POSITIONING RANGE EXTENSION GATEWAYS IN MOBILE AD HOC WIRELESS NETWORKS TO IMPROVE CONNECTIVITY AND THROUGHPUT

Mohiuddin Ahmed, Son Dao, Randy Katz

Abstract— **The dynamic nature of a mobile ad hoc network (MANET) may result in a cluster of nodes being isolated from the remaining network, especially when deployed in a terrain with blockages. In order to facilitate scalability and to provide connectivity between partitions that might occur in wireless networks as a consequence of mobility, we can envision a 'range extension' network that consists of airborne communication platforms, or geostationary/low-earth-orbit satellites. These airborne/satellite nodes will maintain communication links with specific 'gateway' nodes among the mobile ground nodes. To communicate with a node that is geographically distant or belongs to a different network partition, an ad hoc node can relay its data packets through an appropriate mobile gateway and via the range extension network.**

If we envision that the MANET is divided into different groups and a mobile gateway is deployed for each such group, an objective then, is to determine the trajectory of the mobile gateway to best serve the ad hoc group to which it belongs to, in terms of network performance metrics such as throughput and latency. In this paper, this problem of computing the optimal position for a gateway is reduced to a linear optimization problem by means of some simplifying but realistic assumptions. We suggest methods that may be deployed to enable the gateway to follow this optimal trajectory as closely as possible (within the practical constraints imposed by its velocity and maneuverability). Simulation results for various scenarios show a 10-15% improvement in the throughput and latency, per gateway domain, if a gateway has a dynamic trajectory whose locus follows the computed optimal position, as compared to a gateway that is statically placed at a regular position, or to a gateway that has a random trajectory.

Keywords— **mobile ad hoc networks, gateway, convex optimization, trajectory control.**

I. INTRODUCTION

MOBILE ad hoc networking technology ([1]) may be appropriate for linking mobile computers in an office or home environment, deploying wireless sensors in remote or inhospitable terrain, coordinating disaster relief efforts after natural catastrophes, or in tactical deployments for situation awareness applications [2]. A major challenge in the wide deployment of MANETs has been in achieving scalability. Furthermore, due to the range limitations of ad hoc nodes, the network might often be geographically divided into isolated partitions. In order to achieve scalability in terms of efficient communications between geographically distant nodes or between nodes that belong to different isolated partitions (each of which is an ad hoc group by itself), it is desirable to provide a supporting infrastructure in the form of a range extension network. This infrastructure is also essential to interface the MANET with the Internet.

This range extension network could typically consist of airborne relay nodes or low earth orbit/geostationary satellites. In order to interface the ad hoc network with the range extension network, one can envision the deployment of special gateway nodes in the ad hoc network. These are 'on-ground' nodes that might be more power/processing capable than the other ad hoc nodes on the ground and equipped with the appropriate hardware for communicating with the satellite/airborne nodes. This architecture can therefore be visualized to consist of two layers. The first layer includes the ad hoc network, and the second includes the range extension network consisting of satellites or airborne nodes. The mobile gateway provides the interface for the communications between the two layers and hence we shall call a gateway node a *Cross Layer Communication Agent* or CCA from this point onwards. Similar architectures have been previously considered for enabling hierarchical routing or multicasting ([3]–[5]). In contrast, our objective is the determination of the CCA trajectory that a mobile CCA has to follow in order to optimize inter-domain network performance.

If the CCA is affiliated with an ad hoc team of mobile units, the question arises as to where the CCA ought to be located relative to the other mobile nodes. We design a methodology for defining the CCA trajectory based upon the location, loading, etc. of the other nodes in the ad hoc group that the CCA serves. We show that network performance improves (for communication involving nodes in different clusters of nodes), in terms of throughput and latency, if the CCA trajectory is computed based on our methodology.

We derive a relatively simple analytic formulation for the optimal CCA position, which is equivalent to a linear optimization problem. This is discussed in Section III. We also provide an algorithmic implementation of the formulation, and discuss the effect of some of the parameters in this section. In Section IV, we estimate the overhead and the computational complexity for implementing this architecture with the aid of typically used media access control (MAC) and routing protocols. In Section V, we discuss our simulation framework and some preliminary results. We conclude in the final section.

