Gamma_{th}\\ 0 & \text{otherwise.}\end{cases} \tag{2}$$

From (1), we can compute the probability that the received signal strength is larger than a threshold I_{th}. If this probability is lower than the threshold Γ_{th}, we regard this link as broken (with a weight 0) and do not consider it in the topology design.

Due to LOS and narrow beam divergence, we assume that the links are symmetric, i.e., $\Gamma_{ij}=\Gamma_{ji}$ for all $i \neq j$. The main factors affecting link weights are the transmission distance and atmospheric turbulence. If the nominal capacity for all FSO links is C_{opt}, the effective capacity of an FSO link is $C_{ij}=\omega_{ij}C_{opt}$ considering the transmission impairments. The upper tier FSO network, therefore, can be modeled as an undirected and weighted graph $G(V, E)$, where V represents the set of cluster heads and E the set of FSO links. Each link (i,j) has weight ω_{ij} for all $i \neq j$.

[2] In [13], marginal distribution is used to refer to the distribution of fading at a single point in space at a single instant in time.

3. Problem statement

In this section, we formulate the design and optimization problem for the tiered wireless access network. Such network planning problems are combinatorial in nature and often have large sizes. To make the problem tractable, we adopt a divide-and-conquer approach to consider the following two sub-problems: (i) the *optimized clustering problem*, on cluster formation in the underlying WMN; and (ii) the *topology optimization problem*, on designing the upper tier FSO network topology. These two sub-problems are integrated with the overall objectives of minimizing the cost and maximizing the reliability of the tiered wireless network, while satisfying the traffic and delay requirements.

3.1. Lower tier: optimized clustering problem

The first step is to partition the underlying wireless routers into clusters, while considering delay and traffic demand requirements. End-to-end delay consists of queueing delay, processing delay, and transmission (and retransmission) delay induced at each hop. For network planning purpose, we consider time averages of the delay components. This is because clustering is performed at relatively large timescales for which time averages matter. There is no need (or, it would be too costly) to adapt clusters in response to, say, queueing delay fluctuations at small timescales. As in prior work [9], we translate the delay requirement to a bound h_{max}, i.e., the maximum hop count of the shortest path from any mesh router to its cluster head.

Assume that the *traffic matrix* $\mathbf{F} \in \mathcal{R}^{n \times n}$ is given for the mesh network. When the clusters are formed, the aggregate *incoming* and *outgoing* traffic of cluster M_i, denoted by f_{o,M_i} and $f_{M_i,o}$ respectively, can be computed as

$$\begin{cases} f_{o,M_i} = \sum_{u=1, u \notin M_i}^{n} \sum_{v \in M_i} [\mathbf{F}]_{uv} \\ f_{M_i,o} = \sum_{v \in M_i} \sum_{u=1, u \notin M_i}^{n} [\mathbf{F}]_{vu}. \end{cases} \tag{3}$$

Operators may enforce a bound on traffic load, f_{max}, for a cluster, i.e.,

$$f_{M_i} = \max\{f_{o,M_i}, f_{M_i,o}\} \le f_{max}. \tag{4}$$

Assuming that the transmission ranges of mesh radios are identical disks with radius r. For network design purpose, we focus on average link quality over relatively large timescales and ignore fast variations of channel quality. The mesh network can be represented by a graph G, with vertices representing mesh routers and edges representing radio links among neighboring mesh routers. The *adjacency matrix* $\mathbf{A} \in \mathcal{R}^{n \times n}$ of graph G can be derived for given node locations and r. We have the following fact from graph theory.

Fact 1. *Let* **A** *be the adjacency matrix of graph* G*. The number of walks from vertex* u *to* v *in* G *with length* k *is* $[\mathbf{A}^k]_{uv}$ [14].

Recall that cluster diameters are bounded by h_{max}. So we only concern with the maximum hop count h_{max}, i.e., up to the (h_{max})-th power of **A**. From Fact 1, the hop count of the shortest path between two nodes u and v can be computed as

$$h_{uv} = \begin{cases} \min\{k\} & \text{if } [\mathbf{A}^k]_{uv} > 0, \quad k \in [1, \ldots, h_{max}] \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

The first line in (5) is for the case when nodes u and v are less than h_{max} hops away from each other; the second line is for the case when nodes u and v cannot belong to the same cluster. We can define a *hop-count matrix* for graph G, denoted as $\mathbf{H} \in \mathcal{R}^{n \times n}$, with element $[\mathbf{H}]_{uv} = h_{uv}$ as defined in (5), i.e., the hop count of the shortest path from node u to v, for $u, v \in [1, \ldots, n]$.

Let binary variable y_u, $u = 1, \ldots, n$, be the indicator whether the u-th mesh router is chosen to be a cluster head and let x_{uv} be the indicator whether mesh router u is associated with cluster head v. The clustering problem can be formulated as a 0–1 *integer linear programming* (ILP) problem as

$$\text{minimize}: \quad n_c = \sum_{u=1}^{n} y_u \tag{6}$$

$$\text{subject to}: \quad \sum_{v=1}^{n} x_{uv} = 1 \quad \text{for all } u \tag{7}$$

$$\sum_{v=1}^{n} (x_{uv} \cdot h_{uv}) \le h_{max} \quad \text{for all } u \tag{8}$$

$$f_{M_u} \le f_{max} \quad \text{for all } M_u \tag{9}$$

$$x_{uv} \in \{0, 1\}, \quad y_u \in \{0, 1\} \quad \text{for all } u, v. \tag{10}$$

The main objective is to minimize the number of clusters to reduce the cost of the FSO network, under delay and traffic load constraints. Constraint (7) requires that each mesh router is assigned to one cluster. Constraint (8) indicates that the hop counts from all cluster members to the cluster head is bounded by h_{max}. Constraint (9) represents the bound on the aggregated incoming and outgoing traffic for a cluster, which are functions of x_{uv} and y_u as given in (3).

Once the clusters are formed, the new traffic matrix for the inter-cluster traffic, denoted by $\mathbf{F}'$, can be derived as

$$[\mathbf{F}']_{ij} = \begin{cases} y_i y_j \sum_{u=1}^{n} \sum_{v=1}^{n} x_{ui} x_{vj} [\mathbf{F}]_{uv} & \text{if } i \ne j \\ 0 & \text{otherwise.} \end{cases}$$

The number of FSO transceivers at cluster head i, denoted as K_i, can be determined from $\mathbf{F}'$, as

$$K_i = \max\{k_{min}, \min\{k_i, k_{max}\}\}, \tag{11}$$

where $k_{min} \in [1, \ldots, k_{max}]$ is a predefined lower bound on node degrees, $k_i = \lceil f_{M_i} / (\Gamma_{th} \cdot C_{opt}) \rceil$, and $k_{max} = \lceil f_{max} / (\Gamma_{th} \cdot C_{opt}) \rceil$. In the following topology optimization problem, K_i will be the degree for cluster head i. As will be shown in (12), the connectivity of a graph G is upper bounded by its minimum degree $\delta(G)$. Here we enforce a minimum value k_{min} for $\delta(G)$ to avoid trivial cases with poor connectivity (e.g., stubs or trees). Once the degrees for the cluster heads are derived, we have the following facts from graph theory, which will be used in topology optimization.

Fact 2. *The sum of node degrees is equal to twice of the total number of edges, i.e.,* $\sum_{i=1}^{n} K_i = 2 \cdot |E|$ [15].

Fact 3. *The number of edges of a* simple graph (*i.e., loop-free and without multiple edges*) *is upper bounded by* $\binom{n}{2}$ [15].

3.2. Upper tier: topology optimization problem

The next step is to optimize the topology of the upper tier FSO backbone. As discussed, the upper tier FSO network can be modeled as a weighted simple and mesh graph $G(V, E)$ with $|V| = n_c$ vertices and $|E| = m$ edges. The problem is to find a simple graph with vertex set V that achieves maximum connectivity for robustness to atmosphere turbulence and link failures, while satisfying node degree constraints (11).

In the following, we first introduce algebraic connectivity and its properties, and then formulate the topology optimization problem.

3.2.1. Algebraic connectivity preliminaries

There are several graph connectivity measures, including *graph degree and diameter*, *k-vertex/edge connectivity* and *bisection connectivity* [14].[3] In this paper, we consider *algebraic connectivity* from *spectral graph theory*, based on the following considerations [15,17]: (i) algebraic connectivity is well-studied as a key component of spectral graph theory. It is amenable for mathematical modeling and effective algorithm design. Algebraic connectivity can also provide bounds on graph operations such as subgraph, factorization, Cartesian product, disjoint union, and complement of a graph. (ii) Algebraic connectivity has several interesting properties. It is closely related to graph invariants including diameter, minimum degree, and connectivity, thus being more helpful to understand network topology as compared to other measures.

The algebraic connectivity of a graph G is defined to be the second smallest eigenvalue of the *Laplacian matrix* $\mathbf{L}$ of graph G, denoted as $\lambda_2(\mathbf{L})$, where $\mathbf{L} = \mathbf{L}(G)$ [15]. The value of $\lambda_2(\mathbf{L})$ is a nice indication of graph connectivity with the following facts [18].

Fact 4. *A graph is* connected *if and only if* $\lambda_2(\mathbf{L}) > 0$.

Fact 5. *The number of zero eigenvalues of* $\mathbf{L}(G)$ *is equal to the number of disjoint components of* G.

