On the Design of Bi-connected Wireless Mesh Network Infrastructure with QoS Constraints

Djohara Benyamina*, Abdelhakim Hafid*, Michel Gendreau+

*NRL, University of Montreal, Canada
+CIRRELT, University of Montreal, Canada
{benyamid, ahafid, michel.gendreau}@iro.umontreal.ca

Abstract In the design of Wireless Mesh Networks (WMNs), one of the fundamental considerations is the reliability and availability of communication paths between network pairs in the presence of nodes failure. The reliability and deployment cost are important and are largely determined by network topology. Usually, network performance and reliability are considered separately. In this paper, we propose a new algorithm based on ear decomposition for constructing reliable WMN infrastructure that resists the failure of a single mesh node and ensures full coverage to all Mesh Clients (MCs). Via a case study, we show the tied relationship between network deployment cost, performance and reliability in a simultaneous optimization of cost and load balance over network channels. The optimization model proposed is solved using meta-heuristics which provides the network operator with a set of reliable tradeoff solutions.

Key Words: Wireless Mesh Network, Network design, Reliability, Multi-objective optimization, Cost-effective topology.

I. INTRODUCTION

Wireless Mesh Networks (WMNs) are highly reliable, scalable, adaptable and cost-effective which makes them the on-the-shelf coverage guaranteed wireless networks. Basically, WMNs consist of robust infrastructure of interconnected access points (APs), relays (MRs) and gateways (MGs). APs provide internet access to Mesh Clients (MCs) by forwarding aggregated traffic to MRs in a multi-hop fashion until a MG is reached. MGs act as bridges between the wireless infrastructure and the Internet. A simultaneous communications is possible over orthogonal channels if the communicating mesh nodes are equipped with multiple network interfaces (the case of Multi Radio Multi Channel, MR-MC, networks).

It is relatively easy to place such nodes to form a WMN infrastructure and to forward packets from sources (MCs) to destinations (MGs - Internet) or vice versa; however, it is very complex to achieve a desirable performance with this network while ensuring full coverage and reliable services to MCs under financial constraints. Moreover, when a router fails to route the traffic, an alternative path should exist to reroute the traffic. This is the well known survivable problem, which is composed of: 1) defining network restoration policy, and 2) designing robust (survivable) topology. In this paper, we consider the design of robust and cost-effective WMN infrastructure. To guarantee a reliable communication, extra nodes are placed to tolerate single router failure with acceptable overhead and without deteriorating the network performance. Indeed, we have to deploy the minimum number of mesh nodes, necessary to construct a single node fault-tolerant WMN infrastructure with QoS provision, such as full coverage and balanced load over network channels.

Existing WMNs design studies, focus mainly on performance improvement of a pre-deployed network (nodes’ locations and characteristics have already been decided). Some studies are related to network topology control design. The main goal of the topology control is to identify a subset of possible wireless links that provide connectivity for wireless networks, with certain design criteria. The authors in [1] focus on the design of a topology control scheme such that the overall throughput capacity can be maximized. Other studies deal with the construction of networks’ virtual backbones [2], [3]. The main objective of virtual backbone construction is to alleviate the Broadcasting Storm Problem by reducing the communication overhead and simplifying the connectivity management. In the same context (deployed networks), other studies were conducted with the purpose of providing optimized protocols [4], [5], [21].

Another category of studies consider topologies where only gateways or routers are located a priori [6], [9]. They propose techniques to place optimally either gateways or access points to satisfy some QoS constraints. From the bulk of contributions that address WMN design problem where the locations of all mesh nodes are not a priori decided [10][11], there is only one contribution [10] that considers network reliability in their proposed design approach. They define a reliability cost function that allows maximizing the reliability of WMNs. The proposed approach is based on iterative policy that is performed endlessly until a reliable and satisfactory solution (cost-effective solution) is found. However, in constraint optimization problem (e.g., WMN design problem) it is very difficult, if not impossible, to compute a good feasible solution that is also reliable when using an iterative design approach; the exception is when the approach/algorithm is proved to converge (within a finite number of iterations, the desired solution is obtained). Moreover, the reliability of the network has to be jointly considered with network QoS requirements while designing WMNs. Our focus in this paper is to construct reliable networks in the same time when designing cost-effective WMNs. Guaranteed robust infrastructures are obtained in a
finite number of iterations, providing multiple paths to gateways.