II. SYSTEM MODEL

We motivate the discussion of the system architecture in the context of the following scenario. Consider separate groups of mobile ad hoc nodes operating in a terrain with blockages and deployment area restrictions (e.g. troop divisions deployed in a mountainous area). Each group would have one or more CCAs capable of communicating with an airborne or satellite node with which it has a direct line-of-sight connection. As an example, in Figure 1, we have considered two isolated groups or *domains* of mobile nodes, each forming a MANET by themselves, and each having its own CCA. The CCA in each domain is then the conduit via which the ad hoc nodes in the separate domains can send data packets to each other, with the routing assistance of the airborne node. The airborne node can also serve to connect the clusters to a wired infrastructure (e.g. command and control centers outside the theater of operations).

As the nodes in a particular domain move, the objective then

Mohin Ahmed {mohin@hrl.com} and Son Dao {skdao@hrl.com} are with the Information Sciences group at HRL Laboratories, LLC., in Malibu, CA. Randy Katz {randy@cs.berkeley.edu} is with the Computer Science Department at the University of California, Berkeley, CA.

This work was supported by Defense Advanced Research Projects Agency (DARPA) Contract No. N00014-99-C-0322.

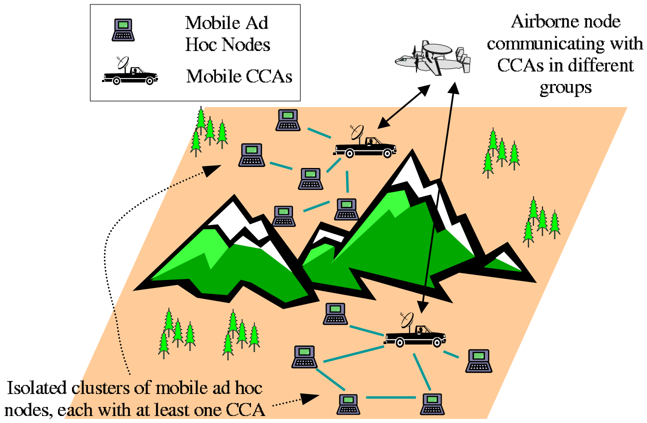

Fig. 1. Ad hoc network of two groups of mobile nodes and CCAs.

is to specify the 'optimal' trajectory for the mobile ground CCA associated with that group. For communication intended for nodes within a given group, the nodes would not be compelled to use the CCA, but would instead rely on the underlying MANET architecture using traditional routing, MAC protocols, etc. By intelligently positioning the CCA, we might expect to achieve better network performance for *inter*-domain node communications, (i.e. data communication between nodes that are in the domains of different, possibly geographically isolated CCAs) than if the CCA were allowed to move randomly with respect to the nodes it its domain.

We can assume any suitable mobility model for depicting the motion of the ad hoc nodes [8], e.g. random waypoint motion, or static sensor nodes, etc. Additionally, the terrain in which the MANETs are deployed may contain regions where either the node or the CCA, or both, cannot reside (e.g. hazard zones, regions of radio obscurity, etc.). These are collectively referred to as *blockages*

Given this scenario, the performance metrics that could potentially be improved by optimizing the CCA trajectory include, but may not be limited to, the following:

- *inter-domain* network data throughput
- *inter-domain* network packet transport delay
- total power expended (or maximum power consumed per packet/bit per node in the CCA)
- data transmission reliability (packet drop/error rate)
- volume of the network control messages and resulting signaling overhead.

The procedure by which the weighted centroid is computed requires the following:

1. Each node is equipped with a GPS device that enables the node to determine its position.
2. Each node can estimate its offered load in real time
3. Terrain information (such as specific coordinates or boundaries of the domain, radio null regions, etc.) is available at each CCA. This can easily be made available at network inception[1].

The details of the actual communication mechanisms that enable the MANET to function are not directly relevant in the development of our analytical formulation for computing the optimum trajectory that our CCA ought to follow (Section III). For *intra*-domain node communications (i.e. communication between nodes that are in the domain of the same CCA), the MANET could rely on well-established protocols such as the IEEE 802.11 MAC protocol for media access control [6], DSR, DSDV, or AODV for routing [7], etc. to establish and maintain connectivity. For *inter*-domain node communications, data will have to be routed through the CCA and via the range extension network.

III. CCA Trajectory Update Algorithm: Formulation and Analysis

In this section we describe the algorithm for determining the trajectory of mobile CCAs such that it is optimal in terms of 'relative position' with respect to the group of ad hoc nodes that it serves. We describe our algorithm assuming that there is a single CCA per domain. However, it is possible for several nodes among a cluster of nodes to be able to communicate with the range extension network and hence, any of these nodes could assume the role of a CCA. Fortunately, in all these cases, the base algorithm that we propose remains unchanged. Increased layers of complexity can be added to the base algorithm to enable the CCAs to participate in node 'hand-off' as in cellular networks, or to intelligently share the load generated by the nodes in the overlapping regions of intersecting domains. These features are not discussed further in this paper due to space limitations.

