Optimizing Complex Cluster Formation in MANETs Using SAT/ILP Techniques

Syed Zohaib Hussain Zahidi, Fadi Aloul, *Member, IEEE*, Assim Sagahyroon, and Wassim El-Hajj

***Abstract*— Over the course of the last decade, there have been several improvements in the performance of Integer Linear Programming (ILP) and Boolean Satisfiability (SAT) solvers. These improvements have encouraged the application of SAT and ILP techniques in modeling complex engineering problems. One such problem is the Clustering Problem in Mobile Ad-Hoc Networks (MANETs). The Clustering Problem in MANETs consists of selecting the most suitable nodes of a given MANET topology as clusterheads, and ensuring that regular nodes are connected to clusterheads such that the lifetime of the network is maximized. This paper proposes the development of an improved ILP formulation of the Clustering Problem. Additionally, various enhancements are implemented in the form of extensions to the improved formulation, including the establishment of intra-cluster communication, multihop connections and the enforcement of coverage constraints. The improved formulation and enhancements are implemented in a tool designed to visually create network topologies and cluster them using state-of-the art Generic ILP and SAT solvers. Through this tool, feasibility of using the proposed formulation and enhancements in a real-life practical environment is assessed. It is observed that the Generic ILP solvers, CPLEX, and SCIP, are able to handle large network topologies, while the 0–1 SAT-based ILP solver, BSOLO, is effective at handling the smaller scale networks. It is also observed that while these enhanced formulations enable the generation of complex network solutions, and are suitable for small scale networks, the time taken to generate the corresponding solution does not meet the strict requirements of a practical environment.**

***Index Terms*— Boolean satisfiability (SAT), integer linear programming, mobile ad-hoc networks (MANETs), optimization.**

I. INTRODUCTION

OVER the past decade, Integer Linear Programming (ILP) and Boolean Satisfiability (SAT) solvers have improved significantly through the introduction of new intelligent algorithms that allowed the solvers to handle a wider range of challenging Engineering problems. While generic-based ILP solvers have been applied to solving ILP models of several real-life optimization problems, relatively few similar attempts have been made using SAT solvers. One such problem is the *clustering problem* in Mobile Ad-Hoc Networks (MANETs).

MANETs are used in a wide-range of applications such as battlefield communication, law enforcement operations, and disaster recovery [1]. *Clustering* in MANETs has been traditionally proposed to solve the scalability issue in *flat* MANETs and prolong the lifetimeof the network. Clustering involves the creation of a hierarchical network where the network is divided into clusters, with certain nodes in each cluster being chosen as *clusterheads*. The process of selecting which nodes would be best suited to be clusterheads and which regular nodes should be assigned (connected) to which clusterhead is known as the *clustering problem*. The clustering problem can be modeled as an optimization problem in ILP.

It is important to note that the suggested ILP formulations and their solutions are not intended to present a solution to the clustering problem in real time scenarios. However, the benefit of the work consists of 4 key areas: 1) formulating many applicable variations of the clustering problem in MANETs, taking into account realistic restrictions, which presents a strong mathematical foundation to the clustering problem; 2) presenting an optimal framework that allows other researchers to suggest smart heuristics and meta-heuristics and compare their results with the optimal ones we provide; 3) introducing an extensive comparison and comprehensive performance evaluation of the state-of-the-art ILP (generic-based and 0–1 SAT-based) solvers in handling the proposed ILP formulations and suggesting which one works best under what circumstances; and 4) the design and development of an intuitive environment which allows customized topology creation, integration with Generic ILP and SAT-based ILP solvers, and the ability to view visual representations of the solutions of the ILP formulations as a network topology.

This paper is organized as follows. Section II presents background information on ILP, SAT and its applications. It also introduces MANETs and defines the clustering problem. Section III covers the existing work done in the use of ILP formulation in modeling the clustering problem. Section IV describes the proposed ILP formulation of the clustering problem in MANETs and details the proposed enhancements including enabling Intra-Cluster communication, Multihop Connections and Coverage restrictions. Section V presents the conducted experiments and the corresponding results and conclusions. Section IV concludes this paper and presents future work.

II. BACKGROUND

A. Integer Linear Programming and Boolean Satisfiability

Integer Linear Programming (ILP) involves maximizing or minimizing a function with respect to certain constraints

Manuscript received January 31, 2013; accepted March 10, 2013. Date of publication March 21, 2013; date of current version April 30, 2013. The associate editor coordinating the review of this paper and approving it for publication was Prof. Paul C.-P. Chao.

S. Zahidi, F. Aloul, and A. Sagahyroon are with the American University of Sharjah, Sharjah 26666, UAE (e-mail: zohaib44@gmail.com; faloul@aus.edu; asagahyroon@aus.edu).

W. El-Hajj is with the American University of Beirut, Beirut 1107 2020, Lebanon (e-mail: we07@aub.edu.lb).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSEN.2013.2254234

where the optimal function and constraints are linear and the used variables can only take integer values [2]. Cases where the integer values are restricted to (0–1) are called Binary ILP Problems. In SAT the constraints between variables are represented using what is called propositional logic. Propositional logic involves the use of AND, OR and NOT operations to construct formulas in the Products-of-Sums form or Conjunctive Normal Form (CNF). The variables can only take Binary values (0–1).

Given constraints expressed in CNF, the goal is to identify a variable assignment that will satisfy all constraints in the problem or prove that no such assignment exists. In a propositional formula, given n variables, there are 2^n different possible variable assignments. In order to *solve* or rather *satisfy* the formula, SAT will go through the search space and determine whether or not there is a satisfying variable assignment. Advanced decision heuristics and intelligent conflict diagnosis techniques can be used to avoid searching through the entire tree of 2^n assignments.

While SAT solvers have traditionally been used to solve *decision* problems, recently SAT solvers have been extended to handle pseudo-Boolean (PB) constraints [3], [4] which are simple inequalities that are equivalent to 0–1 ILP constraints. PB constraints can replace an exponential number of CNF constraints. Another key advantage of PB constraints is the ability to express *optimization* problems which were traditionally handled as ILP problems. Studies have shown that 0–1 SAT-based ILP solvers can compete with the best available generic-based ILP solvers in solving 0–1 ILP problems arising in specific applications [3], [4]. The recent advances in SAT solvers as well as the availability of increasingly affordable high computational power, have allowed larger problem instances to be solved in different applications domains. Such applications include Power Optimization [5], FPGA [6], Network Intrusion [7], Access Control [8], Cryptography [9], Application Mapping [10], Genetics [11] and Scheduling [12].

B. Mobile Ad-Hoc Networks and the Clustering Problem

MANETs are wireless, self-organizing networks consisting of mobile nodes with generally a limited supply/store of energy. These nodes can be for example, laptops, mobile radio terminals or other devices, generally those which are used by humans [13]. There are several challenges faced in enabling MANETs to communicate through a stable, scalable, flexible topology. Over the years much research has been undertaken in enabling MANETS to operate in the optimum state, i.e., minimizing energy consumption and essentially attempting to achieve the maximum network lifetime through optimizing cluster formation, routing and communication. Initially MANET topologies were flat networks or non-hierarchical networks where all nodes had identical roles. Through various tests and simulations scenarios, it was shown that as the number of nodes in flat networks increases, the throughput falls drastically [14].

In addition several, factors such as frequent route breakage, unpredictable topology changes, and routing overhead make it difficult for a flat topology to be scalable [15]. The concept of clustering was introduced to overcome the scalability limitations of a flat network. Clustering involves dividing the network into clusters with certain nodes in each cluster being chosen to be clusterheads. The clusterheads have the responsibility of managing communication and routing for their particular cluster. Consequently, the selection of clusterheads is particularly important [16]. Through this hierarchical configuration of the network, clustering also has the advantage of reducing computational complexity of the underlying network and mitigating the effects of mobility by making a mobile topology “appear” relatively static. Of particular importance are constraints related to network connectivity and energy conservation [17]. Clustering also has the advantage of reducing the information storage overhead for regular nodes as nodes need to be aware of “local” changes (changes in the same cluster), and not global changes (changes occurring in other clusters) [18].

In addition to achieving improved scalability over “flat” networks’, clustering also enables conservation of communication bandwidth as inter-cluster communication is restricted to clusterheads. It may also be possible for clusterheads to implement localized optimization strategies which enable nodes within a cluster to achieve optimal lifetime. In applications such as Wireless Sensor Networks (WSNs) where sensors may have overlapping coverage areas, clusterheads can assign certain nodes within their cluster to be “active” or to “sleep” thereby improving network lifetime and reducing duplication of data gathered [19].

In [18], the authors provide a breakdown of different clustering methodologies and techniques, including.

1) Dominating Set (DS)-based clustering: Emphasis on reducing routing complexity.
2) Low-maintenance clustering: Emphasis on minimizing the cost of cluster maintenance.
3) Mobility-aware clustering: Emphasis on incorporating mobility behavior of nodes into the clustering process in order to increase cluster stability and maximize the time before re-clustering is required.
4) Energy-efficient clustering: Emphasis on achieving maximum network lifetime.
5) Load-balancing clustering: Emphasis on equal distribution of load between clusterheads.
6) Combined-metrics-based clustering: Emphasis on optimizing network lifetime by incorporating multiple metrics that affect clusterhead selection such as initial energy, residual energy, traffic, and mobility.