In this study, we consider the design of single-node fault tolerant WMNs. There are many protection schemes proposed either to prevent or to recover from node failures [12], [13]. Most of them are converted to survivable routing problems which suppose that a bi-connected network is already deployed. In this paper, we propose an efficient algorithm based on ear decomposition theoretical approach to construct a bi-connected WMN infrastructure. Such infrastructure is then able to accommodate the failure of one mesh node (the most common network failure scenario [14]). To the best of our knowledge, this paper is the first to address the design of bi-connected WMNs infrastructure from scratch, with a minimum number of mesh nodes. The main idea is to first place APs in a way to guarantee a full coverage to all MCS. Then construct, by augmentation, a small-sized connected set as a starting infrastructure and iteratively augment the infrastructure by adding new nodes to construct network ears. The algorithm terminates when all APs are included in the constructed ears. The resulted infrastructure is proved to be bi-connected, and can further be improved to meet performance and financial constraints.

The rest of the paper is organized as follows. Section II describes our network model. Section III presents the bi-connectivity construction algorithm. A case study of designing a reliable and cost-effective WMN infrastructure is given in Section IV. In Section V, we show the simulation results. Finally, we conclude the paper in Section VI.

II. NETWORK MODEL

We consider a multi-radio multi-channel WMN and we suppose initially that the mesh nodes operate using the same number of radios \( R \), each with \( k \) channels, \((k>R)\) and \( k \in C \), where \( C = \{1,...,c\} \) and \( c \) can be at most 12 orthogonal channels (if IEEE 802.11a is used installed, may establish a wireless communication with its eight direct-neighbors (Fig. 1.b). This assumption increases the chances of selecting a candidate neighbor among the eight with which a wireless link will be set up in the channel assignment procedure. The maximum degree of \( G \) denoted by \( \Delta \) is bounded by the number of radio interfaces, \( R \).

III. CONSTRUCTION OF A BI-CONNECTED INFRASTRUCTURE

In this section, we present a new algorithm for constructing a bi-connected WMN infrastructure. We first introduce some definitions then, we present the detailed algorithm.

A. Ear Decomposition: Definitions

The technique of ear decomposition has been successfully used in connectivity, bi-connectivity and outer-planarity testing problems [19], [20]. This technique has also been used to restore Wavelength Division Multiplexing (WDM) networks [17], [18] in order to find the protected routing paths.

Definition1: A connected graph \( G(V,E) \) is bi-connected if and only if \( G \) has an open ear decomposition.

Definition2: An ear decomposition \( P={P_0, P_1,...,P_k} \) of a graph \( G(V,E) \) is a partition of \( E \) into an ordered collection of edge disjoint simple paths \( P_0, P_1,...,P_k \). A cycle, and each end point of \( P_i \), \( i=1 \), is contained in some \( P_j, j<i \).

Definition3: An ear decomposition is open, if the endpoints of each \( P_i, i>1 \) are distinct.

The ear decomposition on a graph \( G(V,E) \) is a partition of the edges of the graph into a sequence of ears (simple paths and cycle(s)). Each ear starts by any unused edge from an already-explored vertex, and continue by a shortest path back to another already-explored vertex. The method is mainly based on spanning tree construction and Euler tour to complete the partition of the graph. The time complexity of the sequential algorithm for the ear decomposition problem [19] is \( O((m+n)\log n) \), where \( n=|V| \) and \( m=|E| \), while the parallel shared-memory algorithm complexity is \( O((n+m)/p \log n) \) [20], \( p \) is the number of processors.

Fig. 2 shows an example of ear decomposition on a bi-connected graph with eight vertices.

B. Construction of a Bi-connected Network

Compared to existing survivable network studies using ear decomposition, our approach is original. The primary difference is the way the ear decomposition concept is applied to obtain a bi-connected network. The usual use of ear decomposition is to check for survivable routing paths, assuming that the graph representing the network is bi-connected. We present, in this paper, a new algorithm, which starts from a disconnected graph and by the completion of the decomposition in ears; the graph becomes bi-connected.