Fact 6. *For two graphs* $G_1 = (V, E_1)$ *and* $G_2 = (V, E_2)$ *with* $E_1 \subseteq E_2$, *we have* $\lambda_2(\mathbf{L}_1(G_1)) \le \lambda_2(\mathbf{L}_2(G_2))$.

According to the facts, a graph with $\lambda_2 = 0$ is partitioned, while a graph with $\lambda_2 > 0$ is connected with one component. The algebraic connectivity is non-decreasing as more edges are inserted to the graph. Furthermore, algebraic connectivity is closely related to other connectivity measures. For a connected graph G that is not *complete*, let $K_v(G)$ be the vertex connectivity and $K_e(G)$ the edge connectivity. The following inequalities hold true [18,17]:

$$\frac{4}{D \cdot |V(G)|} \le \lambda_2(\mathbf{L}) \le K_v(G) \le K_e(G) \le \delta(G), \tag{12}$$

where $\delta(G)$ is the minimum degree of the graph, D is the *graph diameter*, and $\mathbf{L}$ is the Laplacian matrix of graph G. The inequalities are illustrated in Fig. 2. It is interesting to see that graphs with the same number of edges can have different λ_2 values, as the second and third graphs in Fig. 2. For a given number of links, it is thus desirable to design a topology that maximizes the algebraic connectivity.

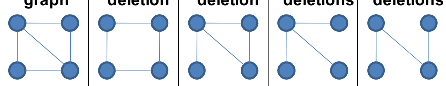
Topology	Origin graph	One edge deletion	One edge deletion	Two edge deletions	Two edge deletions
					
Min degree	2	2	1	1	1
Diameter	2	2	2	2	3
4/(D·|V|)	0.5	0.5	0.5	0.5	0.3333
Algebraic connectivity	2	2	1	1	0.5858

Fig. 2. Algebraic connectivity examples when edges are deleted from an unweighted graph.

3.2.2. Topology design and optimization problem

As a result of the clustering problem, the n_c cluster heads form an upper tier FSO network, which is modeled as a weighted graph $G(V, E)$. Each vertex $v_i \in V$ represents a cluster head with degree K_i, given in (11) as determined by the delay and traffic load requirements. The number of edges is $m = |E| = (1/2)\sum_{i=1}^{n_c} K_i$. Let the set of potential edges be E_{pot} (i.e., set of edges with positive weights). Recall that the link weights are calculated as in (1) and (2), where low quality links are assigned with weight zero. To ensure a feasible solution, a basic condition on E_{pot} is that the network is connected when all the links in E_{pot} is used, which can be achieved by adjusting the thresholds in (1) and (2). We have $E \subseteq E_{pot}$ and $|E_{pot}| = m_{pot}$. The topology design problem is to choose E from E_{pot} such that (i) the degree of each vertex is satisfied, and (ii) the algebraic connectivity is maximized.

We follow the Laplacian matrix definition in [19]. Assume that edge l connects two distinct vertices v_i and v_j, $l \in [1, \ldots, m]$, $v_i, v_j \in V$. We also denote edge l as (i, j) when there is need to distinguish the endpoints. We define $\mathbf{a}_l \in \mathcal{R}^{n_c}$ of the unweighted *incidence matrix* $\mathbf{\Lambda} \in \mathcal{R}^{n_c \times m}$ of graph G as

$$[\mathbf{a}_l]_k = \begin{cases} +1 & \text{if } k = i \text{ and } \omega_{ij} > 0 \\ -1 & \text{if } k = j \text{ and } \omega_{ij} > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

The $n_c \times n_c$ *weighted Laplacian matrix* $\mathbf{L}$ of graph G is

$$\mathbf{L}(G) = \sum_{l=1}^{m} \omega_l \cdot \mathbf{a}_l \cdot \mathbf{a}_l^T = \mathbf{\Lambda} \cdot \mathrm{diag}(\boldsymbol{\omega}) \cdot \mathbf{\Lambda}^T, \tag{14}$$

where diag $(\boldsymbol{\omega}) \in \mathcal{R}^{m \times m}$ is a diagonal matrix with diagonal elements ω_l, $l = 1, \ldots, m$. It follows from the definition that $\mathbf{L}(G)$ is a *positive semi-definite matrix*, i.e., $\mathbf{L}(G) \succcurlyeq 0$. Its

[3] Vertex (edge) connectivity is the minimum vertex (edge) cut for nonadjacent vertices i and j. Bisection connectivity is the likelihood that the graph would be separated into two components [16].

smallest eigenvalue, $\lambda_1(\mathbf{L})$, is zero with eigenvector $\mathbf{1} = [1,\ 1, \ldots, 1]^T$. The eigenvalues also satisfy the following inequality condition: $0 = \lambda_1 \le \lambda_2 \le \lambda_3 \le \cdots \le \lambda_{n_c}$. Since we consider connected graphs, it follows from Facts 4 and 5 that there is only one zero eigenvalue. Therefore we have

$$0 = \lambda_1 < \lambda_2 \le \lambda_3 \le \cdots \le \lambda_{n_c}. \tag{15}$$

The topology optimization problem for the upper tier FSO network can be formulated as follows[4]:

$$\text{maximize}: \quad \lambda_2(\mathbf{L}) \tag{16}$$

$$\text{subject to}: \quad m = \frac{1}{2}\sum_{i=1}^{n_c} K_i \le \binom{n_c}{2} \tag{17}$$

$$d_i = K_i \quad \text{for all } i \tag{18}$$

$$E \subseteq E_{pot}, \tag{19}$$

where d_i is the non-weighted degree of node i. To optimize the topology, we populate edge set E by selecting edges from E_{pot}, such that λ_2 of the resulting graph G is maximized and the delay and traffic load requirements are satisfied. The first constraint is from Facts 2 and 3. The second constraint is on the degree of each cluster head, such that the incoming and outgoing traffic from that cluster can be served.

When the problem is solved, each potential link $l \in E_{pot}$ will either be chosen to be included in E or not chosen. Define binary variables x_l, for all $l \in E_{pot}$, as

$$x_l = \begin{cases} 1 & \text{if potential edge } l \text{ is included in } E \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

Then we have a boolean vector $\mathbf{x} \in \{1, 0\}^{m_{pot}}$ for E_{pot}. The topology optimization problem can be reformulated into a 0–1 ILP problem as

$$\text{maximize}: \quad \lambda_2\left(\sum_{l=1}^{m_{pot}} x_l \cdot w_l \cdot \mathbf{a}_l \cdot \mathbf{a}_l^T\right) \tag{21}$$

$$\text{subject to}: \quad \mathbf{1}^T \cdot \mathbf{x} = \frac{1}{2}\sum_{i=1}^{n} K_i \le \binom{n_c}{2} \tag{22}$$

$$d_i = K_i \quad \text{for all } i \tag{23}$$

$$\mathbf{x} \in \{0, 1\}^{m_{pot}}, \quad \omega_l = \omega_{ij}, \quad l \sim (i,j). \tag{24}$$

The solution is the boolean vector $\mathbf{x}$ that maximizes the algebraic connectivity of the resulting graph under node degree and edge constraints.

3.3. Remarks

Clustering is like a graph partitioning problem, while optimally partitioning a graph according to certain performance measure is NP-hard [9,8]. In [10], it is shown that the problem of adding a specified number of edges to an input graph to maximize the algebraic connectivity of the augmented graph is also NP-hard.

In the next two sections, we develop bounds and heuristic algorithms for the formulated problems. Specifically, we develop a *plane sweeping and clustering* (PSC) algorithm for the WMN clustering problem and a *greedy edge-appending* (GEA) algorithm for the FSO topology optimization problem, along with performance bounds. Both algorithms are evaluated via simulations in Section 6.

4. Lower tier: cluster formation

In this section, we first describe the PSC algorithm for cluster formation in the lower-tier WMN. We then develop a lower bound on the number of clusters, which can be used as a benchmark for the algorithm performance. Finally we discuss ways to reduce the computational complexity of the PSC algorithm.

4.1. Plane sweeping and clustering algorithm

Algorithm 1. Plane sweeping and clustering (PSC) algorithm

```
1   i=0;
2   Compute matrices A_i and H_i for graph G_i = G;
3   Choose the lower left corner node as base node;
4   while some nodes not clustered do
5       i++;
6       Add the base node to cluster M_i;
7       for h = 1 : h_max do
8           while some h-hop neighbors not checked do
9               Find a node u that is h-hops from the base node;
10              if (h_uk > h_max for any k ∈ M_i) or (f_{M_i} > f_max) then
11                  Go to Line 18;
12              else
13                  Add node u to cluster M_i;
14                  Update traffic load f_{M_i};
15              end
16          end
17      end
18      Determine cluster head for cluster M_i;
19      Generate the reduced graph G_i = G_{i-1} - M_i;
20      Generate matrices A_i and H_i for graph G_i;
21      Find the next base node;
22  end
23  if there is an isolated node then
24      Merge to a neighboring cluster with the max diameter;
        Repartition to get two new clusters;
25  end
26  Compute number of FSO devices for each cluster head ;
```

There have been a few clustering heuristics in the literature [20,21], and several schemes that consider QoS constraints [9,22]. However, we find it difficult to apply or enhance the existing algorithms for our problem with both traffic demand and delay constraints. Most existing schemes select a cluster head first and then select cluster members. For example, the recursive dominating set algorithm in [9] first selects a maximum degree node and then all its neighbors become cluster members. This approach may not be suitable for the case when the mesh

[4] It is worth noting that K_i is determined using Γ_{th} as in (11), rather than the exact link reliability $\Gamma_{ij} \ge \Gamma_{th}$, for $(i,j) \in E_{pot}$. As a result, some redundancy is provided for the capacity of cluster M_i, which is useful to accommodate fluctuations in the instantaneous cluster M_i load. If such redundancy is unnecessary, we can modify the (18) with $C_{opt} \cdot \sum_{j:(i,j) \in E} \omega_{ij} \ge \min\{f_{max}, f_{M_i}\}$, for all i.

routers have different traffic demands. Furthermore, prior work in [9,22] considers only the Internet traffic of the mesh routers, but not the inter-cluster traffic.