Depending on the nature of the application, clustering algorithms may also focus on other factors that also require optimization [19].

1) Minimal cluster count: Emphasis on minimizing the number of clusters given that in heterogeneous networks there exist powerful nodes that are more suited to being clusterheads, yet, such nodes are expensive and should be deployed in limited quantities. The focus is minimizing the number of clusters (or minimizing cost) and maximizing the network lifetime.

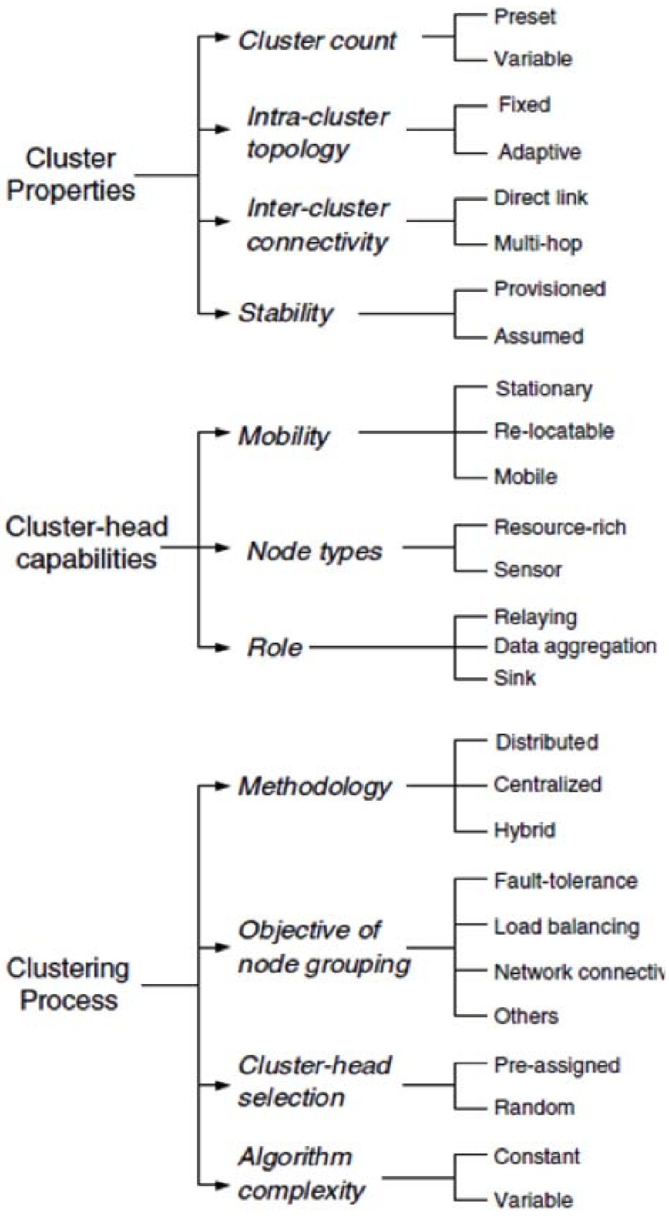


Fig. 1. Cluster properties, clusterhead capabilities and clustering process attributes to be considered [19].

2) Increased connectivity: Emphasis on ensuring that all network nodes have available paths, through their clusterheads, to a sink or Base Station. This property is crucial in WSNs where every node must have an efficient route to a predetermined sink.

Since the clusterhead is mainly responsible for routing, managing its cluster, relaying messages from/to other clusters, etc., its residual energy depletes faster than the other nodes. Hence, choosing another clusterhead to manage the cluster or sometimes re-clustering the whole network is a needed operation. Another reason for re-clustering might be caused by the mobility of the nodes (or clusterheads) where some nodes may move out of range of one clusterhead and in range of another and therefore the topology must adjust accordingly. As a result, re-clustering becomes an important factor in the clustering optimization problem for the goal of achieving fault-tolerance. Re-clustering however is resource intensive and introduces disruptions in the network. Therefore, in cases where a high performance fault-tolerant configuration is required, it is preferable to include the election of backup clusterheads during the clustering process.

The authors in [19] provide a comprehensive listing of variables that must be considered when clustering the network and the different factors that are considered in the clustering process (Fig. 1).

III. EXISTING ILP FORMULATIONS OF THE CLUSTERING PROBLEM

Research in wireless networks and optimization of wireless network operations involved using the application of ILP in order to find the optimal solution to several problems. One example of such a problem is the “minimum-energy multicast” (MEM) problem [20] in wireless ad hoc networks with omnidirectional antennas. Another example is the “minimum transmission broadcast” (MTB) problem [21] where the objective is to avoid the broadcasting storm problem, through the development of a broadcast scheme with the minimum number of transmissions.

However, ILP formulations of the clustering problem in MANETs are limited. The earliest contribution, to the best of our knowledge, can be traced to 2004, where the authors in [15] formulated a non-ILP algorithm clustering mechanism called Virtual Grid Architecture (VGA) and a corresponding ILP formulation to use as a basis for comparison. The objective of the ILP formulation was to find the minimum set of connected clusterheads. The topologies generated were 1-hop, and due to the capabilities of the solver, the network size was limited to 30 nodes when comparing the performance of VGA to ILP. Solving the ILP formulation for their 30 node network using CPLEX took 1011.5 s (∼17 min). Test results showed that topologies generated through the ILP formulation had fewer clusterheads as compared to those generated with VGA. While this paper put forward by the authors in [15] could be considered the first attempt at using ILP formulation in relation to the clustering problem, the first truly significant attempt at applying ILP formulation to the clustering problem was the work put forward by the authors in [22] in 2006. Unlike the model presented in [15], the authors did not focus on obtaining the minimum number of clusterheads but rather focused on the selection of a specified number of clusterheads, the interconnection of regular nodes and clusterheads, and the interconnection of clusterheads in a backbone, such that a specified maximum cluster size was not exceeded, and such that the maximum possible network lifetime was obtained.

The authors, in [22], proposed three different ILP formulations, each with a different approach to the creation of a backbone. The first formulation, Energy Efficient Clustering—Fully Connected Backbone (EEC-FCB), involved connecting the backbone of selected clusterheads through a mesh topology. The second formulation, Energy Efficient Clustering—Connected Backbone (EEC-CB), relaxed the constraints requiring mesh interconnectivity of the backbone of

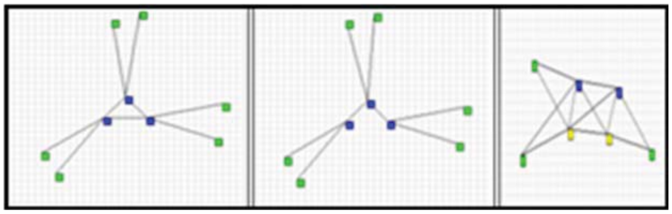


Fig. 2. Different network topologies: fully connected backbone, connected backbone, and redundant models [22].

clusterheads, thereby reducing the number of redundant connections.

The third formulation, Energy Efficient Clustering (EEC-R), formulated the application of a backup clusterhead for each selected clusterhead. Fig. 2 shows the topologies generated by the three formulations.

The EEC-FCB model produced too many redundant links in the backbone, particularly when generating configurations with a large number of clusterheads. The EEC-CB model used a Master Clusterhead (MCH) which reduced the number of redundant links but introduced a possibility of the MCH being a central point of failure. Due to the complexity of the ILP formulations and the limitations of the generic-based ILP solver used, only ILP formulations of networks with up to 9 nodes could be solved. The proposed formulations did not undergo significant testing with a variety of generic-based ILP solvers. Additionally, the coverage radius of nodes was not considered. It was assumed that all nodes could communicate with each other. Nevertheless, this work represented the first significant ILP formulation of the clustering problem, and provided a platform to enhance significantly.

ILP was also applied in the optimization of Wireless Sensor Network's (WSN) lifetime [23] and sensor's localization and coverage [24]. The authors in [13] detail the differences between MANETs and WSNs.

IV. PROPOSED ILP FORMULATION OF THE CLUSTERING PROBLEM

This paper proposes an ILP formulation of the clustering problem, building on the ideas and assumptions put forward in the EEC-CB model presented in [22]. This model improves on weaknesses present in the EEC-CB model and adds redundancy through the use of a *Star-Ring* backbone. Additionally, a proposed enhancement allows coverage to be taken into account.

A. Proposed Base Model

The variables used in [22] are maintained as follows:

$-N$: Total number of nodes in the network (predetermined)

$-P$: Number of clusters heads (predetermined)

$-d_{ij}$: Euclidean distance between nodes i and j

$-K$: Max number of nodes that can be connected to a CH (predetermined)

$-c_{ij}$: Cost of connecting a regular node i to CH j (proportional to d_{ij}^2)

$-h_{jk}$: Cost of connecting CH j to CH k (proportional to d_{jk}^3)

$-x_{ij}$: Variable. 1 if node i is connected to CH j or if node j is connected to CH i; 0 otherwise

$-z_{ij}$: Variable. 1 if CH i is connected to CH j; 0 otherwise

$-y_j$: Variable. 1 if node j is chosen to be a CH; 0 otherwise

$-M_j$: Variable. 1 if node j is a Master CH; 0 otherwise

$-w_{ij}$: Variable. 1 if $x_{ij} = 1$ and $y_j = 1$; 0 otherwise.

