USING SAT & ILP TECHNIQUES TO SOLVE ENHANCED ILP FORMULATIONS OF THE CLUSTERING PROBLEM IN MANETS

Syed Zahidi, Fadi Aloul, Assim Sagahyroon
Department of Computer Science and Engineering
American University of Sharjah, UAE
{b00017408, faloul, asagahyroon}@aus.edu

Wassim El-Hajj
Department of Computer Science
American University of Beirut, Lebanon
we07@aub.edu.lb

ABSTRACT

Improvements over recent years in the performance of Integer Linear Programming (ILP) and Boolean Satisfiability (SAT) solvers have encouraged the modeling of complex engineering problems as ILP. An example 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 enhanced ILP formulations for the Clustering Problem, through the enablement of multi-hop connections and intra-cluster communication, and assesses the performance of state-of-the art generic ILP and SAT solvers in solving the enhanced formulations.

Index Terms — Integer Linear Programming, Boolean Satisfiability, Mobile Ad-Hoc Networks, Clustering Problem, Optimization.

1. INTRODUCTION

The recent introduction of intelligent algorithms in Integer Linear Programming (ILP) and Boolean Satisfiability (SAT) solvers significantly improved the performance of the solvers and allowed for a wide range of challenging engineering problems to be tackled by ILP and SAT. Generic-based ILP solvers have been successfully applied to solve several networking optimization problems; however, fewer 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]. The proposed solution to the scalability issue in *flat* MANET networks is the concept of *clustering*. Clustering involves the creation of a hierarchical network where the network is divided into clusters, with certain nodes in each cluster being chosen to be *clusterheads*. The process of establishing and interconnecting clusters, through the selection of clusterheads and connection of regular nodes to clusterheads is known as the *clustering problem*. The clustering problem can be modeled as an ILP optimization problem. The primary objective of this paper is to present enhancements to the ILP formulation of the clustering problem in MANETs presented in [2]. These enhancements include ILP formulations enabling multihop connections and intra-cluster communication, allowing for more complex network topologies to be generated through ILP. Additionally, this paper presents an evaluation of the performance of the state-of-the-art generic-based and 0-1 SAT-based ILP solvers in handling the proposed enhancements in the ILP formulation of the clustering problem.

This paper is organized as follows. Section 2 presents background information on ILP, SAT, MANETs and the clustering problem. Section 3 describes the existing work done in modeling the clustering problem as ILP. Section 4 describes the proposed enhancements to the ILP formulation of the clustering problem in MANETs. Section 5 presents the tests conducted and an evaluation of the results obtained. The paper is concluded in Section 6.

2. BACKGROUND

This section provides background information on Integer Linear Programming (ILP), Boolean Satisfiability (SAT), Mobile Ad-Hoc Networks and a detailed look at the clustering problem.

2.1. Integer Linear Programming and Boolean Satisfiability

Integer Linear Programming (ILP) involves maximizing or minimizing a function with respect to certain constraints where the objective function and constraints are linear and the used variables can only take integer values [3]. Cases where the integer values are restricted to (0-1) are referred to as Binary ILP Problems. In Boolean Satisfiability (SAT) the constraints between variables are represented using propositional logic. Propositional logic involves the use of AND, OR and NOT operations to construct formulas in the Products-of-Sums form (also called the 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 or prove that no such assignment exists. Advanced decision heuristics and intelligent conflict diagnosis techniques can be used to avoid searching through the entire tree of 2^n variable assignments.

Traditionally, SAT solvers have been used to solve *decision* problems, however, recently SAT solvers have been extended to handle pseudo-Boolean (PB) constraints [4, 5] which are simple inequalities that are equivalent to 0–1 ILP constraints. An advantage of PB constraints is the ability to express *optimization* problems 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 [4, 5]. 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 including: Power [6], FPGA [7], Communications [8], Access Control [9], Cryptography [10], Application Mapping [11], Genetics [12] and Scheduling [13].