![Fig. 1: WMN design problem. (a) network model, (b) WMN grid-like layout](image)

![Fig. 2: Example of ear decomposition. (a) The original graph G. (b) G is decomposed in 5 ears, each is shown in a different line style.](image)
In the following, we present the main steps involved in constructing a bi-connected WMN infrastructure from scratch. We pass over the details of placing APs in a way that guarantees full coverage to all MCs; we call this process coverage insurance (see details in Section IV). We suppose that the algorithm of constructing robust network starts from the dedicated (placed) APs which will automatically be a part of the final infrastructure.

Given an initial set of APs, located on a grid-like layout. Let \( P_k \) be the \( k \)th ear, \( N_k \) be the set of nodes in \( P_k \), and \( T_k \) be the set of edges in \( P_k \). We call the algorithm (algorithm 1) to construct a bi-connected network “BCN algorithm”, which consists of the following five main steps:

1) Use any traversal algorithm to connect, by adding new nodes, a leaf node to the closest neighbor node. We use a modified version of Breadth First Search algorithm (which we call ABFS), embedding an augmentation component, to obtain an initial small-sized connected graph. Fig. 3 illustrates the augmentation process.

2) Select a node \( v_0 \), with the highest degree, \( d \), to be the starting point. Apply the shortest path algorithm, starting from the source \( v_0 \) to find a minimal cycle. Let this cycle be the first ear \( P_0 \), \( k=0 \). If such a cycle does not exist, then apply ABFS to have the required nodes that compose the cycle and update \( G(V,E) \).

3) Update the degree of each node \( v_j \in N_k \) as follows: \( d(v_j) = d(v_j) - 1 \) and \( d(v_i) = d(v_i) - 1 \) for each edge \((v_j, v_i)\in T_k\). Let \( EP \) be the set of nodes, s.t :

\[
EP = \left\{ v_i \in \bigcup_{s \leq k} N_s \mid d(v_j) > 0 \right\}
\]

If there exist some APs not yet included in previous ears, then \( k=k+1 \) and continue looking for new ears (go to step 4).

4) Let \( SP \) be the set of all shortest paths with their endpoints pairs \( \in EP \) and internal nodes \( \notin EP \), as defined below.

\[
SP = \bigcup_{v_i,v_j \in EP} \text{ShortestPa} \text{ths}(v_i, v_j)
\]

If \( SP=\emptyset \) (disconnected nodes), then call ABFS to temporally add the necessary nodes for connectivity constraint and back to step 4.

5) A new ear \( P_k \) is defined as the longest path among the paths in \( SP \).

\[
P_k = \text{LongestPath} \left( h(l_i) \mid l_i \in SP, i \in N \right)
\]

From new nodes added in step 4), keep only nodes that are in \( N_k \). Update \( G(V,E) \) and go to step 3.

---

The algorithm terminates when all APs are included in ears. Fig. 4 depicts the steps 2-5.

**Lemma1**: The steps given above to construct network ears (steps 2 to 5) are not sufficient to construct a bi-connected network.

**Proof**: According to definitions 1 and 3, a graph is bi-connected if and only if the graph has open ear decomposition. However, the way ears are constructed in steps 3, 4 and 5 of BCN algorithm, may form cycles (excluding the starting cycle) and consequently the obtained ear decomposition may not be open. A new ear is formed, by choosing the start and end points as vertices occurring in previous ears and all other vertices in that ear should be new (not yet included in ears). If the start and end points overlap in one node to form a cycle, then the removal of that node will disconnect the graph since vertices of the newly formed ear are disconnected to vertices of previous ears (yellow node, see Fig. 4); this is the well known characteristic of a cut-vertex. Thus, we can then argue that the existence of inclusively more than one cycle proves the non bi-connectivity of the network.

---

A further enhancement of the algorithm to alleviate the problem posed by Lemma 1 is then required. The condition to only accepting distinct end-points of a composed ear \( P_k \), when constructing \( SP \), can be imposed. However, this condition may lead to premature termination of the algorithm. During the iterative execution of the algorithm, we may have a situation where \( |EP|=1 \). In this case, the algorithm will stop without further discovery of new ears; consequently, some APs are left unvisited (do not belong to any ear) and the resulted network is not robust. In Step 6, we complete the algorithm by proposing a simple solution that guarantees a fault-tolerant network without adding new nodes.