$-b_j$: Weight associated with CH j.

The following assumptions which were made in the ILP formulations in [22] are also applicable to the proposed ILP formulation. The variable b, in the objective function, which represents the level of the node's capability to act as a clusterhead, gets its value from an external source (algorithm, tool, etc). This is useful as multiple approaches/algorithms, which determine the suitability of a node in acting as a clusterhead, can be combined with this model without changing the equations, although this is out of the scope of our research. It is assumed that nodes are able to determine each other's position (Example: through the use of GPS).

Mobility is taken into account as it is incorporated in the cost matrices (c and b). A cost value in these matrices can be the aggregation of Mobility, Residual Energy and Traffic as suggested in [25] and [26]. Equation 1 is the objective function to be minimized. The structure of the objective function is kept similar to the one used in the EEC-FCB and EEC-CB models in [22]

$$Min : \left(\sum_{i=1}^{N} \sum_{j=1}^{N} c_{i,j} x_{i,j} + \sum_{j=1}^{N} b_j y_j + \sum_{j=1}^{N} b_j M_j + \sum_{j=1}^{N} \sum_{k=1}^{N} h_{j,k} z_{j,k} \right). \quad (1)$$

The first term in the objective function represents the connections between nodes and clusterheads. The second term represents the selection of nodes to be clusterheads. In the proposed formulation, the MCH is not a regular clusterhead and therefore needs to have its own term in the objective function so that its cost is taken into account when designing the network. This is represented in the third term of Equation 1. The last term represents the connections between clusterheads. The objective function aims to minimize the cost of sending/receiving data along these connections. The constraints used are described below. It is to be noted that, constraints 2, 10, 11 are taken from [22], constraints 3–9, 12–16 are based off of the work in [22], and constraints 17–40 are unique.

Constraint 2 enforces the restriction that there is only one MCH

$$\sum_{j=1}^{N} M_j = 1. \quad (2)$$

Constraint 3 enforces the restriction that the total number of CHs is $P - 1$. That is to say that if there are a total of P clusterheads, there will be 1 MCH and $P - 1$ regular clusterheads

$$\sum_{j=1}^{N} y_j = P - 1. \quad (3)$$

Constraint 4 is the upper limit on the total number of connections a node has. If a node is a regular node it can at most be connected to one other node (this node will be clusterhead as enforced by later constraints). If a node is a clusterhead, it will be connected at most to K other regular nodes (this enforces the restriction of maximum cluster size)

$$\sum_{i=1}^{N} x_{i,j} \leq 1 + (K-1)y_j \quad \forall j. \tag{4}$$

Constraint 5 is the lower limit on the total number of connections a node has. If a node is a regular node it must be connected to at least one other node (which will be a clusterhead as enforced by later constraints). If a node is a clusterhead it must support at least one node. If a node is a MCH it is not restricted to "1 connection to a regular node". Rather, it can have (and in this case it should have) no connections to regular nodes

$$\sum_{i=1}^{N} x_{i,j} \geq 1 - M_j \quad \forall j. \tag{5}$$

Constraint 6 is the upper limit on the maximum number of backbone connections. If a node is a clusterhead it cannot have more than 3 backbone connections. (One will be to a MCH for the star connection, and two will be to other regular clusterheads in order to establish the ring links). If a node is a MCH, it will be connected to all the regular clusterheads $(P-1)$

$$\sum_{\substack{j=1 \\ j \neq k}}^{N} z_{j,k} \leq (P-1)M_k + 3y_k \quad \forall k. \tag{6}$$

Constraint 7 is used to enforce the lower limit on the number of backbone connections. If a node is a regular clusterhead then it has to be connected to at least two other nodes, one other regular clusterhead and one MCH. If a node is a MCH, it has to be connected to all the regular clusterheads $(P-1)$

$$\sum_{\substack{j=1 \\ j \neq k}}^{N} z_{j,k} \geq (P-1)M_k + 2y_k \quad \forall k. \tag{7}$$

Constraint 8 is used to enforce the restriction that backbone connections are only between the MCH and regular clusterheads, or between regular clusterheads. The connections between regular nodes and clusterheads are not counted as backbone connections

$$z_{j,k} \leq \frac{M_j + y_j + M_k + y_k}{2} \qquad \forall j \forall k j \neq k. \tag{8}$$

Constraint 9 is used to enforce the restriction that if a node is selected to be a regular clusterhead, it cannot be the MCH and vice versa. The node can only be one of the two

$$(y_j + M_j) \leq 1 \qquad \forall j. \tag{9}$$

Constraint 10 is used to ensure that nodes are not connected to themselves and Constraint 11 is used to diagonalize the matrix x which represents the connections between regular nodes and regular clusterheads. That is to say that if clusterhead 1 is connected to node 2, it is the same as saying that node 2 is connected to clusterhead 1. Constraint 12 does the same for the z matrix which represents the interconnections between clusterheads

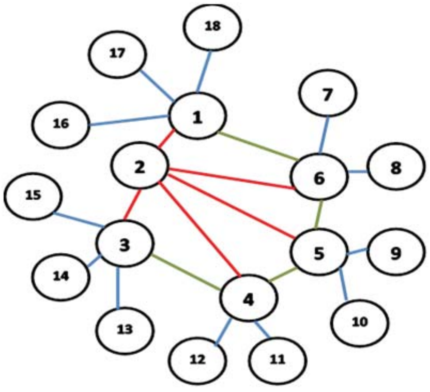


Fig. 3. Star-ring backbone.

$$\sum_{i=1}^{N} x_{ii} = 0 \tag{10}$$

$$x_{i,j} = x_{j,i} \quad \forall i \quad \forall j \tag{11}$$

$$z_{i,j} = z_{j,i} \quad \forall i \quad \forall j. \tag{12}$$

Constraint 13 restricts the total number of connections between regular nodes and clusterheads to the same number as the number of regular nodes; each regular node must be connected to at least one other clusterhead

$$\sum_{i=1}^{N} \sum_{j=i+1}^{N} x_{i,j} = (N-P). \tag{13}$$

Constraint 14 is used to restrict the total number of backbone connections to $2(P-1)-1$. 1 is deducted because the ring will be left "open" as described earlier

$$\sum_{i=1}^{N} \sum_{j=i+1}^{N} z_{i,j} = 2(P-1) - 1. \tag{14}$$

Constraint 15 is used to ensure that clusterheads do not connect to themselves

$$\sum_{i=1}^{N} z_{ii} = 0. \tag{15}$$

Constraint 16 is used to ensure that regular nodes are not connected to each other. When x (non-backbone) connections are made, at least one of the nodes must be a clusterhead

$$x_{j,k} \leq \frac{1 + y_j + y_k}{2} \quad \forall j \forall k j \neq k. \tag{16}$$

All topologies must have at least one MCH and two regular clusterheads. All regular clusterheads must support at least one regular node. MCHs do not connect to any regular nodes. The MCH is not counted as one of the regular clusterheads.

The goal is to have a "Star-Ring" Backbone as shown in Fig. 3, where there are fewer redundant links as compared to the EEC-FCB model [22] (example: unlike a mesh topology, node 1 is not connected to node 3, 4, or 5 directly). Additionally, a single point of failure is avoided unlike the EEC-CB model [22].

Instead of a mesh, in Fig. 3, there is a star connection between node 2, which is the MCH, and the other clusterheads. In addition the clusterheads are also interconnected through a ring.

B. Intra Cluster Communication Enhancement

Intra Cluster communication is introduced for two reasons. The first is that the primary responsibility of the clusterhead should be to route communication between clusters and not within a cluster. The goal is for the clusterhead to conserve as much energy as possible for the communication between clusters, allowing it to last longer in its role as a clusterhead. The second reason behind enhancing the intra cluster communication is that should a clusterhead fail, the nodes within a cluster will be able to communicate.

Equation 17 is the objective function to be minimized. The structure of the objective function is kept similar to the one used in the EEC-FCB and EEC-CB models in [22]. It is the objective function used in the proposed "Star-Ring" model with one additional term

$$Min: \left(\sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}x_{ij} + \sum_{j=1}^{N} b_j y_j + \sum_{j=1}^{N} b_j M_j + \sum_{j=1}^{N}\sum_{k=1}^{N} h_{jk}z_{jk} + \sum_{j=1}^{N}\sum_{k=1}^{N} A_{jk}\nu_{jk} \right). \quad (17)$$

The first term in the objective function represents the connections between nodes and clusterheads. The second term represents the selection of nodes to be clusterheads. The third term represents the cost of selection of the MCH. The fourth term is the cost of connections between clusterheads (backbone). The final term is the additional term added to incorporate the cost of connections between regular nodes within the same cluster, i.e. Intra-Cluster communication. The objective function aims to minimize the cost of sending/receiving data along these connections.