978-1-4577-1379-8/12/$26.00 ©2012 IEEE

2.2. 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 [14]. 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 conducted, it was proven that as the number of nodes in flat networks increases, the throughput falls drastically [15]. In addition several factors such as frequent route breakage, unpredictable topology changes, routing overhead make it difficult for a *flat* topology to be scalable [16]. 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 and because of this, the selection of clusterheads is particularly important [17]. There are several issues that need to be considered when selecting clusterheads. Clusterheads are not selected for the lifetime of the network but rather are re-selected at certain intervals. This is because clusterheads are responsible for routing and communication and therefore use more energy than regular nodes. If they remain clusterheads they will be the first nodes to be depleted. In order to maximize network lifetime, the responsibility of being a clusterhead is rotated between nodes. Another reason for re-clustering is that since the nodes are mobile, some nodes may move out of range of one clusterhead and in range of another and so the topology must adjust accordingly.

3. EXISTING ILP FORMULATIONS OF THE CLUSTERING PROBLEM

Relatively few attempts have been made at using using Integer Linear Programming (ILP). The advantage of using ILP over other heuristic based approaches, is that ILP provides the best possible solution, (i.e. the optimum solution) while heuristic based approaches provide a suboptimal approximation. However, since the problem is NP-complete, the complexity increases exponentially with an increase in the number of variables (in case of the clustering problem this would be as a result of an increase in the size of the network).

The first significant attempt at applying ILP formulation to the clustering problem was the work put forward by the authors in [18] in 2006. The authors 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 [18], 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 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. Figure 1 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.

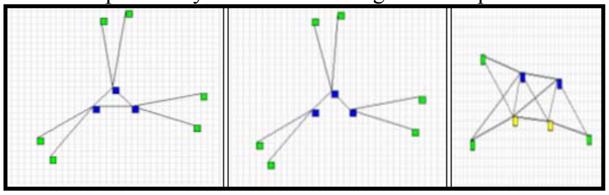

Figure 1. Different network topologies: Fully Connected Backbone, Connected Backbone, and Redundant Models [18].

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. This work represented the first significant ILP formulation of the clustering problem, and provided a platform to enhance significantly. In [2], we proposed an improved ILP formulation of the clustering problem, through the implementation of a Star-Ring Backbone, as shown in Figure 2, reducing the number of redundant links as compared with the EEC-FCB model [18], and avoiding a single point of failure as compared with the EEC-CB model [18]. In addition, an enhancement which enabled the coverage radius of each node to be taken into account was introduced making it possible to generate optimal solutions for heterogeneous networks where nodes had different coverage radii.

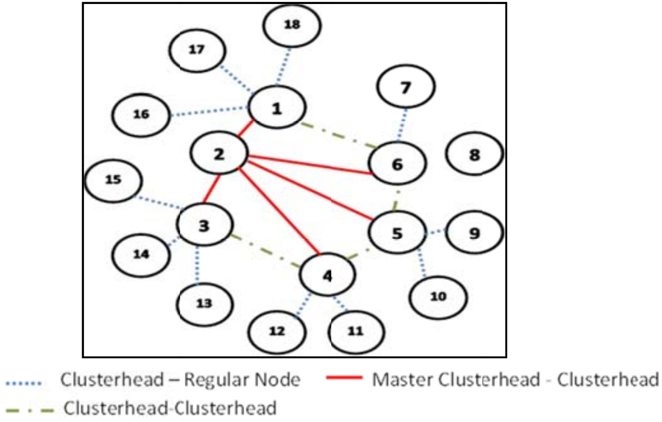

Figure 2. Example topology with a star-ring backbone [2].

The proposed improved formulation was tested with a selection of state of the art, commercial and non-commercial, generic ILP and 0-1 SAT based ILP solvers including CPLEX [19], SCIP [20], BSOLO [21], Pueblo [22] and Minisat+ [23]. It was observed that CPLEX and SCIP performed well, with MINISAT+ being the slowest solver for the presented benchmarks. Among the set of selected solvers, CPLEX and SCIP handled the larger networks well. The SAT solvers BSOLO and Pueblo were very fast for the smaller networks.

4. ILP FORMULATION ENHANCEMENTS

This paper proposes two enhancements to the ILP formulation of the clustering problem presented in [2] which was built on the ideas and assumptions put forward in the EEC-CB model presented in [18]. The proposed enhancements include the ability for nodes to establish multihop connections (2-hop), and also the ability for nodes to communicate with other nodes in the same cluster without going through the designated clusterhead for that cluster (Intra-Cluster communication).

4.1. Variables and Assumptions

The variables used in [2, 18] 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_j: Max number of nodes that can be connected to Clusterhead *j* (CH *j)* (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*; 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
- 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 [2, 18] 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, either through the use of GPS, or other localization techniques.