---

The terms: robust network, reliable network, fault-tolerant network and bi-connected infrastructure are used interchangeably in this paper.
6) For all cut-vertex $u_k$, elect two candidate nodes, that we call recovery-nodes $(u_k, u_l)$, as follows: $u_k$ is the direct neighbor of $u_k$ in $P_l$ and $u_l$ is the direct neighbor of $u_k$ in $P_{k-1}$ and $(u_k, u_l)$ are within the transmission range of each other.

The role of recovery-nodes pair, associated with every cut-vertex $u_k$, is to apply the segment protection policy [14] to recover the network from cut-vertex failures. In such protection policy, the segment between recovery-nodes is activated (bold segment in Fig.4), only when the failure occurs, consequently spare resources are saved, thus reducing channel interferences.

**Lemma 2**: The number of nodes added by BCN algorithm is optimal.

**Proof**: The total number of constructed ears is largely dependent on the number of network nodes. The cardinality of any ear decomposition $P$, for a bi-connected network $G(V,E)$, is equal to $m+n-1$ [19], where $m=|E|$ and $n=|V|$. When a new node is added, the number of wireless links is kept small enough to minimize the number of edges in $E$. This is achieved by assigning, for each added node, only one predecessor (a tree-like graph). The benefit of selecting the longest “shortest path” in step 5 is to construct a minimum number of ears which results in small number of nodes added to transform a connected graph to a bi-connected one. □

**Theorem 1**: Suppose $n$ (resp. $m$) is the number of nodes (resp. edges) of the final graph, the time complexity of constructing WMN robust infrastructure using BCN Algorithm is $O(n \cdot m \cdot \log n)$.

**Proof**: Let $n_0$ (resp. $m_0$) be the number of nodes (resp. edges) in the original graph (composed only with APs). Let $n_1$ (resp. $m_1$) be the number of nodes (edges) in the connected graph (result of step 1), and let $n$ (resp. $m$) be the number of nodes (resp. edges) in the final graph (after completion of BCN algorithm). We have the following relationship $n_0 < n_1 < n$ (resp. $m_0 < m_1 < m$). The time complexity of constructing a connected graph is $O(n_0^2)$. The number of leaf-nodes is at most $(n_0-1)$. Thus the first step needs $O(n_0^2)$ which is bounded by $O(n^3)$. The time complexity of finding the first cycle using shortest path algorithm, is $O(m + n \cdot \log n)$ bounded by $O(m^2 + n \cdot \log n)$. The running time of step 3 is $O(n)$.

Next, in step 4 and 5, checking for all shortest paths with augmentation process, takes at most $O(n^2 \cdot \log n)$. Finally step 6 has at most $O(n^3)$ running time. At the completion of Algorithm 1, we have $(m+n-1)$ constructed ears (see proof of lemma 2), thus the total running time of constructing WMN robust infrastructure using BCN Algorithm is $O(n \cdot m \cdot \log n)$. □

BCN algorithm is composed of two parts. The first part (step 1) consists of adding nodes to connect network APs; the outcome of the first part is then to have a connected network with few links\(^2\). The second part starts from step 2 to step 6 with the purpose of transforming the connected network into a robust (bi-connected) network. The BCN algorithm could also be run on a given infrastructure to enhance its reliability (step 1 is omitted). However, the number of added nodes in step 5 is not guaranteed to be optimal since it is highly dependent on the number of links in the original infrastructure.

**IV. Case Study: Simultaneous Optimization**

The purpose of BCN algorithm is to place a minimum number of mesh nodes that construct a robust WMN infrastructure and provide full coverage to network MCs. However, the resulting infrastructure after running BCN algorithm does not have one of the fundamental features of WMN which is its mesh topology (layout) due to the small number of links established when constructing the infrastructure. In fact, the BCN algorithm is run while designing the WMN infrastructure, where many links are added in when assigning channels (bounded by $R$), thus allowing robust infrastructure design while restoring all WMN characteristics with optimal number of mesh nodes. In this section, we will show how the resulting infrastructure is improved to meet other QoS constraints (i.e., load balance over network links) in a simultaneous optimization framework. Throughout this case study, we show the tied relationship between reliability, and design of cost-effective WMNs.