The proposed enhancement requires the introduction of the following new variables. Variable $\upsilon_{i,j}$ and f_{i,j,m^*} are two new variables used when enabling Intra-Cluster connections where

$$f_{i,j,m^*} = \begin{cases} 1, & \textit{if node } i \textit{ and } j \textit{ are connected} \\ & \textit{to the same clusterhead} \\ 0, & \textit{otherwise} \end{cases}$$

$$\upsilon_{i,j} = \begin{cases} 1, & \textit{if regular node } i \textit{ is connected} \\ & \textit{to regular note } j \\ 0, & \textit{otherwise.} \end{cases}$$

m^* is an index starting from 0, incremented when three conditions ($i \neq j$, $j \neq k$, $i \neq k$) are satisfied and used to indicate a possibility of 2 nodes being connected to the same clusterhead.

m^* is used to indicate the number of possibility, not the identity of nodes involved. There will always be $N - 2$ possibilities. For example, in a network of 7 nodes, when considering node i and node j, one must check if they are both connected to the same clusterhead which could be anyone of the 5 remaining nodes (should they be selected to be clusterheads). In addition to the previously described constraints (constraint 2 through to constraint16), constraints 18 through to constraint 23 are now also enforced to enable intracluster communication.

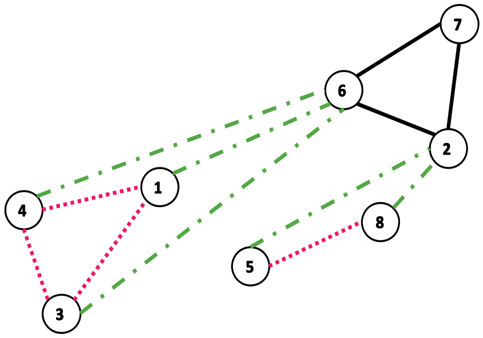


Fig. 4. Sample MANET topology with intracluster conections.

Constraints 18 and 19 are used to identify that node i is connected to node j if they are both connected to the same clusterhead. (N = total number of nodes)

$$(\nu_{i,j} - \sum_{k=0}^{N-3} f_{i,j,k}) \leq 0 \qquad \forall i \forall j i \neq j. \quad (18)$$

$$(N\nu_{i,j} - \sum_{k=0}^{N-3} f_{i,j,k}) \geq \qquad \forall i \forall j i \neq j. \quad (19)$$

Constraint 20 is used to enforce the restriction that a node cannot connect to itself through a hop

$$\sum_{i=1}^{N} \nu_{ii} = 0. \quad (20)$$

Constraint 21 is used to state that node i being connected to node j in the same cluster also implies that node j is connected to node i (Matrix is diagonal)

$$\nu_{i,j} + \nu_{j,i} = 0 \qquad \forall i \forall j. \quad (21)$$

Constraints 22 and 23 are used together to implement an "AND" logic. Node i and node j are connected through an Intra-Cluster connection if they are both connected to clusterhead k, satisfying the m^*th possible clusterhead connection

$$2f_{i,j,m*} - x_{i,k} - x_{j,k} \leq 0 \qquad \forall i \forall j \forall k i \neq j i \neq k j \neq k \quad (22)$$

$$x_{i,k} + x_{j,k} - f_{i,j,m*} \leq 1 \qquad \forall i \forall j \forall k i \neq j i \neq k j \neq k. \quad (23)$$

Fig. 4 shows the solution obtained by solving the formulation with Intra-Cluster communication enabled for the given nodes. As indicated, node 8 and node 5 are connected to clusterhead 2 but are also interconnected. Similarly node 1, 4 and 3 are connected to clusterhead 6 but are also interconnected.

C. Multihop Connections Enhancement

Multihop connections are introduced into the formulation to allow longer, more expensive links to be replaced by shorter less expensive links. Rather than connect directly to a clusterhead which is further away, it is preferable to make a lower cost connection to a clusterhead through another regular node. However, the intermediate regular node will now, in a sense, act like a second tier clusterhead as it will route the communication of the regular node through it to the clusterhead. The cost of this routing must be taken into account. The following objective function is used to incorporate the cost of multihop connections to the proposed Star-Ring base model

$$Min: \left(\sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij} x_{ij} + \sum_{j=1}^{N} a_j y_j + \sum_{j=1}^{N} a_j M_j \right. \\ \left. + \sum_{j=1}^{N} \sum_{k=1}^{N} h_{jk} z_{jk} + \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} B_{jk} b_{ijk} \right). \tag{24}$$

B_{jk} represents the cost of connecting node j and node k. This cost is similar to the cost in the original objective function in the proposed Base Model and in the Energy Efficient Clustering–Fully Connected Backbone (EEC-FCB) and Energy Efficient Clustering-Connected Backbone (EEC-CB) models presented by the authors in [22]. The cost of connecting node j and node k is approximated by *Bj, k* as it is proportional to the distance between the hopping node and the intermediate node which is used to hop to the clusterhead. This is shown in Equation 25

$$B_{j,k} \propto d_{j,k}^{n}. \tag{25}$$

The value of n depends on several factors including primarily the degree to which Multihop connections should be encouraged over direct connections. However, the value of n is not proportional to the square of the distance as with the regular node-clusterhead connections ($n = 2$), and it is not proportional to the cube of the distance as with the clusterhead-clusterhead connections ($n = 3$). Rather, it is somewhere in between. This is because the intermediate node (node j) will be responsible for not only sending its own information to the clusterhead but also routing the information from the hopping node (node k) to the clusterhead. This value can be adjusted or tuned through simulation based on how preferred Multihop connections are over direct connections. Based on our initial tests and simulations, we found that $n = 2.5$ provided solutions where multihop connections were not overly penalized. Larger values of n tended to over-penalize multihop connections and instead prefer to have nodes connect directly to clusterheads, while lower values of n resulted in nodes hopping over adjacent nodes even when the clusterhead was in range.

The following variables are introduced to formulate the multi-hop connection constraints

$$b_{i,j,k} = \begin{cases} 1, & \textit{if node } i \textit{ is connected to clusterhead} \\ & \quad k \textit{ through node } j \\ 0, & \textit{otherwise} \end{cases}$$

$$q_{i,j} = \begin{cases} 1, & \textit{if regular node } i \textit{ is connected} \\ & \quad \textit{to regular node } j \\ 0, & \textit{otherwise.} \end{cases}$$

Variable $b_{i,j,k}$ and $q_{i,j}$ are two new variables used when enabling multihop. These variables are required because the cost of the "hop" connection will be different from regular connections represented by variable $x_{i,j}$. It is also important to remember that certain restrictions must be kept in place, for example, regular nodes can only hop using the regular nodes to connect to clusterheads, since it is illogical if they hop using one clusterhead to connect to another clusterhead. The constraints required to enable multihop connections are as follows.

In addition to constraints 2–4, 7–12 and 14–16, described previously, constraints 26–38 are now added to enable multihop communication. Constraint 26 is used instead of Constraint 6. The maximum connections node "i" can have is K which occurs when node "i" is a clusterhead. In Multihop, the maximum cluster size must also include nodes that are connected to the clusterhead through hops

$$\sum_{\substack{j=1 \\ j \neq k \\ j \neq i}}^{N} \sum_{\substack{k=1 \\ k \neq i}}^{N} (b_{i,j,k} + x_{i,j}) \leq K \quad \forall i. \tag{26}$$

Constraint 27 is used instead of Constraint 5. The minimum number of connections that node "i" should have is 1 if it is a regular node and 0 if it is the MCH

$$\sum_{i=1}^{N} (x_{i,j} + q_{i,j}) \geq 1 - M_j \quad \forall j. \tag{27}$$

Constraint 28 is used instead of Constraint 13 and ensures that the total number of non-backbone connections is equal to $N - P$. This includes both hop based and direct connections

$$\sum_{i=1}^{N} \sum_{j=i+1}^{N} (x_{i,j} + q_{i,j}) = (N - P). \tag{28}$$

Constraint 29 is used to ensure that only those nodes that are connected to the clusterhead ($x_{ik} = 1$) can be used as hopping nodes

$$(t_{i,j,m*} - x_{i,k}) \leq 0 \qquad \forall i \forall j \forall k i \neq j i \neq k j \neq k. \tag{29}$$

m^* is an index starting from 0, incremented when three conditions are satisfied ($i \neq j$, $j \neq k$, $i \neq k$) and used to indicate a potential hop path. m^* is used to indicate the number of potential hope path, not the identity of the possible hop path which would be $t_{i,j,k}$. The former is used because the emphasis is on whether or not a "hop" path was taken and to simplify the coding of the model.

Constraints 30 and 31 are used to ensure that it is not possible to hop off of a clusterhead. That if y_j is 1 or y_i is 1 then all potential hops through y_j and y_i are deemed not possible because y_j or y_i is a clusterhead

$$(t_{i,j,m*} + y_j) \leq 1 \qquad \forall i \forall j i \neq j. \tag{30}$$

$$(t_{i,j,m*} + y_i) \leq 1 \qquad \forall i \forall j i \neq j. \tag{31}$$

Constraint 32 is added to ensure that only a direct connection to the clusterhead or a hop connection to a clusterhead exists from a particular node. The node cannot be connected to the clusterhead both directly and by hopping through another node

$$(x_{i,j} + q_{i,j}) \leq 1 \quad \forall i \forall j. \tag{32}$$

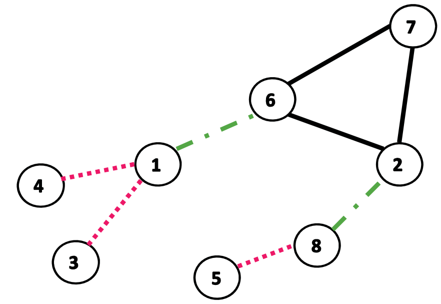


Fig. 5. Sample MANET topology with multihop connections.