The PSC algorithm continually sweeps the network area to capture cluster members while satisfying the delay and load constraints, until every mesh router is assigned to a cluster. We then choose cluster head for each cluster formed. The objective is to minimize the number of clusters, thus reducing the deployment cost of the upper tier FSO network.

The pseudo-code of the PSC algorithm is presented in Algorithm 1. Starting from the lower left corner of the network region,[5] PSC iteratively sweeps the network area and forms a new cluster after each iteration. Each iteration consists of three steps as follows.

4.1.1. PSC Step 1: cluster formation

PSC first chooses a *base node* as starting point for the next cluster to be formed. It then explores the neighbors of the base node within h_{max} hops, and adds the neighboring nodes into the cluster, until one of the conditions, $D_{M_i} \le h_{max}$ or $f_{M_i} \le f_{max}$, is violated (lines 5–17 in Algorithm 1). D_{M_i} is the diameter of the new cluster M_i, which can be computed as

$$D_{M_i} = \max_{u,v \in M_i, u \neq v} [\mathbf{H}_i]_{uv}, \tag{25}$$

where $\mathbf{H}_i$ is the hop-count matrix for cluster M_i as defined in Section 3.1. Recall that $f_{M_i} = \max\{f_{o,M_i}, f_{M_i,o}\}$ is the larger one of the aggregate incoming or outgoing traffic load of cluster M_i, as given in (4).

It is worth noting that in the optimized clustering problem (6)– (10), the delay constraint is translated to $\sum_{u=1}^{n} x_{uv} \cdot h_{uv} \le h_{max}$ for all $u, v \in V$. That is, the hop count of the shortest path from any cluster member to its cluster head should be upper bounded by h_{max}. Since the cluster head could be located anywhere in the cluster (not necessarily at the center, see the discussion of PSC Step 2), this condition is satisfied by bounding cluster diameter with h_{max} in PSC, such that the maximum hop count for any node pair in the cluster does not exceed h_{max}. This also provides redundancy for accommodating fluctuations in the delay components.

4.1.2. PSC Step 2: cluster head selection

When a new cluster M_i is formed, PSC will select a cluster head from the nodes in M_i (line 18 in Algorithm 1). This node will be equipped with FSO transceivers and becomes a node in the upper tier FSO network.

Since all the incoming and outgoing traffic for a cluster will go through the cluster head, it should be judiciously chosen such that the relay traffic load within the cluster is minimized. If there is a gateway node in the cluster, it should be the cluster head, since all the traffic will finally be directed to and from there. If there are more than one gateway nodes in the cluster, it will be split into multiple smaller ones, each using a gateway node as cluster head. Each non-gateway node in the cluster will be associated with the closest gateway node after the partitioning. Note that this case rarely happens since the number of gateway nodes is usually small. If there is no gateway node in the cluster M_i, we select a node as cluster head by solving the following optimization problem:

$$\operatorname*{argmin}_{q \in M_i} \left\{ \sum_{u \in M_i, u \neq q} h_{uq} \left(\sum_{v \notin M_i, v=1}^{n} ([\mathbf{F}]_{uv} + [\mathbf{F}]_{vu}) \right) \right\},$$

where the overall relay load for the cluster's incoming and outgoing traffic is minimized.

4.1.3. PSC Step 3: graph and matrices update

If there is still node not clustered, PSC will obtain a reduced graph G_i by deleting the nodes in cluster M_i from graph G_{i-1}, and update the adjacency matrix $\mathbf{A}_i$ and hop-count matrix $\mathbf{H}_i$ for the reduced graph G_i. PSC then chooses in G_i the closest node from the current base node point, as well as close to certain trajectory (horizontal or vertical), as the next base node, and will repeat Steps (1) and (2) to generate a new cluster (lines 19–21 in Algorithm 1).

PSC sweeps a network plane until all the nodes are covered. However, there could be single-node clusters (e.g., a single node in M_i) in the border zone. In this case, we refine the cluster formation by first identifying the neighboring cluster with the largest diameter (e.g., cluster M_j), and then iteratively reassigning members of M_j to M_i, until the diameter or traffic load constraints of M_i is met, or when the two clusters M_i and M_j have similar number of nodes. It is easy to see that the total number of clusters are still the same, and both refined clusters are feasible.

Line 27 in Algorithm 1 computes the number of FSO transceivers for each cluster head as given in (11). The number is determined by the traffic load f_{max} and the capacity of FSO links. It is the degree of the cluster head, and determines the number of edges for the upper-tier FSO network.

4.2. Lower bound

We next derive a lower bound on the objective value achieved by PSC. To avoid trivial cases[6] and without loss of generality, we assume that every point in the deployment area, S, is covered by at least one mesh router. The lower bound on the number of clusters is given in the following theorem. It will be useful for evaluating the performance of heuristic clustering algorithms, especially for large-scale networks.

Theorem 1. *$n_0 = \lceil 4S/(\pi r^2 h_{max}^2) \rceil \le n_c$ is a lower bound on the total number of clusters.*

Proof. In order to minimize the number of clusters, the size of each cluster should be maximized. For given constraints h_{max} and f_{max}, there are two cases when a cluster is formed, i.e., the cluster size is confined by either

[5] Since PSC is to sweep all the mesh routers, starting from a corner is a better choice than, say, starting from a gateway node. Since a gateway node could be located anywhere in the network, starting from a gateway node may complicate the sweep process.

[6] For example, the lower bound will be one if all the mesh routers are installed at the same location with low traffic load.

(i) the maximum cluster diameter or (ii) the maximum aggregate traffic load.

Case i: Since the cluster size is confined by h_{max}, the minimum number of clusters can be obtained by forming each cluster with diameter h_{max}. With transmission range r, the maximum Euclidean distance between any two cluster members is thus rh_{max}. The maximum coverage of a cluster is $\pi r^2 h_{max}^2/4$, which is a disk with radius $rh_{max}/2$. To cover the deployment area, the minimum number of clusters can be estimated as $\lceil 4S/(\pi r^2 h_{max}^2)\rceil$.

Case ii: Since the cluster size is confined by f_{max}, the maximum cluster diameter bound is not violated when the cluster is formed. As a result, the Euclidean distance of the cluster diameter cannot be larger than rh_{max}. Therefore, the minimum number of clusters for this case cannot be lower than that in Case i.

Therefore, we conclude that the minimum number of clusters is $n_0 = \lceil 4S/(\pi r^2 h_{max}^2)\rceil$. □

4.3. Complexity reduction

PSC involves manipulation of several matrices. Let n be the number of mesh routers that have to be clustered. Assume that the PSC algorithm generates n_c clusters with k cluster members. The computation costs to generate adjacent matrix **A** and **H** are $O(n^2)$ and $O(n^{2.376})$, respectively. Indeed, the $n \times n$ matrix multiplication has complexity $O(n^{2.376})$ using the Coppersmith–Winograd algorithm [23]. The complexities, in addition, to verify h_{max} and f_{max} condition per iteration are $O(k^2)$ and $O(k \cdot n)$, respectively. The merging cost per isolated node is $O(n_c \cdot k)$, which is smaller than $O(n_c \cdot n^{2.376})$. Thus, we can see that the computation cost to update **H** is the dominating part. The computation costs of PSC algorithm is $O(n_c \cdot n^{2.376})$. In this section, we discuss techniques on further reducing the computational complexity.

Theorem 2. *Consider reduced graph $G' = G - k$ for $k \in V(G)$. Let* **H***and* **H**′ *be the hop count matrices for graphs G and G′, respectively. For any $u \in V(G)$ such that $h_{uk} \geq h_{max}$, we have $[\mathbf{H}]_{uv} = [\mathbf{H}']_{uv}$ for all $v \in V(G)$ and $v \neq k$.*

Proof. For $u,\ k \in V(G)$ and $u \neq k$, assume that there is an uk-walk, of which the shortest path has n walks, meaning $h_{uk} = n$. If $h_{uk} > h_{max}$, we have $h_{uk} = 0$. Then, the uk-walk is denoted by $\nu_u = \nu_0 e_0 \nu_1 e_1 \cdots \nu_{n-1} e_{n-1} \nu_n = \nu_k$ for $n \leq h_{max}$. For the reduced graph $G' = G - k$, assume that there is a vertex $v \neq k$; h'_{vk} is the hop count of the shortest path from vertex v to k. We consider the following three cases.

Case i: Assume that $h_{uk} > h_{max}$. If $h_{uv} \leq h_{max}$, h'_{uv} is equal to h_{uv} since the uv-walk does not include ν_k by the case condition $h_{uk} > h_{max}$. If $h_{uv} > h_{max}$, h'_{uv} is also equal to h_{uv} (i.e., both are 0), since deleting a vertex does not reduce the number of any walk. Thus we have $[\mathbf{H}]_{uv} = [\mathbf{H}']_{uv}$.

