DESIGN FRAMEWORK FOR HIERARCHY MAINTENANCE ALGORITHMS IN MOBILE AD HOC NETWORKS

Kyriakos Manousakis, John S. Baras
Institute for Systems Research
University of Maryland
College Park, MD, 20742
USA

and

Anthony J. McAuley, Raquel Morera
Telcordia Technologies
Piscataway, NJ, 08854
USA

ABSTRACT(1)

Domain autoconfiguration techniques allow the quick formation of highly optimized hierarchies that greatly enhance network scalability and overall performance. For example, instead of producing a simple two level hierarchy based only on topology, the optimization can produce multi-level hierarchies that take into account factors such as mission goals and predicted node/link heterogeneity. However, in dynamic networks, such as expected in the future military networks, these highly optimized solutions degrade very quickly. Indeed, if we use standard local maintenance algorithms that do not align well with the optimization goals, then the performance can reach the level of a suboptimal solution in less than two minutes. This paper proposes a taxonomy of local maintenance algorithms into four basic classes and quantifies the performance benefits of using representative approaches that act in accordance with the optimization goals.

INTRODUCTION

The mobile ad hoc networks (MANETs) require no fixed infrastructure, making them ideal for many commercial, emergency and military scenarios. An open question, however, is the ability of MANET networks to scale. Even for protocols (routing, security, QoS) designed specifically for these dynamic environments, when the size of the network becomes too large then these protocols either fail to capture the network dynamics or swamp the network in signaling traffic [1] [2] [3] [4] [5].

Although some protocols can scale to hundreds or even thousands of nodes in certain conditions, in general network scalability has always relied on the generation of hierarchy. For example, the wireline world divides networks into subnets and Autonomous Systems. The affect of hierarchy can be dramatic. For example, in theory, domains can reduce the overall routing protocol overhead with n nodes from $O(n^2)$ to $O(n\log n)$. Network hierarchy allows the applied protocols to operate on smaller subgroups of the network and not on the entire network. Thus, the protocols can handle better the dynamics of smaller groups of nodes. Hierarchy also allows protocols to be tuned to more homogenous conditions. The benefits of a good hierarchy have been shown to outweigh the complexity [6].

In order to cope with the rapid deployment and rapid reconfiguration required for future military networks, this creation of the domains must be done automatically. Moreover, in mobile ad hoc networks (MANETs), such as the FCS (in a Unit of Action) or WIN-T (in a Unit of Employment), there is a need for mechanisms that not only automatically create such hierarchies but also maintain them as the network changes.

The next section overviews the domain generation framework. Section 3 provides the taxonomy of the various local maintenance approaches and presents the characteristics of the specified classes. In section 4 we evaluate the effect of the various local maintenance approaches on the quality of the optimized generated hierarchy. Finally, section 5 highlights the important conclusions drawn.

DOMAIN GENERATION FRAMEWORK

Our hierarchy generation framework [7] organizes the network into domains by taking into consideration the global network environment. As the purpose of each network can be different, we also require the hierarchy to adaptively boost whatever are the network's key performance requirements.

A. Overview of Simulated Annealing Algorithm

Our approach is based on a modified version of a global optimization algorithm, namely the Simulated Annealing (SA) algorithm. SA has been applied for the solution of many combinatorial optimization problems, such as graph partitioning [8] [9]. The hierarchy generation objectives are expressed as cost functions, which upon their optimization (minimization, maximization) generate the desired hierarchical structures [10].

(1) Prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance (CTA) Program, Cooperative Agreement DAAD19-2-01-0011. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

As part of our hierarchy generation framework, the basic objective of the algorithm is to obtain an optimal hierarchy C^* of K domains with respect to a set of pre-specified hierarchy generation objectives. A typical feature of SA is that, towards the optimization process, besides accepting improvements in cost, it also to a limited extent accepts probabilistically deteriorations in cost. This extent depends on the value of the control parameter c_t which is the analogy of the temperature in the physical annealing process. The acceptance of worse moves is important for the algorithm to avoid local minima (maxima), which might result in the formation of low "quality" hierarchical structures.