Let $I$ be the set of positions of traffic concentrations in the service area (Traffic Spots: TSs) and $L$ the set of positions where mesh nodes can be installed (Candidate Locations, CLs). From this section onward, we denote by $n$ the number of TSs and $m$ the number of CLs.

The WMN design problem aims at:

- Selecting a subset $S \subseteq L$ of CLs where a mesh node should be installed so that the signal level is high enough to cover the considered TSs.
- Defining the gateway set by selecting a subset $G \subseteq L$ of CLs where the wireless connectivity is assured so that all traffic generated by TSs can find its way to reach a node in $G$.
- Maintaining the cardinalities of $G$ and $S$ small enough to satisfy the financial and performance requirements of the network planner.

In order to describe the problem formally we introduce the following notation -

Let $I = \{1, ..., n\}$ and $L = \{1, ..., m\}$. In the following, unless otherwise stated, $i$ and $j$ belong to $I$ and $L$ respectively. The rest of the notation used in the problem formulation is summarized in TABLE I.

<table>
<thead>
<tr>
<th>TABLE I: List of Main Parameters/Var. Used in Model Formulation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Param./Var.</strong></td>
</tr>
<tr>
<td>$n$</td>
</tr>
<tr>
<td>$m$</td>
</tr>
<tr>
<td>$d_i$</td>
</tr>
<tr>
<td>$u_{ij}$</td>
</tr>
<tr>
<td>$v_j$</td>
</tr>
<tr>
<td>$c_i$</td>
</tr>
<tr>
<td>$p_i$</td>
</tr>
<tr>
<td>$R$</td>
</tr>
<tr>
<td>$k$</td>
</tr>
<tr>
<td>$a_{ij}$</td>
</tr>
<tr>
<td>$b_{ij}$</td>
</tr>
<tr>
<td>$t_i$</td>
</tr>
<tr>
<td>$g_j$</td>
</tr>
</tbody>
</table>

---

\(^2\) The number of links when connecting the network is kept small enough, to construct a minimum number of ears and thereafter to place a minimum number of nodes that are required to make the infrastructure robust.
Load balancing is a desirable feature to have in a wireless mesh network. It reduces congestion in the network, increases network throughput, and prevents service disruption in case of failure [21]. We formulate the WMN design problem as a multi-objective optimization model. WMN planning solutions under multi-objective approach are more realistic and much preferred by network planners in that they have to be cost-effective (the deployment cost is minimized while the throughput is maximized by balancing loads over network channels). The formulation is given below.

\[
\min \sum_{j \in L} \left( c_j t_j + p_j g_j \right) \\
\left( \sum_{j \in L} \sum_{l \in L} \sum_{qc \in C} \left( \frac{f_{jl}^q}{u_{jl}} \right)^2 \right) \\
\left( \sum_{j \in L} \sum_{l \in L} \sum_{qc \in C} \frac{f_{jl}^q}{u_{jl}} \right)
\]

Subject to:
\[
\sum_{j \in L} x_{ij} = 1 \quad \forall i \in I \quad (3)
\]
\[
x_{ij} \leq a_{ij} t_j \quad \forall i \in I, \forall j \in L \quad (4)
\]
\[
\sum_{i \in I} d_{i} x_{ij} + \sum_{l \in L} \sum_{qc \in C} \left( f_{jl}^q - f_{jl}^q \right) - F_j = 0 \quad \forall j \in L \quad (5)
\]
\[
\sum_{k, l, N_{ij}} y_{jk}^q \leq 1 \quad \forall q \in C, \forall j, l \in L \quad (6)
\]
\[
f_{jl}^q \leq y_{jl}^q \quad \forall q \in C, \forall j, l \in L \quad (7)
\]
\[
\sum_{i \in I} d_{i} x_{ij} \leq v_j \quad \forall j \in L \quad (8)
\]
\[
F_j \leq M g_j \quad \forall j \in L \quad (9)
\]
\[
2 y_{jl}^q \leq b_{jl} \left( z_j^q + z_j^q \right) \quad \forall q \in C, \forall j, l \in L \quad (10)
\]
\[
g_{jl} \leq t_j \quad \forall j \in L \quad (11)
\]
\[
\sum_{l \in L} y_{jl}^q \leq 1 \quad \forall q \in C, \forall j \in L \quad (12)
\]
\[
\sum_{q \in C} z_j^q \leq R t_j \quad \forall j \in L \quad (13)
\]