Case ii: Assume that $h_{uk} = h_{max}$. If $h_{uv} < h_{max}$, h'_{uv} is equal to h_{uv} since the uv-walk does not include ν_k by the assumption $h_{uk} = h_{max}$. If $h_{uv} = h_{max}$, there should be another walk which does not go through ν_k. If not, the condition, $h_{uv} = h_{max}$, will be violated. For the condition $h_{uv} > h_{max}$, h'_{uv} is also equal to h_{uv} since both of them are 0 in this case. Thus $[\mathbf{H}]_{uv} = [\mathbf{H}']_{uv}$ holds true in this case.

Case iii: Assume that $h_{uk} < h_{max}$. If $h_{vk} < h_{max}$ and $h_{uv} = h_{uk} + h_{kv}$, it is possible that the shortest uv-walk is unique and ν_k is in the path. It is possible to have $h'_{uv} \neq h_{uv}$.

We have that $[\mathbf{H}]_{uv} = [\mathbf{H}']_{uv}$ when $h_{uk} \geq h_{max}$. □

Corollary 1. *Consider reduced graph $G' = G - k$ for $k \in V(G)$. Assuming disjoint vertices $u, v, k \in V(G)$, it is possible to have that $h'_{uv} \neq h_{uv}$, if $h_{uv} = h_{uk} + h_{vk}$ for $h_{uk} < h_{max}$ and $h_{vk} < h_{max}$.*

Proof. We have $h_{uv} \neq h'_{uv}$ if $h_{uv} = h_{uk} + h_{vk}$ for $h_{uk} < h_{max}$ and $h_{vk} < h_{max}$. If one of the vertices u or v is more than h_{max} hops from k, we have $h'_{uv} = h_{uv}$ according to Theorem 2. We consider three cases for $h_{uk} < h_{max}$ and $h_{vk} < h_{max}$ as follows:

Case i: If $h_{uv} < h_{uk} + h_{vk}$, it means that the uv-walk does not go through vertex k.

Case ii: If $h_{uv} = h_{uk} + h_{vk}$, it is possible that the shortest path between u and v including vertex k is unique. In that case, when k is deleted, the unique shortest path will no longer exist. We then have $h'_{uv} > h_{uv}$ or $h'_{uv} = 0$ according to (5).

Case iii: The case of $h_{uv} > h_{uk} + h_{vk}$ cannot happen.

Therefore, we have $h'_{uv} \neq h_{uv}$ when $h_{uv} = h_{uk} + h_{vk}$ for $h_{uk} < h_{max}$ and $h_{vk} < h_{max}$. □

Corollary 2. *Consider reduced graph $G' = G - k$ for $k \in V(G)$. Assume that vertices $u,\ v,\ k \in V(G)$ are disjoint. Then $[\mathbf{H}']_{uv}$ can be obtained by the power calculation of an adjacency matrix, denoted as* **A**″*, consisting of rows and columns corresponding to vertices s and $t \in V(G)$, satisfying $h_{ks} \leq 2(h_{max} - 1)$ and $h_{kt} \leq 2(h_{max} - 1)$.*

Proof. Consider disjoint vertices $u,\ v,\ k \in V(G)$, $h_{uv} = h_{uk} + h_{vk}$ for $h_{uk} < h_{max}$ and $h_{vk} < h_{max}$. Therefore $[\mathbf{H}']_{uv}$ could be different from $[\mathbf{H}]_{uv}$. From Corollary 1, vertices u and v can be located at most $h_{max} - 1$ hops from vertex k. Let $p \in V(G')$, $p \neq u$, and $p \neq v$. In order to check $[\mathbf{H}']_{uv}$, p should be located less than $h_{max} - 1$ hops from u and v. According to location conditions of u and v, and p, the adjacency matrix, **A**″, with vertices less than $2(h_{max} - 1)$ hops from k can be used to compute $[\mathbf{H}']_{uv}$. □

As examples, consider the 500-node network shown in Fig. 3. In Fig. 3(a), a node around the center of the region is included into an existing cluster. In Fig. 3(b), a node near the boarder is included into an existing cluster. We show in the figure the set of nodes that are required for getting the reduced adjacency matrix **A**″, obtained for the reduced graph. Such nodes are marked as red squares and green diamonds. According to Theorem 2 and the corollaries, we need to recompute the **H**′ entries for the red square nodes in the figures, while the green diamond nodes are also involved in the computation. Except for the red square nodes, the **H**′ entries for all other nodes are not changed by the node deletion. Compared to recomputing the entire **H**′ using **A**, significant computation reduction can be achieved by using the much smaller adjacency matrix **A**″. We also find that an edge node reduction achieves more computation reduction than central node reductions, as illustrated in Fig. 3. Due to the sequential sweeping operation of PSC, it is usually the case of edge node reduction, resulting in considerable computation savings.

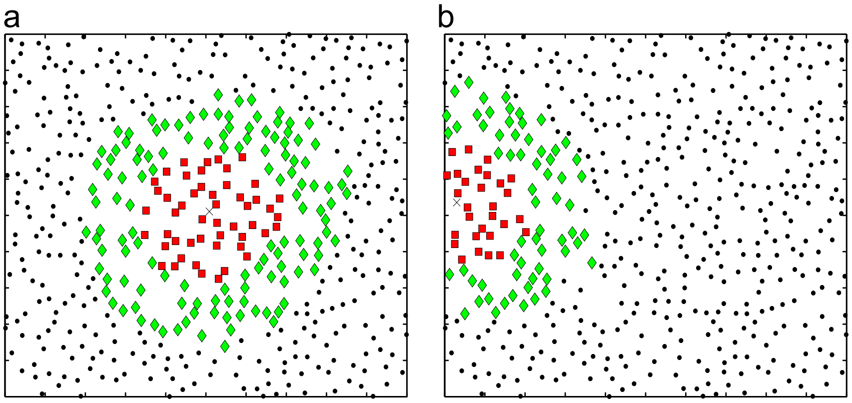


Fig. 3. The subgraph corresponding to $\mathbf{A}''$ (marked as squares and diamonds) after deleting a node from a 500-node network deployed in a 1500×1500 m^2 area with $r = 150$ m and $h_{max} = 3$. (a) A central node. (b) A border node.

5. Upper tier: topology optimization

The objective of topology optimization is to maximize algebraic connectivity for the upper tier FSO network, while satisfying the traffic load requirements. We first derive a theoretic upper bound for the increase in algebraic connectivity when an additional edge is added. We then present the centralized *greedy edge-appending* (GEA) algorithm and its distributed version.

5.1. Theoretic upper bound

The algebraic connectivity upper bound is useful for evaluating the performance of GEA, since the global optimal lies in between the GEA solution (i.e., a lower bound) and the upper bound. Let $G(V, E)$ be a connected and weighted graph and let G' be a graph by adding an edge $l \sim (i, j)$ to G, denoted as $G' = G + l$. The upper bound is given in the following.

Theorem 3. *An upper bound on the algebraic connectivity of graph G' is given as*

$$\lambda_2(G') \le \min\{\lambda_3(G), \lambda_2(G) + \omega_{ij} \cdot (\nu_i - \nu_j)^2\}, \tag{26}$$

where ν_i and ν_j are respectively the ith and jth elements of the normalized eigenvector $\mathbf{v}$ of $\mathbf{L}(G)$ corresponding to eigenvalue $\lambda_2(G)$.

Proof. Let $\mathbf{L}$ and $\mathbf{L}'$ be the Laplacian matrix of G and G', respectively. We first show that $\lambda_2(G') \le \lambda_3(G)$. Assuming that graph G has one component, we have $\mathbf{L}' = \mathbf{L} + \omega_l \cdot \mathbf{a}_l \cdot \mathbf{a}_l^T$. Since the Laplacian matrix $\mathbf{L}$ is symmetric positive semidefinite, it satisfies the following conditions: (i) $\mathbf{L} = \mathbf{L}^T$, (ii) $\mathbf{L} \succcurlyeq 0$, and (iii) $\mathbf{L} \cdot \mathbf{1} = \mathbf{0}$, where $\mathbf{1} = [1,\ 1, \ldots, 1]^T$ and $\mathbf{0} = [0,\ 0, \ldots, 0]^T$.

Let $\lambda_1 \le \lambda_2 \le \cdots \lambda_{n_c}$ be the eigenvalues of $\mathbf{L}$ and $\lambda_1' \le \lambda_2' \le \cdots \lambda_{n_c}'$ be the eigenvalues of $\mathbf{L}'$. According to Theorem 3.2 in [17], we have $\lambda_1 \le \lambda_1' \le \lambda_2 \le \lambda_2' \le \lambda_3 \le \lambda_3' \cdots \le \lambda_{n_c} \le \lambda_{n_c}'$. Therefore, it follows that $\lambda_2(G') \le \lambda_3(G)$.