Constraints 33 and 34 are used to identify that node i is connected to node j if it has hopped taking one of the potential hop paths. (N = total number of nodes)

$$\left(q_{i,j} - \sum_{k=0}^{N-3} t_{i,j,k}\right) \le 0 \qquad \forall i \forall j i \neq j \le 0. \tag{33}$$

$$\left(Nq_{i,j} - \sum_{k=0}^{N-3} t_{i,j,k}\right) \ge 0 \qquad \forall i \forall j i \neq j. \tag{34}$$

Constraint 35 is used to enforce the restriction that a node cannot connect to itself through a hop

$$\sum_{i=1}^{N} q_{ii} = 0. \tag{35}$$

Constraint 36 is used to state that node i cannot hop to j if j has hopped to i. (q connection matrix is not diagonal). Saying that node i has hopped to clusterhead j is not the same as saying the node j has hopped to clusterhead i. Which node is the clusterhead matters unlike with the x connection where just the presence of the connection matters

$$q_{i,j} + q_{j,i} \le 1 \quad \forall i \forall j. \tag{36}$$

Constraints 37 and 38 are used together to implement an "AND" logic. Node k can hop using node j to clusterhead i, if i is a clusterhead and j is connected to i and connecting k to j is possible

$$2b_{i,j,k} - x_{i,j} - q_{j,k} \le 0 \qquad \forall i \forall j \forall k i \neq j i \neq k j \neq k. \tag{37}$$

$$x_{i,j} + q_{j,k} - b_{i,j,k} \le 1 \qquad \forall i \forall j \forall k i \neq j i \neq k j \neq k. \tag{38}$$

Fig. 5 shows the solution obtained by solving the formulation Multihop links enabled for the given set of nodes. As shown, node 5 connects to clusterhead 2 through node 8. Similarly, nodes 3 and 4 connect to clusterhead 6 through node 1.

D. Coverage Enhancement

The proposed Base Model can be extended to take into account the coverage radius of the nodes in the network, and ensure that connections are established only between nodes that are within each other's coverage radius. Similar to the manner in which distances between nodes are used to determine the cost of the connections, they can also be compared to the coverage radius of each node and used to obtain a matrix of nodes to which each node can connect to and to which it can't.

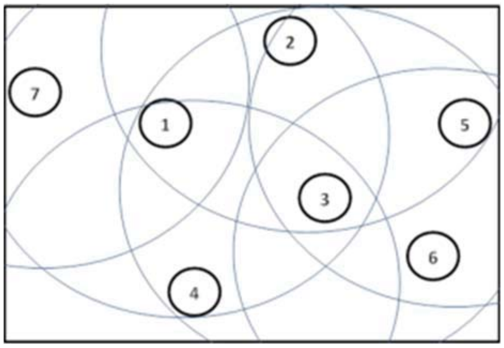


Fig. 6. Sample MANET topology.

This is done by making use of the matrix d, which is used to keep track of the distance between the nodes. Two new matrices are introduced; matrix nc and matrix cv. The variable $cv_{i,j}$ is the binary value which represents whether or not nodes j and i are in each other's coverage radius. This is determined by subtracting the distance between nodes from each node's coverage radius. If both results are positive then they can communicate, otherwise not. The actual establishment of the connection will depend on the cost (which is proportional to the distance)

$$cv_{i,j} = \begin{cases} 1, & \textit{if node i and node j can connect} \\ 0, & \textit{otherwise.} \end{cases}$$

These values can then be used to enforce the possibility of connectivity between nodes using constraints 18 and 19. It is important to keep in mind, that it is being assumed that external localization techniques such as GPS are being used.

In order to enforce coverage restrictions in any of the above models (Star-Ring Model, Star-Ring Model with Intra Cluster Communication, Star-Ring Model with Multihop Communications), Constraints 39 and 40 are now added to ensure that two nodes may only be connected if they lie within each other's coverage radius.

Constraints 39 and 40 ensure that two nodes may only be connected if they lie within each other's coverage radius

$$x_{i,j} \le cv_{i,j} \qquad \forall i \forall j. \tag{39}$$

$$z_{i,j} \le cv_{i,j} \qquad \forall i \forall j. \tag{40}$$

E. Illustrative Example

The example topology shown in Fig. 6 will be used to illustrate the difference when the Star-Ring model is used without coverage and when the coverage constraints are included.

In Fig. 7 (without coverage), node 5 and node 1 were connected and node 3 and node 7 were connected, even though they were outside of each other's coverage radius.

As shown in Fig. 8, connectionsare not made when nodes are not in each other's coverage radius.

V. TOOL DESIGN AND DEVELOPMENT

In order to effectively analyze the solutions generated by the formulation, both with and without enhancement, an intuitive

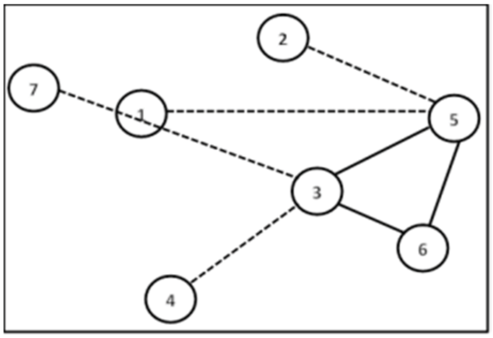


Fig. 7. Topology generated by the ILP formulation without coverage constraints.

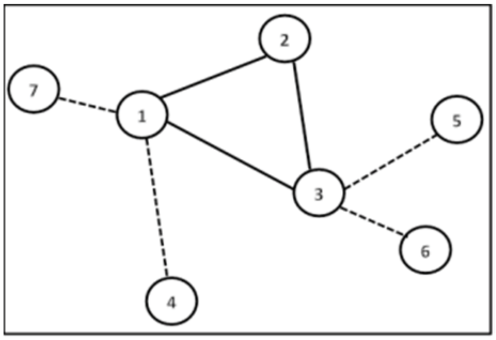


Fig. 8. Topology generated by the ILP formulation with coverage constraints.

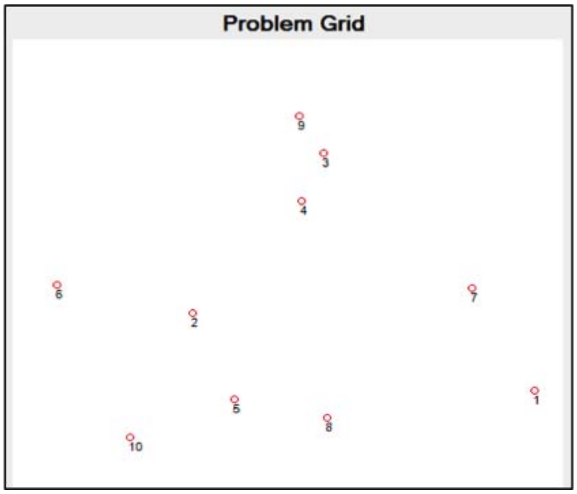
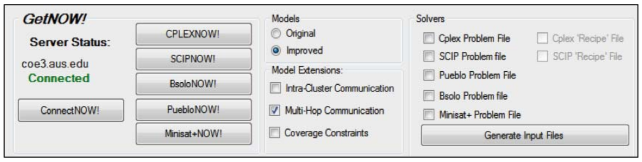


Fig. 9. Tool working area.

tool was designed which allowed creation of custom network topologies. The goal is to allow simplified interaction with a selected set of the state-of-the-art solvers; commercial generic-based ILP solver CPLEX [27], non-commercial generic-based ILP solver SCIP [28], 0–1 SAT-based ILP solvers BSOLO [29], Pueblo [30] and Minisat+ [31].

Fig. 9 is a screenshot of the working area in the tool. The "Problem Grid" is where custom network topologies are created through placing of nodes by clicking on the grid. The desired number of clusterheads and maximum cluster size are entered (the coverage radius of the nodes is also entered if the coverage enhancement will be used).



Server Status:
coe3.aus.edu
Connected

Models
○ Original
● Improved

Model Extensions:
☐ Intra-Cluster Communication
☑ Multi-Hop Communication
☐ Coverage Constraints

Solvers
☐ Cplex Problem File
☐ SCIP Problem file
☐ Pueblo Problem File
☐ Bsolo Problem file
☐ Minisat+ Problem File
☐ Cplex 'Recipe' File
☐ SCIP 'Recipe' File

Fig. 10. Connection status and topology customization panel.