4.2. 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 is that should a clusterhead fail, the nodes within a cluster will still be able to communicate.

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 [18]. It is the objective function used in the proposed 'Star-Ring' model in [2], with one additional term.

$$Min\,(x,y,z){:}\,(\sum_{i=1}^{N}\sum_{j=1}^{N}c_{ij}x_{ij}+\sum_{j=1}^{N}b_jy_j+\sum_{j=1}^{N}b_jM_j+\sum_{j=1}^{N}\sum_{k=1}^{N}h_{jk}z_{jk} + \sum_{j=1}^{N}\sum_{k=1}^{N}A_{jk}v_{jk} \tag{1}$$

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. The third term represents the cost of selection of the Master clusterhead. 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 $v_{i,j}$ and $f_{i,j,m*}$ are two new variables used when enabling Intra-Cluster connections.

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

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

$m*$ is an index starting from 0, incremented when three conditions (*i≠j, j≠k, i≠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: 7 node network. When considering whether 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).The following constraints are implemented in addition to the constraints used to implement the Star-Ring formulation in [2]. The following constraints enable Intra-Cluster communication. Constraints 2 and 3 are used to identify that node *i* is connected to node *j* if one of the possibilities of the both of them being connected to the same clusterhead has occurred. (*N* = total number of nodes)

$$\sum_{i=1}^{N}\sum_{j=1}^{N}(v_{i,j} - \sum_{\substack{k=1\\ i\neq j\\ j\neq k\\ i\neq k}}^{N} f_{i,j,m*}) \leq 0 \tag{2}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N}(Nv_{i,j} - \sum_{\substack{k=1\\ i\neq j\\ j\neq k\\ i\neq k}}^{N} f_{i,j,m*}) \geq 0 \tag{3}$$

Constraint 4 is used to enforce the restriction that a node cannot connect to itself through a hop.

$$\sum_{i=1}^{N} v_{ii} = 0 \tag{4}$$

Constraint 5 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).

$$\sum_{i=1}^{N} v_{i,j} + v_{j,i} \ = 0 \quad \forall j \tag{5}$$

Constraints 6 and 7 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.

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{\substack{k=1\\ j\neq i\\ j\neq k\\ i\neq k}}^{N} 2f_{i,j,m*} - x_{i,k} - x_{j,k} \ \leq \ 0 \quad \forall j \tag{6}$$

$$\sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N}\sum_{\substack{k=1\\ j\neq i\\ j\neq k\\ i\neq k}}^{N} x_{i,k} + x_{j,k} - f_{i,j,m*} \ \leq \ 1 \quad \forall j \tag{7}$$

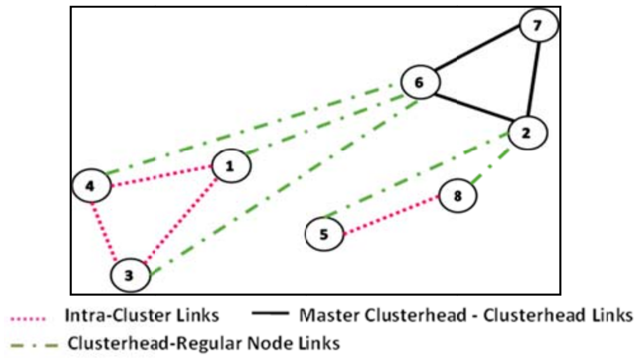
7
6
2
1
4
8
5
3
Intra-Cluster Links
Master Clusterhead - Clusterhead Links
Clusterhead-Regular Node Links

Figure 3. Sample MANET Topology with Intra-Cluster Links

Figure 3 illustrates an example topology with Intra-Cluster communication enabled. As shown, 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.

4.3. 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 Star-Ring base model in [2].

$$\sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}x_{ij} + \sum_{j=1}^{N} a_j y_j + \sum_{j=1}^{N} a_j M_j + \sum_{j=1}^{N}\sum_{k=1}^{N} h_{jk}z_{jk} + \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} B_{ij}b_{ijk} \qquad (8)$$

$B_{i,j}$ represents the cost of connecting node j and node k. This cost is similar to the costs 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 It is similar in that it is again proportional to the distance between the hopping node and the intermediate node used to hop to the clusterhead as shown in Equation 9.

$$B_{j,k} \propto d_{j,k}^{n} \qquad (9)$$

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. A suitable starting value of n can be taken to be 2.5. This value can then be adjusted or tuned through simulation based on how preferred Multihop connections are over direct connections.