We next prove the second part. Let $\mathbf{v} = (\nu_1, \nu_2, \ldots, \nu_{n_c})$ be the normalized eigenvector of $\mathbf{L}$ corresponding to $\lambda_2(\mathbf{L})$. According to the Courant–Fischer formula, the second smallest eigenvalue of $\mathbf{L}'$ can be written as

$$\lambda_2(\mathbf{L}') = \min_{\mathbf{x} \perp \mathbf{1}, \mathbf{x} \ne \mathbf{0}} \frac{\langle \mathbf{L}' \cdot \mathbf{x}, \mathbf{x} \rangle}{\langle \mathbf{x}, \mathbf{x} \rangle},$$

where $\langle \cdot, \cdot \rangle$ is the *inner product* of two vectors [17]. We have

$$\begin{aligned}
\lambda_2(\mathbf{L}') &= \min_{\mathbf{x} \perp \mathbf{1}, \mathbf{x} \ne 0} \frac{\langle \mathbf{L}' \cdot \mathbf{x}, \mathbf{x} \rangle}{\langle \mathbf{x}, \mathbf{x} \rangle} \le \frac{\langle \mathbf{L}' \cdot \mathbf{v}, \mathbf{v} \rangle}{\langle \mathbf{v}, \mathbf{v} \rangle} \\
&= \frac{\sum_{(a,b) \in E(G')} \omega_{ab} \cdot (\nu_a - \nu_b)^2}{\sum_{a \in V(G)} \nu_a^2} \\
&= \frac{\sum_{\{(a,b) \in E(G)\} \cup \{(i,j) \notin E(G)\}} \omega_{ab} \cdot (\nu_a - \nu_b)^2}{\sum_{a \in V(G)} \nu_a^2} \\
&= \sum_{(a,b) \in E(G)} \omega_{ab} \cdot (\nu_a - \nu_b)^2 + \omega_{ij} \cdot (\nu_i - \nu_j)^2 \\
&= \lambda_2(\mathbf{L}) + \omega_{ij} \cdot (\nu_i - \nu_j)^2.
\end{aligned}$$

The fourth equality is due to the fact that $\mathbf{v}$ is a normalized vector. We have $\lambda_2(\mathbf{L}') \le \lambda_2(\mathbf{L}) + \omega_{ij}(\nu_i - \nu_j)^2$. Since both cases should be satisfied, we have that $\lambda_2(G') \le \min\{\lambda_3(G), \lambda_2(G) + \omega_{ij} \cdot (\nu_i - \nu_j)^2\}$. □

5.2. Greedy edge-appending algorithm

Algorithm 2. The greedy edge-appending (GEA) algorithm

```
Build a degree bounded minimum spanning tree;
while there remains an edge to be inserted do
    Append the edge with the largest ω_ij · (ν_i − ν_j)^2
    among remaining edges in E_pot;
      Conditioned on : (d_i < K_i) and (d_j < K_j);
      Break a tie by considering the node pair with the min degree
    first and then that with the max distance;
    Update graph and the corresponding Laplacian matrix;
end
```

The pseudo-code of the GEA algorithm is given in Algorithm 2. The procedure starts with n_c vertices and a null edge set. Therefore, the initial value of $\lambda_2(\mathbf{L})$ is zero and will remain zero until the graph becomes connected (see Facts 4 and 5). To speed up the initialization phase, we first apply an enhanced version of Prim's algorithm to build a *degree bounded minimum spanning tree* (MST) [24]. Next, GEA iteratively picks an edge from the remaining edges in E_{pot} and appends it to the graph such that the algebraic connectivity is progressively improved.

Recall that $\mathbf{v}=(\nu_1,\ \nu_2,\ldots,\nu_{n_c})$ is the normalized eigenvector corresponding to $\lambda_2(\mathbf{L})$. During each iteration, GEA chooses an edge with the largest $\omega_{ij}\cdot(\nu_i-\nu_j)^2$ value among all the remaining edges in E_{pot}. As shown in the proof of Theorem 3, $\omega_{ij}\cdot(\nu_i-\nu_j)^2$ is an upper bound on the increase in algebraic connectivity achieved by adding edge $(i,\ j)$. This strategy will achieve the largest expected increase in $\lambda_2(\mathbf{L})$.

In the case of a tie where multiple edges have the same maximum $\omega_{ij}\cdot(\nu_i-\nu_j)^2$ value, GEA chooses the edge whose endpoint has the minimum degree. If there is a tie again, GEA chooses the edge with the largest distance. This strategy is motivated by the inequality condition (12), i.e., $4/(D\cdot n_c)\le\lambda_2(G)\le\delta(G)$, which implies that algebraic connectivity may be improved by increasing the minimum degree and/or by decreasing the diameter of the graph.

Recall that m is the total number of edges to be inserted. It takes $O(n_c^2)$ to build the minimum spanning tree with n_c-1 edges [24]. Since a Laplacian matrix of a connected graph is symmetric tridiagonal, it takes $O(n_c^2)$ to compute its eigenvector [25]. The complexity of appending the remaining $(m-n_c+1)$ edges mainly comes from computing eigenvector $\mathbf{v}$, each takes $O(n_c^2)$, for the remaining edges in E_{pot}. The overall GEA complexity is $O(n_c^2+(m-n_c+1)n_c^2m_{pot})$.

5.3. Alternative and distributed algorithms

In addition to the bottom-up approach taken in GEA, alternative methods can be explored. For example, a top-down approach that deletes an edge from current topology starting from $G(V,\ E_{pot})$. In each iteration, the edge with the minimum $\omega_{ij}\cdot(\nu_i-\nu_j)^2$ value will be selected and deleted, until $m_{pot}-m$ edges are deleted. Note that this approach may not be suitable for large networks where $m_{pot}\gg m$.

As another intuitive edge-appending algorithm, we can select the edge with the largest weight to insert, since a more reliable edge could make stronger connection than weak link. That is, the strategy is to select the edge with maximum ω_{ij} among unselected potential edges. This algorithm can guarantee the highest link weight sum, but the resulting algebraic connectivity is hard to predict. We call it *simple greedy heuristic*, and compare its performance with the GEA algorithm in the simulations.

The GEA algorithm described in Algorithm 2 is executed at a centralized entity with global information. It would be interesting to develop a distributed version of GEA that does not require the centralized entity. Such a distributed version executed at FSO node i is presented in Algorithm 3. First, a distributed MST algorithm is executed at each node to form a tree topology [26], as given in lines 1–3. As in prior work [27], we assume that nodes within a connected graph component share topology information with each other. Therefore, once the tree is formed, the FSO nodes can share and obtain information of the current topology. Then, node i iteratively identifies a neighbor j, such that inserting edge (i,j) achieves the largest potential increase in algebraic connectivity among all other possible links to node i's neighbors. Then node i will request to connect to node j, using one of the remaining FSO transceivers at each node. If it succeeds, it will increase its node degree by 1 and continue to find the next edge to insert. Note that since each node autonomously decides which edge to insert, it is possible that when node i tries to insert edge $(i,\ j)$, node j has already used up all of its transceivers in the meantime. In this case, node i will give up and try to find the next edge to insert to other neighbors. This procedure, shown in lines 4–10, is repeated at each node until all its transceivers are used.

In the distributed GEA, the next edge is chosen from those edges in E_{pot} that connect itself to its neighbors. In the centralized GEA, however, the next edge is chosen from all the remaining edges in E_{pot}. Thus, the distributed GEA algorithm is expected to provide smaller potential increase in algebraic connectivity than that achieved by the centralized GEA algorithm. As will be shown in our simulation studies, it slightly sacrifices connectivity performance to eliminate the need for a centralized entity.

Algorithm 3. Distributed GEA algorithm at FSO node i

1 **while** *the network is partitioned* **do**
2 | Participate in the distributed MST algorithm;
3 **end**
4 **while** *node degree constraint* (23) *is not met* **do**
5 | Read current topology information;
6 | Select a neighbor $j=\arg\max\{\omega_{ij}\cdot(\nu_i-\nu_j)^2\}$;
7 | **if** *succeed in inserting edge* (i,j) **then**
8 | | Increase node degree by 1;
9 | **end**
10 **end**

6. Simulation results

In this section, we evaluate the performance of PSC and GEA. The algorithms are implemented in Matlab. We use fixed transmission range r for the lower tier radio network, and simulate log-normal fading for the upper tier FSO links. In each simulation, a large number of mesh routers are uniformly distributed in a square region and a small number of gateway nodes are randomly chosen from the set of mesh routers.

The traffic matrix is randomly generated as follows. Each mesh router generates two types of traffic: (i) *Internet traffic* that is forwarded to/from a gateway, and (ii) *inter-cluster* traffic that is delivered between two mesh routers. The inter-cluster traffic is randomly distributed among mesh routers and the Internet traffic is directed to the closest gateway. We assume that the incoming Internet traffic rate is twice that of the outgoing Internet traffic (i.e., more downloading traffic than uploading traffic).

6.1. Optimized clustering results

We first evaluate the PSC performance with regard to number of clusters and computation reduction. Although there is considerable literature on clustering in wireless networks, we find that many existing algorithms are designed for highly specific problems and may not apply for the FSO-based wireless access network considered in this paper. For example, energy efficiency has been the major consideration for clustering in wireless sensor networks [20,5], while power conservation is not an issue in WMNs since the mesh routers are usually plugged into power outlets. In addition, traffic demand between clusters, as represented by the traffic matrix, is not considered in many prior work on clustering in WMNs [9]. Therefore, we compare the PSC algorithm with the lower bound in this section to avoid unfair comparisons.