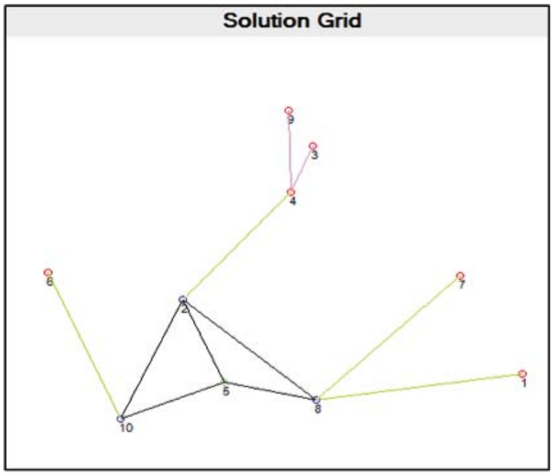


Fig. 11. Solution grid.

Once the nodes have been placed and the desired parameters entered, there are two alternatives to generate the corresponding ILP formulation and have it solved by the desired solver. The first method is the "manual" method. When using this method, the first step is selecting the model to use (Example: original FCB model in [22], proposed SR formulation, desired enhancements if any), and then selecting solver(s) for which the ILP formulation should be generated. This is useful for solving batches of files, which can be manually sent to the solvers, solved, and the solutions imported. The second method is the "automatic" method or the "GetNow" method, where the tool will establish an SSH and SCP connection to the server on which the solvers are installed, through the use of the TamirSSH library [32]. The model and enhancements need to be selected, and the tool will generate the files, transfer them to the server, execute the desired solver, and retrieve the generated solution. This is useful for testing one topology at a time. Shown in Fig. 10, is the panel in the tool which displays the connection status to the server with the solvers, the model selection options, and the right side options, to generate files for manual use.

The generated solution whether retrieved manually or automatically can then be viewed in the "Solution Grid," side-by-side with the original topology, as shown in Fig. 11.

In the bottom right hand corner of Fig. 11, are the options to customize the view of the generated solution, allowing the user to see only the clusterhead backbone connections, individual node connections, and only connections within a cluster, etc. The solution shown in Fig. 11 is the result obtained by solving the corresponding ILP formulation of the topology in Fig. 9, for the proposed SR model with the Multihop Connection enhancement enabled.

The tool also provides a comprehensive set of other features such as the ability to generate large scale network topologies with randomized node placements if desired, and to generate the ILP formulations for all of them for a selected model, selected set of enhancements and for a selected set of solvers. In addition it also allows the user to add comments and take print outs of topologies and the corresponding solutions along with the user's comments. Credentials to log in to the server on which the solvers are located are also stored in a database in the tool, and can be edited, thereby removing the need for the user to manually log in to the server.

VI. TESTING AND RESULTS

Testing was carried out using the following solvers: commercial generic-based ILP solver CPLEX [27], non-commercial generic-based ILP solver SCIP [28], 0–1 SAT-based ILP solvers BSOLO [29], Pueblo [30] and Minisat+ [31]. All experiments were conducted on an Intel Xeon 3.2 Ghz workstation running Linux with 4 GB of RAM. Three sets of tests were conducted; the first set with the proposed Star-Ring model with no enhancements enabled, the second set with the proposed Star-Ring model and the Intra-cluster communication enhancement enabled, and the third set with the proposed Star-Ring model and the Multihop Connection enhancement enabled. All the network topologies were automatically and randomly generated through the developed tool. The network configurations consider three key parameters: #N: the number of nodes in the topology, #CH: The desired number of clusterheads in the solution, #MCS: The maximum clustersize which is the maximum number of nodes a clusterhead can support. The minimum MCS value is obtained for a given number of nodes and clusterheads as follows. If there is a network of 40 nodes, and the desired number of clusterheads is 3, then the total number of regular nodes is 37. The 3 clusterheads must support the 37 regular nodes, however, one of the clusterheads is the MCH which does not connect to any regular nodes, therefore the 2 regular clusterheads must support the 37 regular nodes. This implies that each clusterhead must support 19 nodes or more. In this way the minimum MCS value is determined and is used in our test cases.

A. *Testing Solver Performance with the Proposed Star-Ring Model*

Testing was carried out for various network configurations. For a fixed number of nodes, the number of clusterheads, and the maximum supported clustersize were varied to analyze their effect on the solvers' performance in solving the ILP formulation. The performance of the set of solvers was assessed in solving the proposed ILP Star-Ring model (without any enhancements enabled). The results are shown in Table I.

For each network configuration shown in Table I, 100 tests were generated and solved by the different solvers. For each network configuration in Table I, the average time to solve the corresponding 100 instances, is shown. A timeout of 15 min (900 s) was set for all solvers. As can be seen from Table I, the SAT solvers such as BSOLO and Pueblo performed well for the smaller scale networks.

TABLE I

SOLVER PERFORMANCE IN SOLVING SR ILP FORMULATION OF THE CLUSTERING PROBLEMS

Network Configurations			**Solver Times (s)**				
			Proposed SR Model				
#N	***#CH***	***#MCS***	***CPLEX***	***SCIP***	***BSOLO***	***Pueblo***	***Minisat+***
5	3	1	0.257	0.014	0.002	0.001	0.038
7	3	2	0.285	0.023	0.007	0.008	0.180
9	3	3	0.376	0.060	0.026	0.040	1.180
11	3	4	0.468	0.148	0.063	0.349	5.79
13	3	5	0.637	0.428	0.281	-	31.85
15	3	6	0.725	1.017	0.950	-	242.52
40	3	19	**5.73**	121.18	!	!	!
45	3	21	**8.27**	162.58	!	!	!
50	3	24	**16.29**	303.25	!	!	!
7	4	1	0.356	0.051	0.013	0.011	0.37
9	4	2	0.546	0.150	0.055	0.072	6.73
11	4	3	0.571	0.297	0.152	0.531	76.19
13	4	3	0.795	0.967	1.030	-	349.54
15	4	4	0.903	1.709	4.753	-	!
40	4	12	**45.33**	304.85	!	!	!
45	4	14	**94.39**	385.25	!	!	!
50	4	16	**271.92**	665.64	!	!	!
9	5	1	0.532	0.353	0.098	0.107	8.59
11	5	2	0.745	0.9	0.366	1.833	200.95
13	5	2	0.834	2.058	1.633	-	!
15	5	3	1.035	3.212	7.433	-	!
40	5	9	**112.92**	489.4	!	!	!
45	5	10	**195.17**	!	!	!	!
50	5	12	**427.51**	!	!	!	!

Note: "-" and "!" represents "CANNOT SOLVE" and "TIMEOUT," respectively.

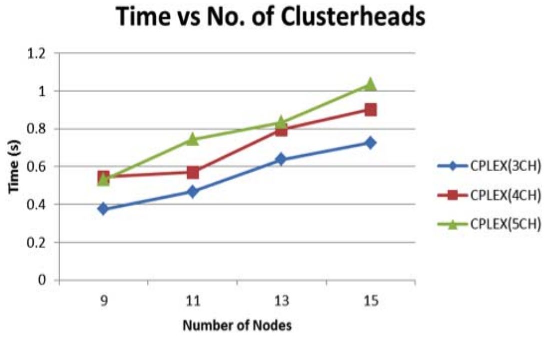


Fig. 12. Dependency of CPLEX solver runtimes on the number of desired clusterheads in the solution.

However, CPLEX and SCIP proved to be faster as the size of the network increased. The comparison of times taken by CPLEX to solve a topology with a fixed number of nodes but with different specified clusterheads is shown in Fig. 11.

From Fig. 12, it is observed that for a given number of nodes, CPLEX take a longer time to solve topologies which have a larger number of clusterheads. The other solvers behave similarly.

It can be clearly seen, that for a fixed number of nodes, if topologies with a larger number of clusterheads is to be generated, the solvers will take more time to generate the

TABLE II
SOLVER PERFORMANCE IN SOLVING SR WITH INTRACLUSTER COMMUNICATION ILP FORMULATION

Network Configurations			**Solver Times (s)** *Proposed SR Model*				
#N	***#CH***	***#MCS***	***CPLEX***	***SCIP***	***BSOLO***	***Pueblo***	***Minisat+***
5	3	1	0.459	0.019	0.004	**0.002**	0.061
7	3	2	1.810	1.657	0.055	**0.022**	0.366
9	3	3	1.810	10.168	0.172	**0.106**	4.244
11	3	4	13.701	44.178	0.500	**0.566**	48.341
13	3	5	58.213	167.48	**3.642**	-	453.63
15	3	6	310.03	!	**30.127**	-	!
7	4	1	0.354	0.070	**0.022**	0.025	0.764
9	4	2	1.463	5.791	0.207	**0.139**	25.139
11	4	3	5.543	28.558	**0.645**	1.066	367.87
13	4	3	31.538	116.23	**9.057**	-	!
15	4	4	119.61	589.97	**74.844**	-	!
9	5	1	0.898	0.387	**0.132**	0.174	22.154
11	5	2	4.197	16.867	**1.047**	2.462	456.81
13	5	2	28.365	71.119	**7.314**	-	!
15	5	3	75.028	204.69	**74.363**	-	!

Note: "-" and "!" represents "CANNOT SOLVE" and "TIMEOUT," respectively.

solution. Similar, observations can be made for the other solvers used. Additionally, it is observed that Pueblo [30] is unable to handle certain instances and ends up in the "Cannot Solve" state shown by a "-" in Table I. This is due to Pueblo's inability to handle problems with large coefficients. The large coefficients present in the ILP formulations are the costs associated with interconnecting nodes. (The cost of the link connecting a regular node to a clusterhead is proportional to the square of the distance between the nodes, and the cost of interconnecting clusterheads is proportional to the cube of the distance between the clusterheads [22].)