In general, SA starts from a large value of the control parameter c_0, such that almost every move gets accepted. The value of the control parameter is cooled down (decreases) carefully with respect to a cooling schedule. In every iteration a new hierarchy C' is generated with a small perturbation on the currently optimal one C_t^*. The difference in their costs is $\Delta E = E^* - E'$. E^* is the cost of the currently optimal solution and E' is the cost of the new generated solution. In case of minimization, the new hierarchy C' is accepted as the new currently optimal $C_{t+1}^* \leftarrow C'$ with respect to Metropolis criterion:

$$P_{c_{t+1}}\left(C_{t+1}^* \leftarrow C'\right) = \begin{cases} 1 & \text{if } \Delta E > 0 \\ \exp\left(\frac{\Delta E}{c_{t+1}}\right) & \text{if } \Delta E \le 0 \end{cases}$$

SA algorithm terminates when the stop criterion is satisfied.

The SA weakness is its practical slow convergence time. By adjusting the various characteristics of SA, we found we could trade a small loss in optimality for over 100x reduction in convergence times [7]:

- Less than 1ms for 100 nodes
- Less than 20secs for 1000 nodes networks.

B. Hierarchy Generation Objectives

There are many hierarchy generation objectives, expressed in the form of cost functions [7] [10]. The first class of objectives, we have introduced, has to do with the structural characteristics of the generated domains. For example:

- Balanced Size or Diameter.
- Minimum number of Border Routers.
- Minimizing stretch due to hierarchical routing.

The second class of objectives had to do with the mobility characteristics of the participating nodes. For example

- Grouping together nodes with similar mobility characteristics (e.g., speed, direction)
- Grouping together links which have been estimated to remain active for long periods of time.

Interesting cost functions generally combine multiple objectives (e.g. either topological or mobility objectives). Upon the optimization of the corresponding multi-objective cost functions, the generated hierarchies simultaneously satisfy all of the requested objectives.

C. Topological Constraints

The hierarchy must satisfy certain topological constraints (e.g. create feasible hierarchies). In particular we want every node within a domain to be able to reach all other members of the same domain only by utilizing intra-domain links. From this isolation we can take full advantage of the aggregation and abstraction provided from the application of hierarchy (spatial reuse in the assignment of codes, minimization of control and communication information).

HIERARCHY MAINTENANCE SCHEMES

Even though, the domain generation framework we have introduced is capable of constructing optimized hierarchical structures, since the network environment under consideration is dynamic, these structures will soon become sub-optimal and infeasible (the topological constraints may not be satisfied). It would be inefficient and expensive to apply the global hierarchy generation mechanism on every topological change. Thus localized reaction is preferred for maintaining the hierarchical structure. Apart from the faster reaction, the maintenance phase should be capable of maintaining the "quality" of the generated hierarchy. This will prolong the generation mechanism's reapplication interval for as much as possible resulting in lower overhead and more efficient utilization of the network resources. In this work we have categorized and studied the impact of various local hierarchy maintenance schemes on the preservation of the hierarchical structures "quality".

A. Taxonomy of Local Maintenance (LM) Schemes

The main trade off in distributed maintenance is between overhead and quality. We identified four classes of LM schemes:

- **A0**: Zero Overhead Local Maintenance
- **A1**: Objectives Independent Local Maintenance
- **A2**: Node Dependent Local Maintenance
- **A3**: Domain Dependent Local Maintenance

Figure 1 provides the LM schemes taxonomy we introduced in this study with respect to the amount and quality (i.e. relevance to the generation objectives) of information involved in the maintenance decisions.