We follow the setting in [9] to consider a small network with 175 mesh routers uniformly deployed in a $1000 \times 1000\ \text{m}^2$ region. There are two gateway nodes. The radio transmission range is $r = 100$ m, and the minimum distance between any two mesh routers is 60 m. The cluster diameter bound is set to $h_{max} = 4$. Applying PSC, the mesh routers are divided into 20 clusters, as illustrated in the lower tier of Fig. 4. The upper tier in Fig. 4 is constructed by GEA.

In Fig. 5, we plot the number of clusters formed by PSC for increasing cluster diameter bound h_{max}, where the same setting as in Fig. 4 is used. For each h_{max} value, we randomly generate 25 different topologies for the mesh network using different random seeds. For each random topology, the PSC algorithm is executed to form the clusters. Then each point on the PSC curve is the average of the 25 samples of number clusters. This way, we can eliminate the impact of the random topology on the algorithm performance. We also use the 25 samples to compute the 95% confidence intervals, which are plotted as error bars on the PSC curve in Fig. 4, indicating that we

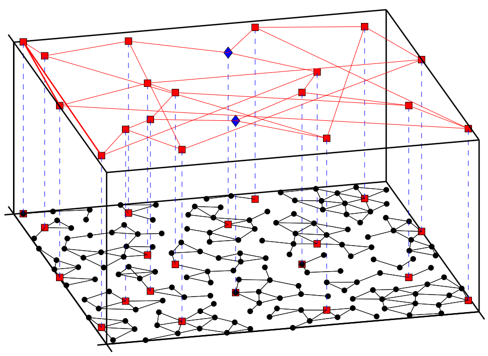


Fig. 4. A tiered network designed using PSC and GEA.

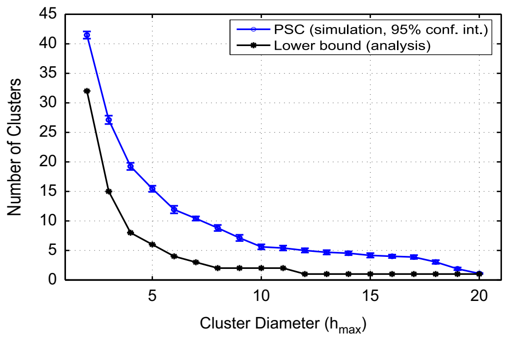


Fig. 5. PSC performance: number of clusters for various h_{max} values.

are 95% confident that the number of clusters will fall within the range of the bars. The confidence intervals are generally negligible, implying that the PSC performance is quite stable. As expected, the number of clusters quickly decreases for increased h_{max}, and converges to one when $h_{max} = 20$. We also plot the lower bound as given in Theorem 1. The lower bounding curve lies below the PSC curve. It can provide a rough estimate for the number of clusters formed, but it is not very tight for some h_{max} values. It would be an interesting future work item to further tighten this bound.

Finally we examine the computational cost reduction as given in Theorem 2 and Corollaries 2.1 and 2.2. Let $G'' \subseteq G'$ be the *subgraph* corresponding to $\mathbf{A}''$. We define the computation reduction ratio as

$$\gamma = \frac{|V(G')| - |V(G'')|}{|V(G')|} \times 100(\%). \qquad (27)$$

A larger ratio implies multiplication of smaller matrices and larger computation reduction. Fig. 6 shows the reduction ratios by deleting a node from a 500-node network and a 1000-node network. It can be seen that deleting a border nodes results in smaller $\mathbf{A}''$ than deleting a central node, and more significant reductions can be achieved for larger networks.

6.2. *Topology optimization results*

When the clusters are formed, location and degree information of cluster heads are used as input to GEA algorithm for optimizing the top tier topology. We assume that $\Gamma_{th} = 0.9$, $I/I_0 = 0.8$, and $\lambda = 1,550$ nm. The link reliability Γ_{ij} and weight ω_{ij} are determined by distance and weather condition using the FSO channel model. The distances are computed from cluster head locations. For weather condition, the index of refraction structure parameter C_n^2 is randomly generated in $[10^{-16}, 10^{-14}]\ \text{m}^{-2/3}$. The typical weather condition is set to $C_n^2 = 10^{-15}\ \text{m}^{-2/3}$.

6.2.1. *Small FSO network*

We first examined a small network with the same setting as that in Fig. 4. With cluster diameter bound $h_{max} = 3$, the mesh routers are divided into 26 clusters. The traffic demand requires that $m = 39$ FSO edges should be added in the top tier, and the total degree is $\sum_{i=1}^{26} K_i = 2 \cdot m = 78$. We first

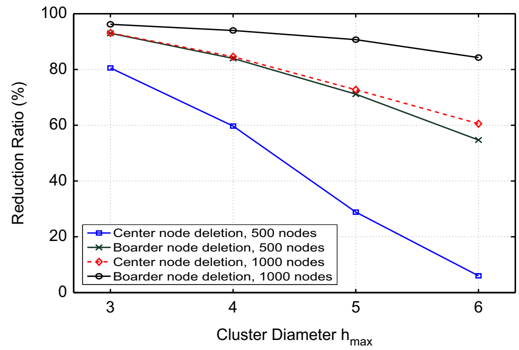


Fig. 6. Computational cost reduction achieved by using the reduced adjacency matrix $\mathbf{A}''$.

execute the enhanced Prim's algorithm to build a degree bounded minimum spanning tree, which uses 25 edges. GEA then iteratively inserts 14 edges to the tree graph.

In Fig. 7, we plot the resulting algebraic connectivity after each edge is appended. For comparison, we also plot the upper bound as given in Theorem 3 and the global optimal values found by an *exhaustive search*, which is possible for this small network. We also implement the distributed GEA algorithm and the simple greedy algorithm, and plot the corresponding algebraic connectivity curves in the figure.

We find that the upper bound provides a very good approximation for the global optimum. The GEA curve overlaps with the global optimal curve for most of the iterations, except for the 26th, 33rd, 35th, and 38th edge insertions. When GEA terminates, the upper bounding algebraic connectivity is 0.6357 and the GEA value is 0.6353, which are all identical to the global optimum. There is a moderate gap between the centralized GEA curve and the distributed GEA curve. The distributed GEA terminates with algebraic connectivity value 0.5698, which is 89.7% of the global optimal solution. This shows that the distributed GEA slightly sacrifices the achievable algebraic connectivity in order to exclude the need for a centralized entity. Although the simple greedy algorithm always chooses the link with the best quality to insert and achieves the largest link weight sum among all the algorithms, it performs poorly with respect to algebraic connectivity. The algebraic connectivity achieved by the simple greedy algorithm is 0.2624, which is 41.3% of the global optimum.

6.2.2. Large FSO network

We next design a large-sized network deployed in a $6000 \times 6000\ m^2$ area. The PSC algorithm forms 184 clusters, and 326 edges are to be inserted according to the traffic demand. The unweighted node degree of each node ranges from 2 to 5. Building the spanning tree uses 183 edges, and the remaining 143 edges are iteratively appended by GEA.

In Fig. 8, we plot the algebraic connectivity traces as edges are iteratively inserted. Due to the large network size, exhaustive search is not feasible. We plot the upper bound as given in (26) as indicator of the global optimal solutions. We find the GEA curve overlaps with the upper bound curve for most of the iterations. When GEA terminates, it achieves the same algebraic connectivity value 0.3527 as the upper bound. Clearly, not only the upper bound is highly tight, the greedy heuristic GEA algorithm also achieves the global optimum for this large network. Similarly to the case of small network, the distributed GEA and simple greedy algorithms also show similar trends. At the last iteration, the algebraic connectivity of distributed GEA algorithm is 0.3010, which is 85.3% of the upper bound. However, the simple heuristic algorithm achieves an algebraic connectivity value of 0.0413, which is 11.7% of the upper bound.

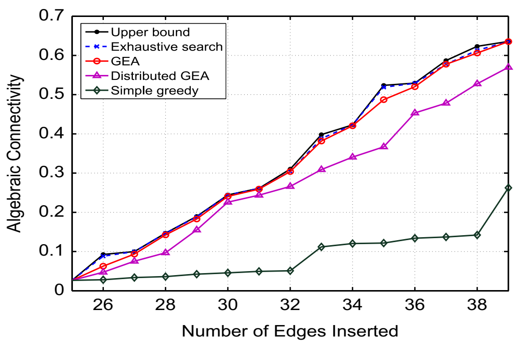


Fig. 7. Algebraic connectivity for the small network.

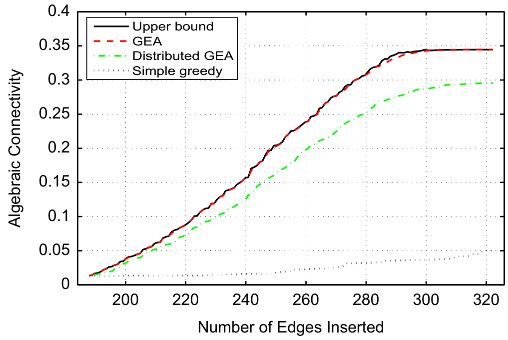


Fig. 8. Algebraic connectivity for the large network.

Using the same 184-cluster FSO network, we examine the impact of several network parameters on algebraic connectivity in Fig. 9, including the minimum node degree and link weight. The minimum node degree is determined by the traffic matrix and is an indicator of the network traffic demand. Link weights are determined by weather conditions. In Fig. 9(a), we plot algebraic connectivity for increased minimum node degree. It can be seen that the three algebraic connectivity curves all increase with the minimum node degree. This is because a larger minimum node degree allows more edges to be inserted. According to Fact 6, algebraic connectivity increases when more edges are inserted. However, both GEA curves increase with minimum node degree at a much fast rate than the simple greedy algorithm curve, indicating their effectiveness in building well connected graphs.