In this model, the objective function (1) minimizes the total cost of the network including installation cost $c_j$ and additional gateway installation cost $p_j$. The load-balanced objective function (2) is the minimization of the standard deviation of the ratio of traffic flows over the network links. Constraint (3) and Constraint (4) assign a TS $x_{ij}$ to an Access Point (AP) installed at location CL$_i$; Constraint (3) makes sure that the TS, is assigned to exactly one and only one AP installed at CL$_i$, while constraint (4) implies that the TS, and the assigned AP are within the coverage area. Constraint (5) defines the flow balance for each mesh node at CL$_j$. Constraint (6) limits link interferences, while inequalities (7) and (8) respectively define the flow-link capacity and the demand-radio access capacity constraints. Constraint (9) stipulates that the flow routed to the wired backbone is different from zero only when the installed mesh node is a gateway. We assign $M$ a very large number to limit the capacity of the installed gateway. Constraint (10) forces a link between CL$_j$ and CL$_l$ using the same channel $q$ to exist only when the two devices are installed, wirelessly connected and tuned to the same channel $q$. Constraint (11) ensures that a device can be a gateway only if it is installed. Constraint (12) prevents a mesh node from selecting the same channel more than once to assign it to its interfaces. Constraint (13) states that the number of links emanating from a node is limited by the number of its radio interfaces; it also states that if a channel is assigned only once to a mesh node, it is a sufficient condition for its existence.

All the above constraints are called hard constraints with the exception of constraint (5) which is then called a soft constraint. To construct fault-tolerant topologies, we add two more constraints to guarantee two disjoint paths based networks.

\[
\sum_{l \in L} \sum_{q \in C} y_{jl}^q \geq 2 \quad \forall j \in L \quad (16)
\]
\[
\sum_{l \in L} \sum_{q \in C} y_{jl}^q \leq 1 \quad \forall j \in L - G \quad (17)
\]

There must be at least two node-disjoint paths from each node $j \in S$ to some gateway in $G$, as shown by constraint (16). The idea we use to ensure node disjoint paths is that for a node $j$, there can be at most one incoming flow originating from node $l \in S$ (Equation 17). It is not hard to see that this condition is sufficient and necessary to guarantee vertex disjoint paths. Note that (17) automatically prevents the flows from forming cycles at the intermediate nodes on the way to the gateway(s).

### A. Solving the WMN Design Problem

WMN design is a fairly complex problem; its difficulty lies in the fact that it tries to simultaneously address many criteria (i.e., minimize deployment cost, maximize throughput by balancing loads, full coverage to MCs and, robust infrastructure with minimum nodes installed). Joint optimization of the above criteria is defined as a Multi-Objective search Problem (MOP). Solving a MOP returns a...
set of Pareto optimal solutions. Each solution represents a different trade-off between the objectives that is said to be “non-dominated”. We use a variant of MOPSO, Multi-Objective Particle Swarm Optimization, as the optimization technique [22], [23] to solve the model. We use the hyper-grid approach for maintaining diversity and we incorporate a mutation factor to boost the exploration capability of the standard MOPSO [23]. More details on how a multi-objective generic model is solved using MOPSO can be found in [24].

A particle in the swarm is a position in the search space; it represents a set of assignments that is a solution to the problem. In our case, each position (particle) provides information about user connectivity ($x_{ij}$), device installation ($t_j$) and ($z_{ij}^{+}$), device connectivity ($y_{ij}^{+}$), gateway existence ($g_j$), link flows ($f_{ij}^h$), and gateway/backbone link flows ($F_i$). We consider a feasible solution a solution that satisfies all hard and soft constraints. During the search, non-feasible solutions that violate only the soft constraint (7) can be included in the population. This increases the likelihood of a non-feasible solution to mutate and provide a feasible one in later generations. The followings are the phases involved in the resolution of the proposed model.

First, feasible planning solutions are constructed (Fig. 5); i.e., place, for each particle in the swarm, a subset $S_1$ of APs to cover all TSs (coverage insurance design stage that satisfy Constraints 3 and 4), invoke $BCN$ algorithm, given in Section III, to construct the bi-connected infrastructure, place gateways, assign channels as in [25] and assign flows using Edmond’s algorithm [26]. All remaining constraints (i.e., 5-17) are then satisfied.