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

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

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

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:

Constraint 10 is the updated version of Constraint 6 in [2]. 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.

$$\sum_{\substack{j=1\\ j\neq k\\ j\neq i}}^{N}\sum_{\substack{k=1\\ k\neq i}}^{N} b_{i,j,k} + x_{i,j} \le K \qquad \forall i \qquad (10)$$

Constraint 11 is the updated version of Constraint 5 in [2]. The minimum number of connections that node 'i' should have is 1 if it is a regular node and 0 if it is the master clusterhead.

$$\sum_{i=1}^{N} x_{i,j} + q_{i,j} \ \ge \ 1 - M_j \qquad \forall j \qquad (11)$$

Constraint 12 is the updated version of Constraint 13 in [2] and ensures that the total number of non-backbone connections is equal to N-P. This includes both hop based and direct connections.

$$\sum_{i=1}^{N}\sum_{j=i+1}^{N} x_{i,j} + \ q_{i,j} \ = (N-P) \qquad (12)$$

Constraint 13 is used to ensure that only those nodes that are connected to the clusterhead (x_{ik}=1) can be used as hopping nodes.

$$\sum_{i=1}^{N}\sum_{j=1}^{N}(\sum_{\substack{k=1\\ i\neq j\\ j\neq k\\ i\neq k}}^{N} t_{i,j,m*} - x_{i,k}) \ \le 0 \qquad (13)$$

$m*$ is an index starting from 0, incremented when three conditions are satisfied (i≠j, j≠k, i≠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.

Constraint 14 and 15 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.

$$\sum_{i=1}^{N}\sum_{j=1}^{N}(\sum_{\substack{k=1\\ i\neq j\\ j\neq k\\ i\neq k}}^{N} t_{i,j,m*} + y_j) \ \le 1 \qquad (14)$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N}(\sum_{\substack{k=1\\ i\neq j\\ j\neq k\\ i\neq k}}^{N} t_{i,j,m*} + y_i) \ \le 1 \qquad (15)$$

Constraint 16 is added to ensure that only either 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.

$$\sum_{i=1}^{N} x_{i,j} + q_{i,j} \;\leq\; 1 \quad \forall j \tag{16}$$

Constraints 17 and 18 are used to identify that node *i* is connected to node *j* if it has hopped taken one of the potential hop paths. (*N* = total number of nodes)

$$\sum_{i=1}^{N}\sum_{j=1}^{N} q_{i,j} - \;(\sum_{\substack{k=1\\ i\neq j\\ j\neq k\\ i\neq k}}^{N} t_{i,j,m*}) \;\leq 0 \tag{17}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N} Nq_{i,j} - \;(\sum_{\substack{k=1\\ i\neq j\\ j\neq k\\ i\neq k}}^{N} t_{i,j,m*}) \;\geq 0 \tag{18}$$

Constraint 19 is used to enforce the restriction that a node cannot connect to itself through a hop.

$$\sum_{i=1}^{N} q_{ii} = 0 \tag{19}$$

Constraint 20 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 the *x* connection where the presence of a connection matters.

$$\sum_{i=1}^{N} q_{i,j} + q_{j,i} \;\leq\; 1 \quad \forall j \tag{20}$$

Constraints 21 and 22 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.

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{\substack{k=1\\ j\neq i\\ j\neq k\\ i\neq k}}^{N} 2b_{i,j,k} - x_{i,j} - q_{j,k} \;\leq\; 0 \quad \forall j \tag{21}$$

$$\sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N}\sum_{\substack{k=1\\ j\neq i\\ j\neq k\\ i\neq k}}^{N} x_{i,j} + q_{j,k} - b_{i,j,k} \;\leq\; 1 \quad \forall j \tag{22}$$

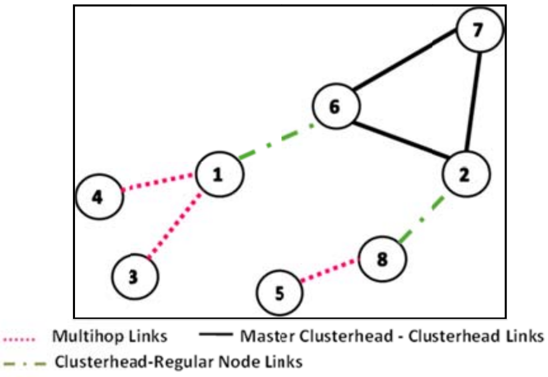
7
6
1
2
4
3
8
5
Multihop Links
Master Clusterhead - Clusterhead Links
Clusterhead-Regular Node Links

Figure 4. Sample MANET Topology with Multihop Connections.