In Fig. 9(b), we reduce link reliability by certain ratios and plot the resulting algebraic connectivities. Since link reliability largely depends on weather condition, this is equivalent to examine the algorithms under various weather conditions. As expected, all the three curves decrease as link reliability is reduced. When link reliability is reduced by 50%, the GEA algebraic connectivity is reduced in half, i.e., from 0.3527 to 0.1764. The distributed GEA value is reduced from 0.3010 to 0.1575, and the simple greedy algorithm value is constantly low and is reduced from 0.0413 to 0.0206. Interestingly, choosing the most reliable edges as in the simple greedy algorithm does not necessarily provide strongly connected topology with respect to algebraic connectivity.

6.2.3. Comparison with regular topologies

A natural question that arises at this juncture is “how about topologies with regular structures?” To answer this question, we consider grid network deployment, such as

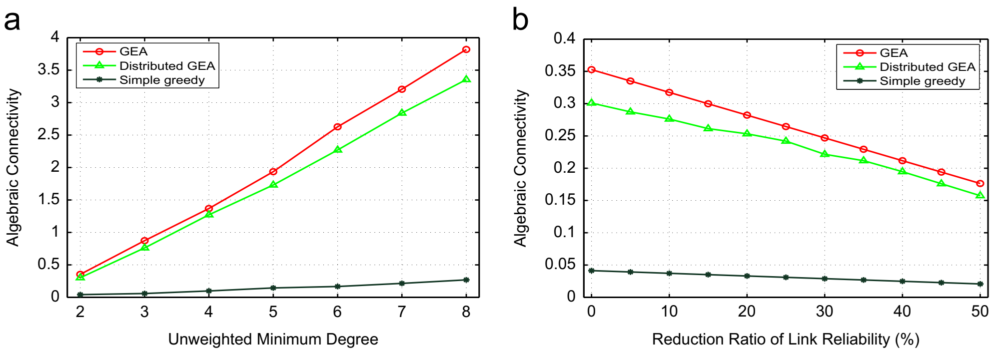


Fig. 9. Algebraic connectivity vs. network parameters. (a) The impact of minimum degree. (b) The impact of link reliability.

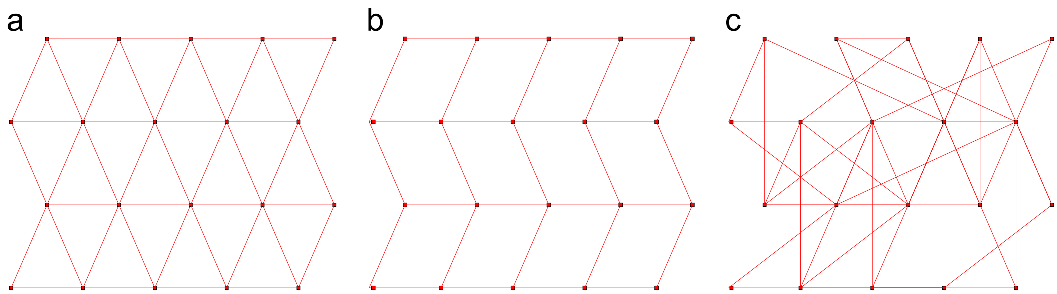


Fig. 10. Lattice topologies of 20-node networks. (a) Triangle lattice. (b) Square lattice. (c) GEA topology.

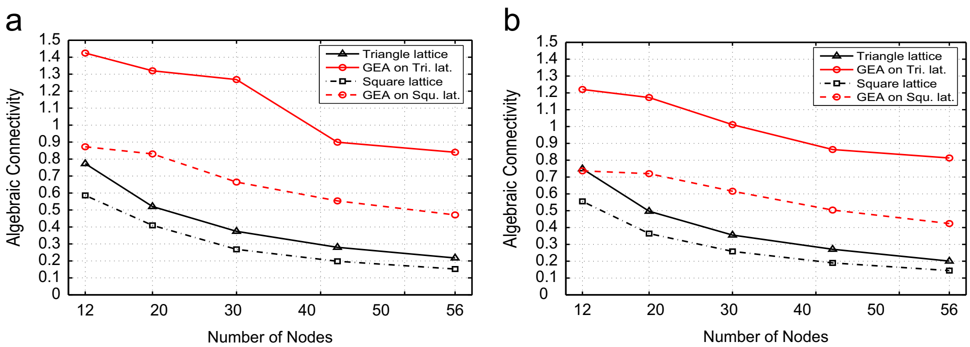


Fig. 11. Algebraic connectivity of lattice networks. (a) Unweighted lattice network. (b) Weighted lattice network.

the triangular and square lattice networks as shown in Figs. 10(a) and (b). We assume that each node in the triangular lattice network can have a fixed degree from 2 to 6, which depends on its location (i.e., center or edge). Each node in the square lattice network can have a degree between 2 and 4. We also execute GEA for the same network deployment and the same number of edges. The topology designed by GEA is plotted in Fig. 10(c), which looks quite different from the lattice ones.

How about algebraic connectivity performance? In Fig. 11, we plot the algebraic connectivity curves for different network sizes. For comparison, we first assume that FSO links are not weighted (or, all the links have identical reliability 1) in Fig. 11(a). We find that lattice networks are not well connected by the measure of algebraic connectivity. The algebraic connectivity of the 20-node triangular network is 0.5204, which is only 39.4% of the corresponding GEA value. For the 56-node square network, the algebraic connectivity is 0.1522, which is only 32.3% of the corresponding GEA value. This is due to the fact that algebraic connectivity is determined by various graph invariants, not only edge/vertex connectivity, but also graph diameter and minimum degree. In the lattice networks, nodes only connect to their neighbors, so their network diameters are usually larger than that of the GEA topology, indicating smaller algebraic connectivity according to inequality (12).

There are several interesting observations to make. First, algebraic connectivity is a decreasing function of network size, since a larger network usually has a larger diameter. Second, it can be seen that the square lattice network is a subgraph of the triangular lattice network. According to Fact 6, the triangular lattice network always has better connectivity than the square lattice network. Finally, we also considered weighted lattice networks, in which each FSO can have different weights according to the channel model in Section 2.2. The results are plotted in Fig. 11(b), where similar trends can be observed for the four curves. However the algebraic connectivity value in Fig. 11(b) is always lower than the corresponding value in Fig. 11(a), due to the worse link qualities.

7. Related work

This paper is closely related to the class of network planning work [28–30,22]. Network planning problems usually belong to the class of combinatorial optimization problems, which are NP-hard, and *metaheuristics*, e.g., simulated annealing [29] or genetic algorithms [30,22] are used to provide sub-optimal solutions. The main limitations of these approaches are the lack of performance guarantees and the relatively high complexity.

This work is also related to FSO research. See excellent surveys in [7,31]. Major FSO research has focused on the PHY so far, such as hardware architecture [31] and optical channel modeling [13]. Recently, there have been several work on the design [32–34] and (re)configuration [7,27,35,36] of FSO networks. As in the class of network planning work, ILPs are usually formulated and various heuristic algorithms are proposed to provide sub-optimal solutions. In [33,34], the authors study two generic family of mesh-based topologies for FSO networks: GPeterNet, a graph theoretic framework based on the well known Petersen graph, and FraNtiC, a fractal geometric architecture, for arbitrary access network deployments.

FSO links and traditional RF links are complementary to each with respect to data rate, interference, robustness, and range. Several papers have investigated the hybrid RF/FSO networks for enhanced performance [36–38]. In [37], genetic algorithms are used to improve the capacity performance with a minimum number of hybrid FSO/RF gateways. In [36], the authors propose to adaptively adjust both transmission power (of RF and FSO transmitters) and the optical beamwidth, to meet prescribed QoS requirements. In [38], the authors investigate radio signal transmission over terrestrial optical wireless channels under a WiMAX network setting, and provide an outage probability analysis. The “pseudo-wired” FSO links are highly desirable for interference management and security. As contrast to wired links, the FSO links also allow great flexibility for adaptive to network dynamics since they are steerable. In [39], the authors exploit slow-fading FSO channels and propose an adaptive transmission algorithm that can adjust transmit power and modulation according to channel status information feedback. In [40], the authors propose a fiber-bundle approach for beam steering to enhance the tolerance of optical link misalignment.

WMNs are known to suffer from capacity and fairness problems, especially when size grows [4]. A hierarchical network architecture represents a solution to the scalability problem, as observed in wireless sensor networks [20,5]. In [20], Bandyopadhyay and Coyle present a hierarchical clustering algorithm for wireless sensor networks with the objective of energy efficiency. In [9], the authors study the problem of optimal placement of the gateway nodes in a WMN and propose a recursive *dominating set* (DS) selection algorithm under delay and bandwidth constraints. Genetic algorithms are used in [22] to improve the capacity performance with minimum number of hybrid FSO/RF gateways. A similar approach, using as wired backend the Passive Optical Network architecture, has been widely studied under the definition of WOBAN (Wireless-Optical Broadband Access Networks) [41]. The main difference here is that, unlike the optical backhaul network that has a static star topology [41], the FSO links considered in this paper can be flexibly redirected to form different topologies in response to weather change or node/link failures, while the mesh topology of the FSO network can provide richer connectivity.