![Fig. 5: A Feasible Particle Position: (a) TSs assigned. (b) $S_1$ augmented, MGs selected.](image)

Then, the main algorithm loop (Algorithm 2) is invoked; based on the initial feasible solutions and $BCN$ algorithm, it goes on mutating planning solutions in the swarm from generation to generation, with a bias towards selecting the fittest solutions.

**Algorithm 2: Planning Resolution**

Input : MaxGeneration, $f_{Mut}$.
Output : REP, Repository

Construct_Initial_Soft&Hard_feasible_solutions ()

t=0

while (t<MaxGeneration)

for each particle in the swarm

$S_1$< Mutate($S_1$, $f_{Mut}$)

$S$< Augment($S_1$)

Y1< Construct_Bi-Connected_infrastructure() /* $BCN$ algorithm*/

Y< Assign_channels(Y1)

G< Select_getways()

Compute_flows()

Construct_New_Particle()
endfor

Compute_Velocities
Update_Positions()
Evaluate_Particles
REP< Insert_feasibleNonDominated_Solutions()
Update_ParticuleBest

During the exploration of the search space, each particle has access to two pieces of information: the best Potential Solution (PS) that it has encountered and the best PS encountered by its neighbors. This information is used to direct the search by computing velocities: velocity,$[d] = iw *$ velocity,$[d] + rand * (pBest$[d] – position$[d]) + rand*$ (REP$[h] – position$[d]), where rand is a random number in the range of [0,1]. The index $h$ for the REP is chosen randomly from the selected hypercube, iw is the inertia weight. A large inertia value will cause the particles to explore more of the search space, while small one directs the particles to a more refined region. The importance of inertia weight was pointed out by Shi and Eberhart [28] who reported that 0.4 is the best value. A position in the search space is a solution to our planning problem; however, the values, returned by Update_Positions() procedure, are not guaranteed to be integers (0 or 1). For that reason we do extra processing to encode these values and make sure that the retained positions are integers.

**V. PERFORMANCE EVALUATION**

In this section, we evaluate the performance$^3$ of Algorithm 2 (coupled with $BCN$ algorithm) under many deployment scenarios. We carry out four set of simulations, where we vary one key parameter at a time ($m$, $n$, $R$, $d$) while maintaining others fixed. We define the Standard Setting (SS) of the WMN as the following: SS=[{(n:150), (m:49), (d:2Mb/s), (c:54Mb/s), (R:3), (k:11)}]. The algorithm is coded in the Java programming language and all the experiments were carried out on a Pentium M 1.5 GHz. Unless stated otherwise, we use the standard setting SS. The positions of the n TSs are randomly generated. A run of our main algorithm (Algorithm 2) involves 100 generations each with a population size and archive size of 50 and 30 particles respectively.

We take a mutation factor (fmut) of 0.5 as our standard setting based on our recent experiments [27] (mutating at a rate of 50% of the population leads to the best Pareto front). Results are reported after 10 runs. Additional filtering process is required to maintain the non-dominance aspect of the collected Pareto fronts. For each simulation set we plot the resources utilization (APs, Total nodes added by $BCN$ algorithm, Links and, MGs) where only cheapest solutions are considered. We mean by added nodes, the number of nodes required to connect the network and additional nodes necessary to bi-connect the infrastructure (i.e., actual number of relays and gateways composing the final infrastructure; see Fig. 5.b). The number of added nodes on connected infrastructure to make it robust is not shown, as the design and the construction of the infrastructure are done from scratch. We also plot optimal planning solutions in objective space graph (Pareto optimal solutions).

$^3$ The $BCN$ algorithm is invoked by Algorithm 2 to design robust WMN infrastructure upon the aforementioned multiple criteria satisfaction. The evaluation of $BCN$ algorithm could not be done solely without evaluating design solutions after running Algorithm 2.
A. Effect of varying number of candidate locations m.

In this set of simulations, we perform four different experiments by varying the grid-size \((m=49, 64, 81, 100)\). All Pareto optimal solutions found are plotted in the same graph. Fig.6.b shows that the best Pareto front is obtained when \(m=49\), and the grid-size of 7x7 is largely sufficient to construct a robust and cost effective WMN infrastructures given the standard setting SS.