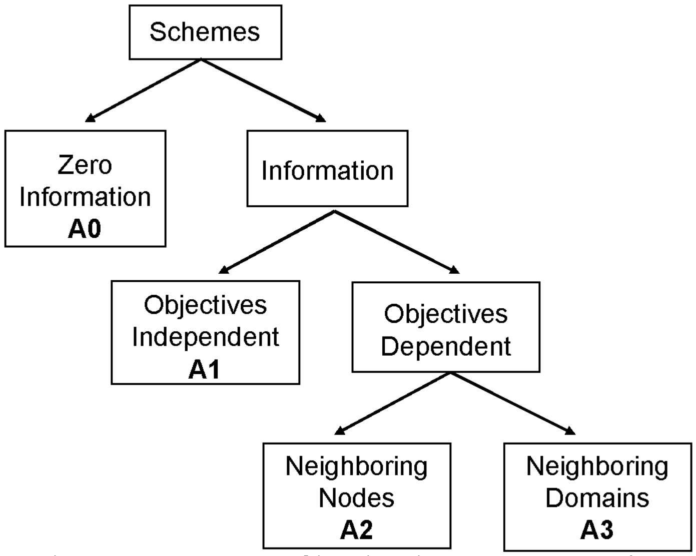

Figure 1. Taxonomy of local maintenance approaches

In general, it is expected that the more relevant to the generation objectives is the available information (better quality metrics) during the maintenance phase, the better the "quality" of the hierarchical structure is preserved. For example if the hierarchy generation objective is the construction of robust to mobility domains, then the local maintenance is better to utilize metrics related to the speed, direction and position of the participating nodes for the reconstruction of the hierarchy.

In general the maintenance method is triggered locally by the nodes that become infeasible (e.g. the nodes lose connectivity to their original domains) due to the topological changes. A brief overview of the LM classes' characteristics is:

- **A0. Zero Information Local Maintenance** This approach does not require any information to be collected from the network for the reconstruction of the hierarchical structure. The approach in terms of overhead is optimal, since it does not utilize any bandwidth resources.
- **A1. Objectives Independent Local Maintenance** The schemes of this approach collect and utilize local information for the reconstruction of the hierarchical structure. The information, however, is unrelated to the metrics that has been utilized from the hierarchy generation mechanism for the construction of the optimized hierarchical structure. For example when the generation objectives enforce the formation of robust to mobility domains, the speed and direction of nodes is required so that are grouped based on their mobility similarities. Whereas during the maintenance the nodes may have access only to information unrelated to the mobility characteristics of the neighboring nodes (e.g. IDs of the neighboring nodes).
- **A2. Node Dependent Local Maintenance.** This approach (A2), as opposed to the previous two, is aware of the hierarchy generation objectives and the corresponding schemes attempt to maintain the "quality" of the generated hierarchy by utilizing metrics related to these objectives. However, the maintenance decisions are based on information gathered only from the immediate neighbors (e.g. one hop neighbors).
- **A3. Domain Dependent Local Maintenance.** Like scheme (A2), A3 utilizes information (metrics) relevant, to the hierarchy generation objectives, for maintaining the "quality" of the hierarchical structures. Whereas, unlike A2, this approach bases its restructuring decisions on information collected from the entire neighboring domains. Clearly, this approach requires the most overhead.

B. Representative Examples

This section provides representative schemes from each of the four hierarchy maintenance classes we defined above.

- **A0. Random**. As its name reveals, the random maintenance mechanism is probabilistic. Specifically, the nodes seeking to join a domain, randomly select one of available neighboring domains by utilizing the uniform distribution. If V^k is the set of neighboring domains C_i of node k defined as:

$$V^k = \left\{ C_i : \exists j \in C_i \text{ s.t. } j \xleftrightarrow{\text{1 hop}} k \right\}$$

then node k selects a domain C_i with probability $p_k(C_i)$, where

$$p_k(C_i) = \frac{1}{|V^k|} \tag{1}$$

- **A1. Lowest ID (LID)**. The lowest ID (LID) scheme requires that each node has a unique ID. The ID of the node i with the lowest ID among the nodes of the same domain C_i defines also the ID of this domain. When a node k seeks to join a new domain, it selects the domain C_i^* with the lowest ID from the set V^k of its candidate neighboring domains.
- **A2. Node Dependent Cost Function**. This scheme uses similar to the generation phase metrics, which have been collected from the immediate (one hop) neighboring nodes. For example if the hierarchy generation objective is to construct robust to mobility

domains by grouping together nodes with similar mobility characteristics (speed, direction); a node seeking new domain applying a LM scheme from A2 class will join the same domain as its neighboring node with the more similar mobility characteristics (speed, direction).