Overall, it is observed that CPLEX and SCIP perform well, with MINISAT+ being the slowest solver for the presented benchmarks. Among the set of selected solvers, CPLEX and SCIP handle the larger networks well as they almost never timeout. In the case of the Star-Ring (SR) model, the SAT solvers BSOLO and Pueblo are very fast for the smaller networks, however as the size of the network increases, their time-to-solve increases faster than the generic ILP solvers, i.e. CPLEX and SCIP. MINISAT+ times out and CPLEX and SCIP are the fastest solvers for the SR models. The solver used to solve the ILP formulation presented in [22], timed out when solving for more than 9 node topologies. In our tests, conducted with the proposed ILP formulations and enhancements, solvers such as CPLEX and SCIP were able to handle ILP formulations of networks up to 50 nodes.

CPLEX in particular is far from timing out even at 50 node topologies. It is important to note that the timeout used in testing was 15 min (900 s).

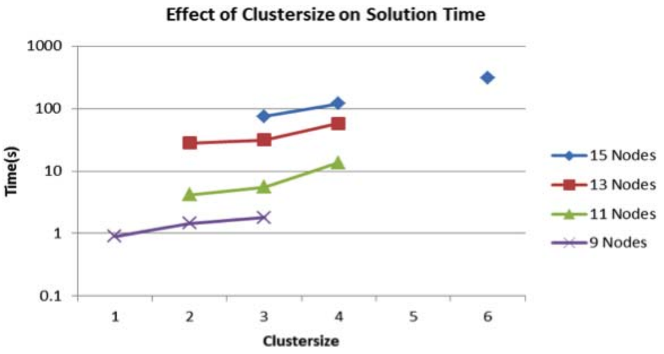


Fig. 13. Dependency of CPLEX solver runtimes on maximum clustersize.

B. Testing Solver Performance With the Intra Cluster Communication Enhancement

The second set of tests was identical to the first in every respect, except that in this case the Intra-Cluster Communication enhancement was enabled.

For each network configuration shown in Table II, 100 tests were generated and solved by the different solvers. The average time to solve the corresponding 100 instances, is shown. A timeout of 15 min (900 s) was set for all solvers. As can be seen from Table II, the SAT solver BSOLO is the fastest among the selected solvers while Pueblo performed well for the smaller scale networks.

As was the case with the Star-Ring model, it is observed that Pueblo [30] is unable to handle certain instances and ends up in the "Cannot Solve" state shown by a "-." This is due to Pueblo's inability to handle problems with large coefficients. The large coefficients present in the ILP formulations are the costs associated with interconnecting nodes.

Additionally, from Fig. 13, it is observed that for a given number of nodes, CPLEX takes a longer time to solve topologies which have greater specified maximum clustersize. Understandably, this is due to the increased number of intracluster links which need to be generated in larger clusters. The other solvers behave similarly.

C. Testing Solver Performance With the Multihop Connection Enhancement Enabled

The third set of tests was identical to the first in every respect, except that in this case the Multihop Connection enhancement was enabled. For each network configuration shown in Table III, 100 tests were generated and solved by the different solvers. The average time to solve the corresponding 100 instances, is shown. A timeout of 15 min (900 s) was set for all solvers. It is observed that BSOLO performs well for small scalenetworks while CPLEX is clearly the faster solver for thelarger scale networks and is significantly ahead of SCIP. Pueblo is unable to handle any of the formulations, while MINISAT+ times out in most of the cases. CPLEX and SCIP perform well for the larger scale networks without timing out in any case. Pueblo is unable to handle any instance with large coefficients and MINISAT+ is the slowest solver for both proposed enhancements.

TABLE III
SOLVER PERFORMANCE IN SOLVING THE SR WITH MULTIHOP ILP FORMULATION

Network Configurations			**Solver Times (s)**				
			Proposed SR Model				
#N	***#CH***	***#MCS***	***CPLEX***	***SCIP***	***BSOLO***	***Pueblo***	***Minisat+***
5	3	1	0.277	0.038	**0.007**	-	0.372
7	3	2	0.600	2.005	**0.187**	-	237.85
9	3	3	**1.621**	9.485	2.451	-	!
11	3	4	**11.116**	28.294	75.725	-	!
13	3	5	**50.044**	125.69	765.24	-	!
15	3	6	**168.97**	329.92	!	-	!
7	4	1	**50.225**	150.82	688.56	-	!
9	4	2	**190.57**	401.77	!	-	!
11	4	3	0.705	0.599	**0.231**	-	135.94
13	4	3	12.102	68.562	**8.449**	-	!
15	4	4	**69.610**	255.45	310.90	-	!
9	5	1	**1.621**	9.485	2.451	-	!
11	5	2	**11.116**	28.294	75.725	-	!
13	5	2	**50.044**	125.69	765.24	-	!
15	5	3	**168.97**	329.92	!	-	!

Note: "-" and "!" represents "CANNOT SOLVE" and "TIMEOUT," respectively.

The Intra-Cluster communication and Multihop enhancements increase the complexity of the ILP formulation, requiring solvers to take more time to generate the enhanced topologies as compared to standalone Star-Ring topologies.

D. Discussion on Scalability

As mentioned earlier, a major benefit of clustering the network is achieving salability. The proposed ILP formulation organizes the network in a hierarchical fashion to ensure scalability and energy efficiency. Given that the suggested ILP solution to the presented problem gives the optimal layout of the underlying network, it is certain that the best available scalability is achieved. The ILP, however, takes more time to find the optimal solution as the network gets bigger. Hence, the size of the network has no effect on the results of the ILP solution, but has an effect on the time it takes to solve the problem. As a result, the presented benchmarks come in very handy for other researchers who can suggest smart heuristics or meta-heuristics to the clustering problem and compare their results with the results of our optimal ILP solution. This provides means for researchers to evaluate the efficiency of their approaches.

It is also important to note, that the proposed formulations assume that the desired number of clusterheads and the maximum clustersize are known. Non-ILP based approaches and adaptations such as constant density clustering [33] which aim to improve scalability and ensure stable density of clusterheads per unit area, can assist in determining the optimal number of clusterheads from a geospatial perspective. However, factors such as residual energy and resource characteristics of a node are not considered in [33] leading to a non-optimal network lifetime. By adapting the approach proposed in [33] to be included as part of the clusterhead selection cost matrix in the proposed formulation, it may be possible to obtain the optimal number of clusterheads as well.

VII. CONCLUSION

This paper puts forward an improved ILP formulation to solve the clustering problem in MANETs. The proposed model presented the use of a Star-Ring backbone. Additionally, the proposed formulation included the ability to enforce coverage constraints to ensure that only connections that are within the physical limitations of the node are established. The enhancements include the ability for nodes within the same cluster to communicate without going through the designated clusterhead, and the ability to establish multihop links. Using the proposed ILP formulations and enhancements together with a custom designed tool, it was possible to test the performance and analyse the feasibility of Generic ILP and SAT solvers (CPLEX [27], SCIP [28], BSOLO [29], Pueblo [30] and Minisat+ [31]) in solving the clustering problem for MANETs. SAT solvers, BSOLO and Pueblo performed well for small scale networks while CPLEX and SCIP were able to handle the larger scale topologies. In most cases, CPLEX was the fastest solver from the selected set of solvers. It is observed that while these enhanced formulations enable the generation of complex network solutions, and are suitable for small scale networks, the time taken to generate the corresponding solution does not meet the strict requirements of a practical environment. However, as mentioned earlier, the formulations provide a strong mathematical foundation to the clustering problem by presenting an optimal framework that allows other researchers to suggest smart heuristics and meta-heuristics and compare their results with the optimal ones provided in this paper. In addition, the comparison and comprehensive performance evaluation of the state-of-the-art ILP (Generic-based and 0–1 SAT-based) solvers in handling the proposed ILP formulation and enhancements has illustrated which solver works best under different circumstances.

REFERENCES

[1] S. Chinara and S. Rath, "Energy efficient mobility adaptive distributed clustering algorithm for mobile ad hoc network," in *Proc. Int. Conf. Adv. Comput. Commun.*, 2008, pp. 265–272.

[2] A. Schrijver, *Theory of Linear and Integer Programming*.New York, NY, USA: Wiley, 1999.

[3] F. Aloul, A. Ramani, I. Markov, and K. Sakallah, "Generic ILP versus specialized 0-1 ILP: An update," in *Proc. Int. Conf. Comput.-Aided Design*, 2002, pp. 450–457.

[4] D. Chai and A. Kuehlmann, "A fast pseudo-Boolean constraint solver," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 3, pp. 305–317, Mar. 2005.

[5] A. Sagahyroon, F. Aloul, and A. Sudnitson, "Using SAT-based techniques in low power state assignment," *J. Circuits, Syst., Comput.*, vol. 20, no. 8, pp. 1605–1618, 2011.

[6] A. Ling, D. Singh, and S. Brown, "FPGA programmable logic block evaluation using quantified Boolean satisfiability," *IEEE Comput. Digital Tech.*, vol. 153, no. 3, pp. 165–172, May 2006.