As expected, the increase in the number of candidate locations leads to an increase in the number of added nodes, the number of links and the number of gateways (Fig.6.a). The number of APs remains relatively stable, since neither the number of MCs nor the traffic demand change. The first reason behind the increase of network resources (except APs) is that increasing the number of CLs increases the probability of a MC not being connected to Internet through a multi-hop wireless path (disconnected mesh nodes), leading to install more nodes and establishing more wireless links to satisfy connectivity constraints. Even if the network is connected, additional nodes and links are also added in order to construct a robust infrastructure, which is performed by BCN algorithm.

B. Effect of changing the number of mesh clients n.

We also study how our algorithm would behave when the number of MCs varies. Notice that the remarkable increase in the number of deployed APs (see Fig. 7.a) is more related, in the first place, to the increase of users that need to be covered and connected to Internet, then in the second place, to fulfill the load balance requirement. Compared to the number of added nodes and links, the number of added gateways is not significant (at most one gateway is added for every 50 new MCs). The main reason behind this noticeable gain is that diverse disjoint paths are available to connect all MCs to Internet through MGs (robust infrastructure), hence deploying few gateways are enough to continue providing reliable services to MCs. As shown by Fig.7.b, the more MCs are added the more network planner has to pay for robust topologies that are load-balance guaranteed.

C. Effect of changing the number of radio interfaces R.

Another performed endeavor to evaluate the performance of our algorithm consists of varying the number of radio interfaces \(R\) from 2 to 5. A slight decrease in the number of gateways and APs followed by a significant decrease in the number of added nodes is noticed when \(R\) shifts from 2 to 3 (Fig. 8.a). However, when \(R\) shifts from 3 to 4 then from 4 to 5, the number of gateways remains fixe, while the number of added nodes to construct a robust infrastructure, increases. This can be explained by interferences caused when the number of wireless links increases, which leads to place new nodes so that alternative paths could be found to route the traffic. Notice also, from Fig. 8.b, that the best Pareto front is obtained with four radio interfaces instead of five. From the above, we can then stipulate that the performance and robustness of a WMN infrastructure is better achieved when \(R=3\) (less interferences) or \(R=4\) (best tradeoff solutions), and no additional gain is obtained when more than four radio interfaces are deployed.

D. Impact of demand variation.

The last set of simulations carried out to evaluate our approach, consists of gradually increasing traffic demand \(d_i\) from 1 to 5 Mb/s. Fig.9.a shows that the number of network resources (APs, added nodes, MGs, and Links) are linearly dependant on the traffic demand the network has to support, which is as expected. However, the increase in the number of MGs seems following the same pattern as \(d_i\) variation (for one Mb/s of more traffic, only one MG is added in). This fact could be explained by the benefit of designing a biconnected network, where alternative paths are available to reroute the traffic to accessable MGs. Fig.9.b shows that the less \(d_i\) is, the more the load balance is enhanced, and the less the deployment cost is, which is obvious.
Fig. 9: Impact of varying traffic demand. (a) Resources utilization. (b) Pareto Optimal solutions.

VI CONCLUSION

In this work, we address one of the fundamental problems for designing a WMN: how to construct a robust network while designing the WMN infrastructure under QoS and financial constraints? We devise a new approach based on ear decomposition to construct a robust infrastructure with a minimum number of nodes. Through a case study, where we formulate the WMN design as a simultaneous optimization of deployment cost and load balance objectives, we show the impact and benefits of our devised approach in designing cost-effective and robust WMN infrastructures. In the light of the results obtained in Section V, a spectrum of alternative trade-off solutions is provided to the network planner allowing a flexible decision making. It has been shown that the key parameter \( R \) is crucial in determining the cost effectiveness of the produced infrastructure. Even if the resulted infrastructure is robust, selecting a large value of \( R (R>4) \) may deteriorate the performance of the network and may increase the total number of mesh nodes forcing the network planner to pay more (expensive topologies) for unworthy networks. Varying the number of MCs (\( n \)) and Traffic demand (\( d_i \)), have proved the effectiveness and scalability of our approach in designing robust and cost-effective WMN infrastructures. Next we will investigate the impact of limiting the number of communication hops while designing robust WMNs.

REFERENCES