Intuitively, one might expect that the performance of the network would be best if we position the CCA near the *weighted geographic centroid* of the domain. This section describes the procedure that computes this weighted centroid. The parameters that the CCA can take into account in formulating the optimization problem could include node positions, each node's offered load, data traffic patterns, priority of the generated traffic, the channel signal to interference noise ratio (SIR), among others. The choice of which parameters to include is determined by the specific network metrics (listed in Section II) that are of importance. For the purposes of this discussion, we consider position and offered load as the primary parameters and the network throughput and the average delay experienced by inter-domain data packets as the basic performance metrics.

Figure 2 represents a typical scenario showing an ad hoc group with a single. As shown in the figure, the domain and the terrain blockages can always be represented by simple rectangular regions, with a granularity chosen according to the desired level of resolution. This representation is useful since the constraints that govern the position of a CCA can then be described by simple linear equations (based on the coordinates of the rectangular boundaries), rather than arbitrary curves.

A fundamental characteristic of MANETs, (also shown in Figure 2) is that, not all the nodes in the domain have a direct link to the CCA. Some nodes will be outside the single-hop radio range of the CCA and will have to route their data packets to the CCA via multiple hops through other ad hoc nodes[2]. However, the cost function that we use in computing the weighted geographic centroid takes the offered load of the individual nodes

[1] Position based schemes have previously been suggested and studied for ad hoc networks, [3], [5].

[2] These data packets are destined for nodes in other domains and thus *must* be routed through the CCA to the range extension network.

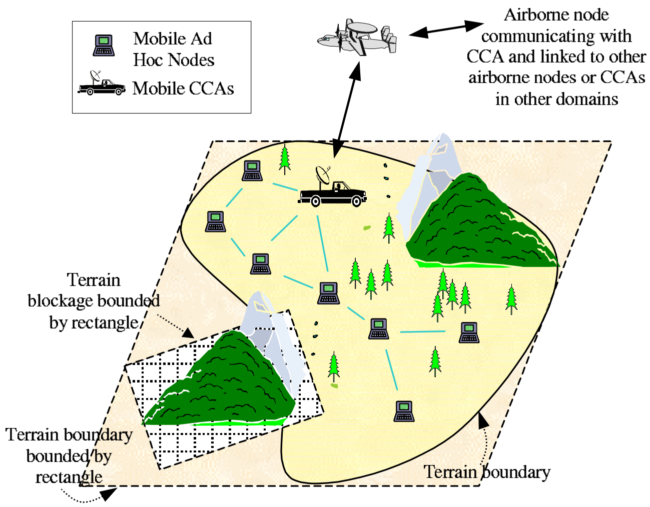

Fig. 2. CCA domain with location bounds.

and the priorities of the packets generated at each node into account. Thus, with successive iterations of the trajectory control algorithm, the CCA eventually will be closest to the most heavily loaded (or highest priority) nodes, which can thus reach the CCA in a single, or minimum number of hops.

A. Optimization Formulation

We assume that the CCA acquires the coordinates of each node in its domain that needs to send inter-domain data packets[3]. For a domain with n nodes, the optimization problem can be formulated as:

$$\text{minimize} \quad \sum_{i=1}^{n} f(\tau_i, \rho_i) \cdot |\mathbf{x}_0 - \mathbf{x}_i| \tag{1}$$

$$\text{subject to} \quad \mathbf{w}_1 \leq \mathbf{x}_0 \leq \mathbf{w}_2 \tag{2}$$

$$\mathbf{x}_0 \geq \mathbf{b}_{2k} \tag{3}$$

$$\mathbf{x}_0 \leq \mathbf{b}_{1k} \tag{4}$$

The optimization variable in the norm minimization expression, Eq. 1, is the desired CCA position, represented by the 2-D position vector, $\mathbf{x_0}$, with reference to any suitable origin in the terrain of interest. The other $\mathbf{x_i}$'s represent the coordinate vectors of the mobile nodes which are obtained at each sampling instant. The weighting factor, $f(\rho_i, \tau_i)$ is a user defined function that depends on the i^{th} node's load, ρ_i, and priority, τ_i. We do not consider the data load due to intra-domain communication among the nodes, although this may affect the available bandwidth for inter-domain communications. Depending on the type of traffic being generated by the nodes, the function $f(\rho_i, \tau_i)$ can be defined appropriately to reflect CBR, or variable bit rate (VBR) traffic and with or without defined priorities. The terms $\mathbf{w_i}$ are the vector coordinates representing the rectangles circumscribing the domain (bottom left and the top right points), and $\mathbf{b_{ik}}$'s are similar vectors that represent the boundary of the k^{th} blockage. Therefore, we are minimizing the sum of the weighted geometric distance from the CCA to each of the nodes, subject to linear boundary and blockage constraints.