Figure 4 is an example topology with Multihop links enabled. As shown, node 5 connects to clusterhead 2 through node 8. Similarly, nodes 3 and 4 connect to clusterhead 6 through node 1.

5. EXPERIMENTAL RESULTS

Testing was carried out using the following solvers: commercial generic-based ILP solver CPLEX [19], non-commercial generic-based ILP solver SCIP [20], 0-1 SAT-based ILP solvers BSOLO [21], Pueblo [22] and Minisat+ [23]. The SAT solvers were among the winners in recent SAT competitions. All experiments were conducted on an Intel Xeon 3.2 Ghz workstation running Linux with 4 GB of RAM. Testing was carried out for various network tests were generated and solved by the different solvers. The times shown for each network configuration in Table I are the average of the corresponding 100 instances for each network configuration generated with the Star-Ring base model enhanced with Intra-Cluster communication. A timeout of 15 minutes (900 seconds) was set for all solvers. As can be seen from Table I, the SAT solver BSOLO is the fastest among the selected solvers while Pueblo performed well for the smaller scale networks. From Figure 5, it is observed that for a given number of nodes, CPLEX takes a longer time to solve topologies which have greater specified maximum clustersize. This is due to the increased number of intra-cluster links which need to be generated in larger clusters.

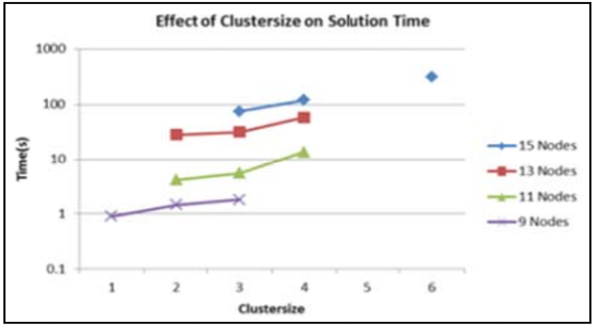

Figure 5. Effect of maximum clustersize on CPLEX solver times

As was the case in [2], with the Star-Ring model, it is observed that Pueblo [22] 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. (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 [18].) Shown in Table II are the results for similar tests conducted with the Star-Ring formulation enhanced with Multihop connections. It is observed that BSOLO performs well for small scale networks while CPLEX is clearly the faster solver for the larger 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. solvers. Overall, it is observed that as in [2], 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. 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 in [2].

TABLE I: SOLVER PERFORMANCE IN SOLVING THE SR+IC ILP FORMULATIONS OF THE CLUSTERING PROBLEM. '-' AND '!' REPRESENTS 'CANNOT SOLVE' AND 'TIMEOUT', RESPECTIVELY.

Network Configurations			**Solver Times (seconds)**				
			Proposed SR Model +Intra Cluster Communication				
# N	***#C H***	***#M CS***	***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**	-	!

TABLE I I: SOLVER PERFORMANCE IN SOLVING THE SR+MH ILP FORMULATIONS OF THE CLUSTERING PROBLEMS. '-' AND '!' REPRESENTS 'CANNOT SOLVE' AND 'TIMEOUT', RESPECTIVELY.

Network Configurations			**Solver Times (seconds)**				
			Proposed SR Model+Multihop Links				
# N	***#C H***	***#M CS***	***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	!	-	!

6. CONCLUSION

This paper proposes two enhancements to the ILP formulation developed in [2]. 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 analyze the performance of generic-based ILP and 0-1 SAT-based ILP solvers. The SAT solver, BSOLO, performed well for small scale networks while the generic-based ILP solvers CPLEX and SCIP were able to handle the larger scale topologies without timing out. 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.