[7] S. Kim and J. Lee, "A system architecture for high-speed deep packet inspection in signature-based network intrusion prevention," *J. Syst. Archit., EUROMICRO J.*, vol. 53, nos. 5–6, pp. 310–320, May 2007.

[8] G. Hughes and T. Bultan, "Automated verification of access control policies using a SAT solver," *Int. J. Softw. Tools Technol. Transf.*, vol. 10, no. 6, pp. 503–520, Oct. 2008.

[9] I. Mironov and L. Zhang, "Applications of SAT solver to cryptanalysis of hash functions," in *Proc. Int. Conf. Theory Appl. Satisfiab. Test.*, 2006, pp. 102–115.

[10] S. Tosun, O. Ozturk, and M. Ozen, “An ILP formulation for application mapping onto Network-on-Chips,” in *Proc. Appl. Inf. Commun. Technol.*, 2009, pp. 1–5.

[11] A. Graca, I. Lynce, J. Marques-Silva, and A. L. Oliveira, “Haplotype inference with pseudo-Boolean optimization,” *Ann. Oper. Res.*, vol. 184, no. 1, pp. 137–162, 2011.

[12] A. Wasfy and F. Aloul, “Solving the University class scheduling problem using advanced ILP techniques,” in *Proc. IEEE GCC Conf.*, Nov. 2007, pp. 1–5.

[13] J. Garcia-Macias and J. Gomez, “MANET versus WSN,” in *Sensor Networks and Configuration: Fundamentals, Standards, Platforms, and Applications*. New York, NY, USA: Springer-Verlag, 2006, ch. 17, pp. 369–388.

[14] K. Xu, X. Hong, and M. Gerla, “An ad hoc network with mobile backbones,” in *Proc. IEEE Int. Conf. Commun.*, vol. 5. 2002, pp. 3138–3143.

[15] J. N. Al-Karaki, A. E. Kamal, and R. Ul-Mustafa, “On the optimal clustering in mobile ad hoc networks,” in *Proc. Consum. Commun. Netw. Conf.*, 2004, pp. 71–76.

[16] H. Safa, O. Mirza, and H. Artail, “A dynamic energy efficient clustering algorithm for MANETs,” in *Proc. IEEE Int. Conf. Wireless Mobile Comput., Netw. Commun.*, Oct. 2008, pp. 51–56.

[17] J. G. Olascuaga-Cabrera, E. Lopez-Mellado, A. Mendez-Vazquez, and F. F. Ramos-Corchado, “A self-organization algorithm for robust networking of wireless devices,” *IEEE Sensors J.*, vol. 11, no. 3, pp. 771–780, Mar. 2011.

[18] J. Yu and P. Chong, “A survey of clustering schemes for mobile ad hoc networks,” *IEEE Commun. Surv. Tuts.*, vol. 7, no. 1, pp. 32–48, Jan. 2005.

[19] A. Abbasi and M. Younis, “A survey on clustering algorithms for wireless sensor networks,” *Comput. Commun.*, vol. 30, nos. 14–15, pp. 2826–2841, Oct. 2007.

[20] S. Guo and O. Yang, “Minimum-energy multicast routing in static wireless ad hoc networks,” in *Proc. Veh. Technol. Conf.*, vol. 6. Sep. 2004, pp. 3989–3993.

[21] F.-W. Chen and J.-C. Kao, “Game-based broadcast over reliable and unreliable wireless links in wireless multihop networks,” in *Proc. IEEE Int. Conf. Commun.*, vol. 5. Apr. 2002, pp. 3138–3143.

[22] W. El-Hajj, D. Kountanis, A. Al-Fuqaha, and H. Harbi, “Optimal hierarchical energy efficient design for MANETs,” in *Proc. Int. Conf. Commun. Mobile Comput.*, 2006, pp. 287–292.

[23] A. Bari, A. Jaekel, and S. Bandyopadhyay, “Clustering strategies for improving the lifetime of two-tiered sensor networks,” *Comput. Commun.*, vol. 31, no. 14, pp. 3451–3459, Sep. 2008.

[24] J. Yick, A. Bharathidasan, G. Pasternack, B. Mukherjee, and D. Ghosal, “Optimizing placement of beacons and data loggers in a sensor network—A case study,” in *Proc. Wireless Commun. Netw. Conf.*, vol. 4. 2004, pp. 2486–2491.

[25] W. El-Hajj, Z. Trabelsi, and D. Kountanis, “Fast distributed dominating set based routing in large scale MANETs,” *Comput. Commun.*, vol. 30, nos. 14–15, pp. 2880–2891, Oct. 2007.

[26] W. El-Hajj, A. Al-Fuqaha, M. Guizani, and H. Chen, “On efficient network planning and routing in large scale manets,” *IEEE Trans. Veh. Technol.*, vol. 58, no. 7, pp. 3796–3801, Sep. 2009.

[27] *IBM ILOG CPLEX Optimizer*. (2011) [Online]. Available: http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/

[28] *SCIP: An Open Source MIP Solver and Constraint Integer Programming Framework*. (2011) [Online]. Available: http://scip.zib.de/

[29] V. Manquinho and J. Marques-Silva, “On using cutting planes in pseudo Boolean optimization,” *J. Satisfiab., Boolean Model. Comput.*, vol. 2, pp. 209–219, Mar. 2006.

[30] H. Sheini and K. Sakallah, “Pueblo: A hybrid pseudo-Boolean SAT solver,” *J. Satisfiab., Boolean Model. Comput.*, vol. 2, nos. 1–4, pp. 165–189, 2006.

[31] N. Een and N. Sorensson, “An extensible SAT-solver,” in *Proc. Int. Conf. Theory Appl. Satisfiab. Test.*, 2003, pp. 502–508.

[32] T. Gal. (2008). *SharpSSH—A Secure Shell (SSH) Library for .NET* [Online]. Available: http://www.tamirgal.com/blog/page/SharpSSH.aspx

[33] H. Yang, L. Ritchie, A. Richa, and M. Reisslein, “MANET routing with provably low complexity through constant density clustering and route request broadcast,” *Wireless Personal Commun.*, vol. 43, no. 2, pp. 605–621, Oct. 2007.

Syed Zohaib Hussain Zahidi received the B.Sc. degree in computer engineering degree and the M.Sc. degree in computer engineering degree from the American University of Sharjah, Sharjah, UAE, in 2008 and 2012, respectively.

His current research interests include applied integer linear programming, mobile ad-hoc networks, wireless sensor networks, mobile application development and the field of social business and social collaboration.

Mr. Zahidi is currently the Web and Social Business Manager for IBM, Middle East and Africa.

Fadi Aloul (M’03–SM’08) received the B.S. degree in electrical engineering (*summa cum laude*) from Lawrence Technological University, Southfield, MI, USA, and the M.S. and Ph.D. degrees in computer science and engineering from the University of Michigan, Ann Arbor, MI, USA.

He is currently an Associate Professor of computer science and engineering with the American University of Sharjah, Sharjah, UAE. He was a Post-Doctoral Research Fellow with the University of Michigan in 2003 and a Visiting Researcher with the Advanced Technology Group, Synopsys, in 2005. He has more than 90 publications in international journals and conferences and a U.S. patent. His current research interests include design automation, optimization, and computer security.

Dr. Aloul has received a number of awards, including the prestigious Sheikh Khalifa, UAE’s President, Award for Higher Education, AUS Excellence in Teaching Award, and the Semiconductor Research Corporation Research Fellowship. He is a Senior Member of ACM.

Assim Sagahyroon received the M.Sc. degree from Northwestern University, Evanston, IL, USA, and the Ph.D. degree from the University of Arizona, Tuscon, AZ, USA.

He was a Tenured Faculty Member with the Department of Computer Science and Engineering, Northern Arizona University, and then joined the Department of Math and Computer Science, California State University. He has been involved in industry with Lucent and Zhone Technologies. His current research interests include power consumption and testing of VLSI designs, hardware description languages, FPGAs, computer architecture, and innovative applications of emerging technology.

Wassim El-Hajj received the B.S. degree from the American University of Beirut (AUB), Beirut, Lebanon, in 2000, and the M.S. and Ph.D. degrees from Western Michigan University, all in computer science, in 2002 and 2006, respectively.

He is currently an Assistant Professor with the Computer Science Department, American University of Beirut. He joined the Information Security Department, UAE University, as an Assistant Professor. He then joined the Electrical and Computer Engineering Department, AUB, as a Visiting Professor, and then joined the Computer Science Department as a Full-Time Faculty. He was a frequent speaker and a technical reviewer at more than 20 leading international conferences. He has a valuable industrial experience with Boeing and Ten Strategic Consulting. He received many research grants for his research in information security, data mining, and network planning. His research activities culminated with more than 30 publications, some of which were published in prestigious journals and conferences, such as the IEEE Transactions, BMC Bioinformatics, the IEEE Globecom, and the IEEE ICC.

Dr. El-Hajj is the recipient of numerous recognitions, most notably, the WMU Excellence in Research Award for three years in a row, Honorable Mention for Graduate Research and Creative Scholar, the Outstanding Graduate Student Award for two consecutive years, and the Teaching Effectiveness Award which is considered the Highest Teaching Award with Western Michigan University.