Algebraic connectivity is a useful tool from *spectral graph theory* [15,19]. It has been used in several papers as a measure of connectivity [42,43]. In an interesting work [42], Ghosh and Boyd present an SDP formulation as well as a greedy perturbation heuristic for adding edges into existing graph, with the objective to maximize its algebraic connectivity. In [43], the authors propose a multi-level algorithm for finding the best locations for a given set of relays, for enhancing the connectivity of wireless sensor networks. A standard SDP problem is formulated and solved in each level.

8. Conclusion

We studied the problem of design and optimization of a tiered wireless access network. The lower tier mesh network is first partitioned into clusters, and then the topology of the upper tier FSO network is optimized. The objective is to maximize connectivity (and thus robustness) while meeting traffic demand and delay requirements. We presented a PSC algorithm for cluster formation and a GEA algorithm for topology optimization, and derived bounds on their performance. Simulation studies show that the algorithms are effective for the design and optimization of the tiered access network, as indicated by the closeness of their performance to the performance bounds.

Acknowledgments

This work is supported in part by the National Science Foundation under Grant CNS-1145446, and through the Broadband Wireless Access & Application Center (BWAC) site at Auburn University.

References

[1] I.K. Son, S. Mao, Design and optimization of a tiered wireless access network, in: Proceedings of IEEE INFOCOM'10, San Diego, CA, 2010, pp. 1–9.
[2] I.F. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: a survey, Computer Networks 47 (4) (2005) 445–487.
[3] J. Jun, M.L. Sichitiu, The nominal capacity of wireless mesh networks, IEEE Wireless Communications Magazine 10 (5) (2003) 8–14.
[4] J. Shi, O. Gurewitz, V. Mancuso, J. Camp, E.W. Knightly, Measurement and modeling of the origins of starvation in congestion controlled mesh networks, in: Proceedings of IEEE INFOCOM'08, Phoenix, AZ, 2008, pp. 1633–1641.
[5] A.A. Abbasi, M. Younis, A survey on clustering algorithms for wireless sensor networks, Computer Communications 30 (14/15) (2007) 2826–2841.

[6] B. Liu, Z. Liu, D. Towsley, On the capacity of hybrid wireless networks, in: Proceedings of IEEE INFOCOM'03, San Francisco, CA, 2003, pp. 1543–1552.
[7] C.C. Davis, I.I. Smolyaninov, S.D. Milner, Flexible optical wireless links and networks, IEEE Communications Magazine 41 (3) (2003) 51–57.
[8] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman & Co., New York, NY, 1990.
[9] B. Aoun, R. Boutaba, Y. Iraqi, G. Kenward, Gateway placement optimization in wireless mesh networks with QoS constraints, IEEE Journal on Selected Areas in Communications 24 (11) (2006) 2127–2136.
[10] D. Mosk-Aoyama, Maximum algebraic connectivity augmentation is NP-hard, Operations Research Letters 36 (6) (2008) 677–679.
[11] E. Korevaar, I.I. Kim, B. McArthur, Atmospheric propagation characteristics of highest importance to commercial free space optics, in: Proceedings of the International Congress of Mathematicians, Bulgaria, Cyprus, 2003, pp. 1–12.
[12] D.J. Heatley, D.R. Wisely, I. Neild, P. Cochrane, Optical wireless: the story so far, IEEE Communications Magazine 36 (12) (1998) 72–82.
[13] X. Zhu, J.M. Kahn, Free-space optical communication through atmospheric turbulence channels, IEEE Transactions on Communications 50 (8) (2002) 1293–1300.
[14] J.L. Gross, J. Yellen, Graph Theory and Its Applications, CRC Press, New York, NY, 1999.
[15] F.R.K. Chung, Spectral Graph Theory, American Mathematical Society, Providence, RI, 1997.
[16] G.W. Flake, R.E. Tarjan, K. Tsioutsiouliklis, Graph clustering and minimum cut trees, Internet Mathematics 1 (4) (2004) 385–408.
[17] B. Mohar, The Laplacian spectrum of graphs, in: Y. Alavi, G. Chartrand, O. Oellermann, A. Schwenk (Eds.), Graph Theory, Combinatorics, and Applications, Wiley, Hoboken, NJ, 1991, pp. 871–898.
[18] M. Fiedler, Algebraic connectivity of graphs, Czechoslovak Mathematical Journal 23 (2) (1973) 298–305.
[19] S. Boyd, Convex optimization of graph Laplacian eigenvalues, in: Proceedings of International Congress of Mathematicians, Spain, 2006, pp. 1311–1319.
[20] S. Bandyopadhyay, E.J. Coyle, An energy efficient hierarchical clustering algorithm for wireless sensor networks, in: Proceedings of IEEE INFOCOM'03, San Francisco, CA, 2003, pp. 1713–1723.
[21] J. Lian, K. Naik, G.B. Agnew, A framework for evaluating the performance of cluster algorithms for hierarchical networks, IEEE/ACM Transactions on Networking 15 (6) (2007) 1478–1489.
[22] M.N. Smadi, S.C. Ghosh, A.A. Farid, T.D. Todd, S. Hranilovic, Free-space optical gateway placement in hybrid wireless mesh networks, IEEE/OSA Journal of Lightwave Technology 27 (14) (2009) 2688–2697.
[23] D. Coppersmith, S. Winograd, Matrix multiplication via arithmetic progressions, in: Proceedings of ACM STOC'87, New York, NY, 1987, pp. 1–6.
[24] R. Jothi, B. Raghavachari, Degree-bounded minimum spanning trees, Discrete Applied Mathematics 157 (5) (2009) 960–970.
[25] I.S. Dhillon, B.N. Parlett, Orthogonal eigenvectors and relative gaps, SIAM Journal on Matrix Analysis and Applications 25 (3) (2003) 858–899.
[26] R.G. Gallager, P. Humblet, P. Spira, A distributed algorithm for minimum-weight spanning trees, ACM Transactions on Programming Languages and Systems 5 (1) (1983) 66–77.
[27] F. Liu, U. Vishkin, S. Milner, Bootstrapping free-space optical networks, IEEE Journal on Selected Areas in Communications 24 (12) (2006) 13–22.
[28] R.M. Krishnaswamy, K.N. Sivarajan, Design of logical topologies: a linear formulation for wavelength-routed optical networks with no wavelength changers, IEEE/ACM Transactions on Networking. 9 (2) (2001) 186–198.
[29] C. Ersoy, S.S. Panwar, Topological design of interconnected LAN-MAN networks, IEEE Journal on Selected Areas in Communications 11 (8) (1993) 1172–1182.
[30] R. Elbaum, M. Sidi, Topological design of local-area networks using genetic algorithms, IEEE/ACM Transactions on Networking 4 (5) (1996) 766–778.
[31] V.W. Chan, Free-space optical communications, IEEE/OSA Journal of Lightwave Technology 24 (12) (2006) 4750–4762.
[32] P.C. Gurumohan, J. Hui, Topology design for free space optical networks, in: Proceedings of IEEE ICCCN'03, Dallas, TX, 2003, pp. 576–579.
[33] S. Ghosh, K. Basu, S.K. Das, What a mesh! Architectures for next generation radio access networks, IEEE Network 19 (5) (2005) 35–42.
[34] S. Ghosh, K. Basu, S.K. Das, MeshUp: self-organizing mesh-based topologies for next generation radio access networks, Ad Hoc Networks 5 (6) (2007) 652–679.
[35] A. Desai, S. Milner, Autonomous reconfiguration in free-space optical sensor networks, IEEE Journal on Selected Areas in Communications 23 (8) (2005) 1556–1563.
[36] O. Awwad, A. Al-Fuqaha, B. Khan, G. Brahim, Topology control schema for better QoS in hybrid RF/FSO mesh networks, IEEE Transactions on Communications 60 (5) (2012) 1398–1406.
[37] M.N. Smadi, S.C. Ghosh, A.A. Farid, T.D. Todd, S. Hranilovic, Free-space optical gateway placement in hybrid wireless mesh networks, IEEE/OSA Journal of Lightwave Technology 27 (14) (2009) 2688–2697.
[38] N. Vaiopoulos, H. Sandalidis, D. Varoutas, WiMAX on FSO: outage probability analysis, IEEE Transactions on Communications 60 (10) (2012) 2789–2795.
[39] M. Karimi, M. Uysal, Novel adaptive transmission algorithms for Free-Space Optical links, IEEE Transactions on Communications 60 (12) (2012) 3808–3815.
[40] D. Zhou, P. LoPresti, H. Refai, Enlargement of beam coverage in FSO mobile network, IEEE/OSA Journal of Lightwave Technology 29 (10) (2011) 1583–1589.
[41] A.S. Reaz, V. Ramamurthi, M. Tornatore, S. Sarkar, D. Ghosal, B. Mukherjee, Cost-efficient design for higher capacity hybrid wireless-optical broadband access network (WOBAN), Computer Networks 55 (9) (2011) 2138–2149.
[42] A. Ghosh, S. Boyd, Growing well-connected graphs, in: Proceedings of IEEE CDC'06, San Diego, CA, 2006, pp. 6605–6611.
[43] A.S. Ibrahim, K.G. Seddik, K.R. Liu, Improving connectivity via relays deployment in wireless sensor networks, in: Proceedings of IEEE GLOBECOM'07, Washington, DC, 2007, pp. 1159–1163.