REFERENCES

[1] S. Chinara and S. Rath, "Energy Efficient Mobility Adaptive Distributed Clustering Algorithm for Mobile Ad Hoc Network," in *International Conference on Advanced Computing and Communications*, 2008.

[2] S. Zahidi, F. Aloul, A. Sagahyroon and W. El-Hajj, "On using ILP in solving the clustering problem in MANETs," in *Int'l Conf. on Information Science, Signal Processing and their applications*, [Accepted], 2012.

[3] A. Schrijver, "Theory of Linear and Integer Programming," *John Wiley & Sons Ltd*., 1999.

[4] F. Aloul, A. Ramani, I. Markov and K. Sakallah, "Generic ILP versus specialized 0–1 ILP: an update," in *International Conference on Computer Aided Design* (ICCAD), 450–457, 2002.

[5] D. Chai and A. Kuehlmann, "A fast pseudo-Boolean constraint solver," in *Design Automation Conference* (DAC), 2003.

[6] A. Sagahyroon, F. Aloul, and A. Sudnitson, "Using SAT-Based Techniques in Low Power State Assignment," in *Journal of Circuits, Systems, and Computers* (JCSC), 20(8), 1605-1618, 2011.

[7] A. Ling, D. Singh and S. Brown, "FPGA Programmable logic block evaluation using quantified Boolean Satisfiability," in *IEEE Computers and Digital Techniques*, 153(3), 165-172, May 2006.

[8] F. Aloul and M. El Tarhuni, "Multipath Detection Using Boolean Satisfiability Techniques," in *Journal of Computer Networks and Communications* (JCNC), vol. 2011, Article ID 365107, 2011.

[9] G. Hughes and T. Bultan, "Automated Verification of Access Control Policies using a SAT solver," in *International Journal on Software Tools for Technology Transfer* (STTT), 10(6), 503-520, October 2008.

[10] Il. Mironov and L. Zhang, "Applications of SAT solver to Cryptanalysis of Hash Functions," in *SAT*, 102-115, 2006.

[11] S. Tosun, O. Ozturk, and M. Ozen, "An ILP formulation for application mapping onto Network-on-Chips," in *AICT*, 2009.

[12] A. Graca, I. Lynce, J. Marques-Silva and A. Oliveira, "Haplotype inference with pseudo-Boolean optimization," in *Annals of Operations Research*, 184(1), 137-162, 2011.

[13] A. Wasfy and F. Aloul, "Solving the University Class Scheduling Problem Using Advanced ILP Techniques," in *GCC Conference*, 2007.

[14] J. Garcia-Macias and J. Gomez, "MANET versus WSN," in *Sensor Networks and Configuration: Fundamentals, Standards, Platforms, and Applications*. Springer, ch. 17, 369-388, 2006.

[15] K. Xu, X. Hong and M. Gerla, "An Ad Hoc Network with Mobile Backbones," in *IEEE International Conference on Communications*, 2002.

[16] J. Al-Karaki, A. Kamal and R. Ul-Mustafa, "On the Optimal Clustering in Mobile Ad hoc Networks," in *Consumer Communications and Networking Conference*, 2004.

[17] H. Safa, O. Mirza and H. Artail, "A Dynamic Energy Efficient Clustering algorithm for MANETs," in *IEEE Int'l Conference on Wireless & Mobile Computing, Networking and Communnication,* 2008.

[18] W. El-Hajj, D. Kountanis, A. Al-Fuqaha and H. Harbi, "Optimal Hierarchical Energy Efficient Design for MANETs," in *Int'l Conference on Communications and Mobile Computing*, 287-292, 2006.

[19] IBM ILOG CPLEX Optimizer, 2011. [Online]. *http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/*

[20] SCIP: An Open Source MIP Solver and constraint integer programming framework, 2011. [Online]. *http://scip.zib.de/*

[21] V. Manquinho and J. Marques-Silva. "On using cutting planes in pseudo Boolean optimization," in *JSAT*, 2:209–219, 2006.

[22] H. Sheini and K. Sakallah, "Pueblo: A Hybrid Pseudo-Boolean SAT Solver," in *Journal on Satisfiability, Boolean Modeling and Computation*, 2(1-4), 165-189, 2006.

[23] N. Een and N. Sorensson, "An extensible SAT-solver," in *Int'l Conf. on Theory and Applications of Satisfiability Testing*, 502–508, 2003.