- **A3. Domain Dependent Cost Function**. This scheme uses similar to the generation phase metrics as the previous scheme but these metrics have being collected from the entire neighboring domains and not just from the immediate neighboring nodes.

EVALUATION OF HIERARCHY MAINTENANCE SCHEMES

To provide a basic understanding, this section uses example metrics and cost function to evaluate the cost of the maintained hierarchical structure for each of the four approaches given in the previous section. Generalizations of these observations are shown in the following section.

A. Representative Cost Function and Network

Consider as hierarchy generation objective the construction of robust to mobility domains by grouping together nodes of similar mobility characteristics. In the hierarchy generation phase the domains were formed by applying SA to optimize cost function (2).

$$J(C) = \min_{C}\left(\sum_{z=1}^{K}\left[\sum_{i,j=1}^{|C_z|} U^2_{r^z_{i,j}}\right]^2\right) \quad (2)$$

where,

C_i: Cluster i

$|C_i|$: Size of cluster i

$U_{r_{i,j}}$: Relative Velocity of nodes i,j

The relative velocity $U_{r_{i,j}}$ of two nodes i, j is defined from (3), (4) and (5).

$$U_{r_{i,j}} = \sqrt{U^2_{X_{i,j}} + U^2_{Y_{i,j}}} \quad (3)$$

$$U_{X_{i,j}} = S_i \cos\theta_i - S_j \cos\theta_j \quad (4)$$

$$U_{Y_{i,j}} = S_i \sin\theta_i - S_j \sin\theta_j \quad (5)$$

where,

S_i: Speed of node i

θ_i: Direction of node i

Assume the resulting optimized hierarchy of Figure 2. Due to mobility, node 11 modifies the topological structure of the network and seeks to join a neighboring domain. Such an event triggers the maintenance phase. By applying the representative schemes, introduced above, we can evaluate their impact on the "quality" of the maintained hierarchical structure.

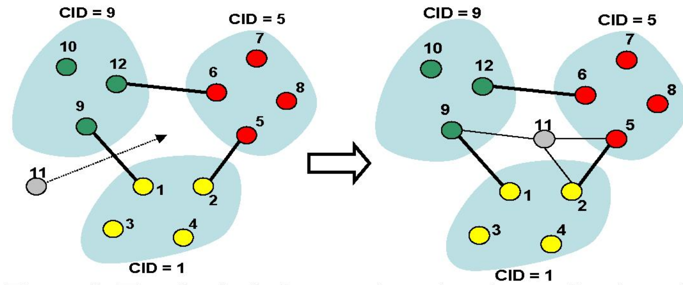

Figure 2. Topological change triggering the application of local maintenance

Assume also that the mobility metrics - speed (Sp) and direction (Dr) - of the nodes are provided from Table 1.

Table 1. Mobility characteristics of the nodes

ID	Sp	Dr	ID	Sp	Dr	ID	Sp	Dr
1	0	0	5	4	45	9	3	45
2	0	0	6	5	60	10	3	30
3	0	0	7	5	45	11	4	45
4	0	0	8	6	60	12	2	30

B. Application of the Local Maintenance Schemes

Figure 3 shows the variety of the selections made by Node 11 (from Figure 2) after applying the representative schemes from each one of the four LM classes:

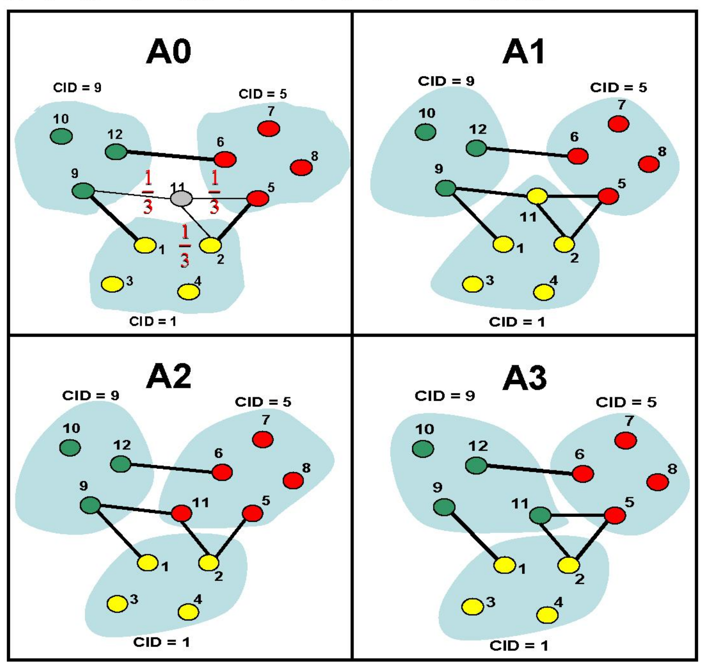

Figure 3. Hierarchy generated by the LM schemes

- **A0**: Node 11 detects three neighboring domains, and will decide to join randomly one of these. The probabilities of Node 11 (p_{11}) joining domain C_i are:

$$p_{11}(C_1) = p_{11}(C_2) = p_{11}(C_3) = \frac{1}{3}$$

- **A1**: Node 11 will decide to join domain C_1 because it has the lowest ID among its neighboring domains:

$$C_1 = 1, C_2 = 5, C_3 = 9$$

- **A2**: With respect to speed and direction values given in Table 1, Node 11 has speed 4m/s and direction of 45 degrees. The neighboring nodes of Node 11 and their corresponding domains are represented from the following (node ID, domain ID) pairs:

$$(2, C_1 = 1), (5, C_2 = 5), (9, C_3 = 9)$$

  Node 5 presents the more similar mobility characteristics to Node 11. Hence, Node 11 selects the host domain of Node 5 $(C_2 = 5)$.

- **A3**: Node 11 collects the appropriate metrics (e.g. speed and direction) from each one of the nodes lying in its neighboring domains. Using function (2), Node 11 evaluates the cost of each one of the possible maintained structures. The computed costs for each case are provided from the following (domain ID, cost) pairs:

$$(C_1 = 1,\ 81.77), (C_2 = 5,\ 26.77), (C_3 = 9,\ 25.13)$$

  According to the above (domain ID, cost) pairs, Node 11 selects to join domain $(C_3 = 9)$, which will result in the hierarchical structure with the lowest cost.

C. Comparison of the Four LM Schemes

On the example above, each scheme results in a different hierarchical structure with different cost ("quality"). Table 2 below provides the cost of the resulting hierarchies from each LM scheme we applied.

Table 2. Cost of the hierarchy after applying the various LM schemes

Approach	Cost
A1. Objectives Independent (LID)	$C_{A1} = 81.7689$
A2. Node Dependent (A2)	$C_{A2} = 26.7673$
A3. Domain Dependent (A3)	$C_{A3} = 25.1318$
A0. Zero Information (Random)	$C_{A1} \vee C_{A2} \vee C_{A3}$

Some important observations are:

1. The highest quality (lowest cost) maintained hierarchy was established from approach A3. Approach A3 is expected to perform the best, because it takes into consideration larger amount of information, which is of better quality. The weakness of this class of maintenance schemes is that they require larger overhead for the collection of the appropriate amount of metrics. Whereas, the quality of the maintained hierarchy compensates for this drawback.
2. Even though, the Random scheme of approach (A0) does not use any metrics for the maintenance (zero overhead), it is statistically expected to perform better than the LM schemes of approach A1 (i.e. LID), with respect to the quality of the maintained hierarchy.

IMPACT OF LM MAINTENANCE SCHEMES ON DOMAIN QUALITY

This section justifies that the impact of the LM schemes on the maintained hierarchy cost for the above example is the common case for their performance. We use two cost functions from [7] to construct optimized hierarchies. Then, for a pre-specified amount of time, we applied the various LM schemes and we were evaluating the cost of the maintained hierarchy.

A. Impact on "Balanced Size" Domains

On a network of 100 nodes we generated 10 domains using the SA-based hierarchy generation mechanism. The objective was to construct "balanced size" domains. By optimizing (minimizing) cost function (6) (see [7]):

$$J(C) = \min_{C}\left(Var\left(|C_1|^2, \ldots, |C_K|^2\right)\right) \qquad (6)$$

we obtained 10 domains of 10 nodes each. Then, we applied, the representative LM schemes of approaches (A0), (A1) and (A3) for 500 seconds of network time on the optimized hierarchy.

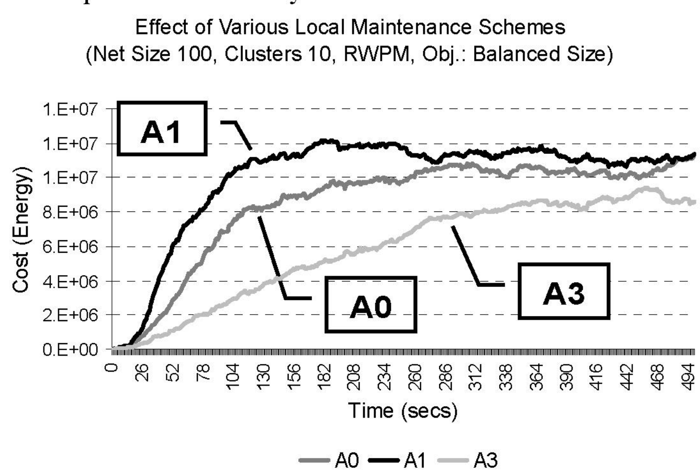

Figure 4. Impact of three maintenance approaches on the "balanced size" domains

Figure 4 gives the average cost per second (out of 100 experiments) of the maintained hierarchy. The topology was changing every second with respect to Random Waypoint Mobility (**RWPM**) model, with maximum speed 10m/s and no pause time. Every second we were evaluating the cost of the maintained hierarchy using cost function (6).

Although scheme (A3) requires the most overhead, it performs the best on preserving the quality (cost) of the hierarchy. Interestingly, the Random scheme (A0) maintains better the quality of the hierarchy compared to LID scheme (A1), even though does not require any overhead.

B. Impact on "Robust to Mobility" Domains

In a second set of experiments we generated 6 domains in a network of 100 nodes. This time, upon its optimization from SA, cost function (2) grouped together nodes with similar mobility characteristics. After obtaining the optimized hierarchical structure, we applied the four representative maintenance approaches for 250 seconds of network time. The topology of the network was changing every second with respect to Reference Point Group Mobility (RPGM) model [11] (we predefined 6 groups of nodes with distinctive mobility characteristics, so the cost function applied had to locate the various mobility groups). Figure 5 gives the average cost per second (out of 100 experiments) for the various LM schemes.

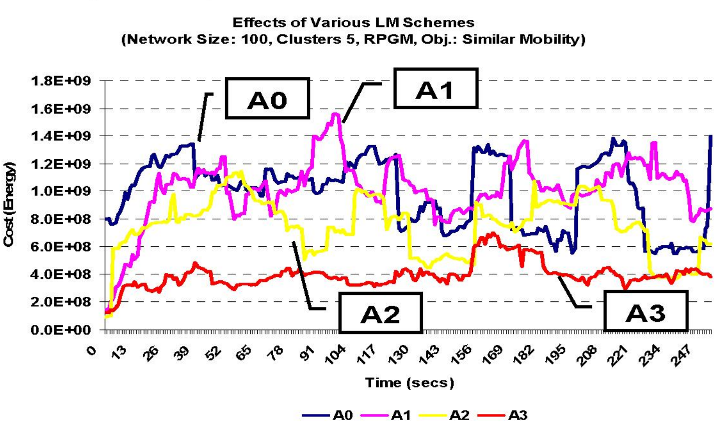

Figure 5. Impact of maintenance approaches on the "robust to mobility" domains

As in the previous scenario, approach (A0) on average performs better than (A1), but both of them perform worse than (A3). Also, by comparing (A2) with (A3), (A3) performs better (as expected due to the larger amount of information it utilizes for the maintenance decisions).

CONCLUSIONS

This paper presents a taxonomy of LM schemes, depending on: a) whether they are aware of the hierarchy generation objectives utilized during the generation phase and b) the amount of information available to them. We show that by ignoring the importance of the LM algorithm, the hierarchy may end up harming the performance of the network instead of improving it. The maintenance algorithm has to be designed in accordance to the performance objectives required. The most commonly used approach applied today, the Lowest ID approach (A1), consistently performs the worst. Better for both quality and overhead is a Random Approach (A0). However, tailoring the maintenance to the hierarchy generation objectives consistently preserves the quality of the hierarchy. Furthermore, the larger the amount of the relevant information available (A3), the better the maintenance and the more is prolonged the interval for the reapplication of the expensive SA-based generation mechanism This longevity is critical to maintaining the sort of effective and powerful network needed to support future military needs.

REFERENCES(2)

[1]. R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," IEEE Journal on Selected Areas in Communications, pp. 1265-1275, September 1997

[2]. Rajesh Krishnan, Ram Ramanathan and Martha Steenstrup, "Optimization Algorithms for Large Self-Structuring Networks, Proceeding of IEEE INFOCOM '99, New York, USA, March 1999.

[3]. S. Basagni, "Distributed and Mobility-Adaptive Clustering for Multimedia Support in Multi-Hop Wireless Networks," Proceedings of Vehicular Technology Conference, VTC 1999-Fall, Vol. 2, pp. 889-893

[4]. M. Chatterjee, S. K. Das, D. Turgut, "WCA: A Weighted Clustering Algorithm for Mobile Ad hoc Networks, " Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks), Vol. 5, No. 2, April 2002, pp. 193-204

[5]. R. Morera, A. McAuley, "Flexible Domain Configuration for More Scalable, Efficient and Robust Networks" MILCOM, October 2002.

[6]. K. Manousakis, J. McAuley, R. Morera, J. Baras, "Routing Domain Autoconfiguration for More Efficient and Rapidly Deployable Mobile Networks," Army Science Conference 2002, Orlando, FL

[7]. K. Manousakis, A. J. McAuley, R. Morera, "Applying Simulated Annealing for Domain Generation in Ad Hoc Networks," International Conference on Communications (ICC 2004), Paris, France, 2004.

[8]. S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, Optimization by Simulated Annealing, Science 220 (13 May 1983), 671-680

[9]. D. S. Johnson and L. A. McGeoch, "The Traveling Salesman Problem: A Case Study in Local Optimization," in E. H. Aarts and J. K. Lenstra (eds.), "Local Search in Combinatorial Optimization," John Wiley and Sons, Ltd., pp. 215-310, 1998

[10]. K. Manousakis, J.S. Baras," Dynamic Clustering of Self Configured Adhoc Networks Based on Mobility ," in the Proceedings of Conference on Information Sciences and Systems, Princeton,NJ,March 2004

[11]. X. Hong, M. Gerla, G. Pei, and C. Chiang. A group mobility model for ad hoc wireless networks. In Proceedingsof the ACM International Workshop on Modeling and Simulation of Wireless and Mobile Systems (MSWiM), August 1999.

(2) The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied of the Army Research Laboratory or the U.S. Government.