As stated earlier, this problem is a non-linear optimization problem [10]. However, since the cost function is the L^2 norm, and the constraints are linear, the problem is a convex program (all norm functions are convex)[4]. Furthermore, the convex optimization problem itself can be easily transformed into a simple linear program (LP) and the cost function can be replaced by an equivalent linear cost function. The resulting LP can then be solved far more efficiently via modern interior point methods [10]. An implementation would require the CCA to perform the **CCA Trajectory Update Algorithm** which is as follows.

- *Input constants (set 'a priori')*: terrain and blockage boundaries, sampling times, optimization metric of interest.
- *Output*: optimum CCA location computed at specific sampling times.
- {*While nodes in the domain have inter-domain data packets to send*}, **DO**:

1. Collect or estimate the position of each node, $\mathbf{x_i}$, at each sampling instant.
2. Collect from each node, an estimate of its current load and the priority that it desires.
3. Perform a local computation to solve the LP equivalent to the optimization problem in Equation 1 and obtain optimum CCA location for that sampling instant.
4. Move towards the optimal location in the most suitable manner, as allowed by the physical constraints[5].
5. Repeat at next sampling instant.

The motion of the CCA can be further governed by certain rules to prevent race conditions and such. As an example, one can have a hysteresis rule that helps to prevent excessive CCA sensitivity, wherein a computed 'new' CCA location has be greater than a minimum of some pre-specified δ units from the present location before we decide to move the CCA.

IV. Computational Complexity

A. MAC Protocol, Routing Support and Overhead

The overhead involved in the execution of the trajectory control algorithm given earlier can be estimated as follows. In steps 1 and 2 of the algorithm, the CCA has to obtain state information from all the nodes. Firstly, the identity of the CCA (e.g. an IP address or MAC address) has to be made known to all the nodes. This information can be software programmed into all the nodes before network deployment, or can acquired on a reactive basis, as required. The overhead in this case is about the same as in discovering a specific node within a MANET. Secondly, to obtain position/loading information, if there are n mobile nodes in the network, then for these nodes to transmit an update every sampling period, the number of messages required is on the order of $O(n^2)$ (assuming flooding is employed to transport these messages) [4]. However, we note that this control information can simply be 'piggy-backed' onto the routing update messages[6], or embedded in the MAC layer 'hello' or even onto the data payload that is routed to the CCA. This makes the overhead required for gathering state information for the algorithm to be essentially the same as, or mariginally in-

[3] We will discuss the technique by which this aquisition is realized and the resulting overhead incurred involved in Section IV.

[4] Since the problem is formulated as a convex program, the solution is found by numerical methods.

[5] The problem of navigating from one location to another, on a 2-dimensional surface with obstacles with no pre-established paths, is a vast area of research in robotics and is not discussed in this paper. Please refer to http://www.jpl.nasa.gov.

[6] Most of these parameters are usually single 8-bit or 16-bit numbers, per node, so payload data length is not an issue.

cremental to, the overhead required for enabling the underlying routing and MAC protocols. For the overhead incurred in deploying various MANET protocols, the reader is referred to [7].

B. Optimization Complexity

The computation of the trajectory itself involves solving a linear program numerically. It is well known that modern interior point LP solvers have a worst-case performance of $O(n^3)$ [10], where *n* is the number of variables in the LP. Thus, for a network of 100 nodes, each of which is assumed to generate inter-domain data packets, we would expect calculations on the order of 100^3 or 1 million iterations per update period. A typical update period is of the order of 0.5 seconds in our simulations. Currently available 'off-the-shelf', inexpensive microprocessors can process on the order of tens of millions of instructions per second [7]. For a network with 1000 nodes, the complexity increases significantly, requiring 1 billion iterations per update period. In such cases, however, it is far more efficient to simply deploy more CCAs, and thus sub-divide the larger domain into smaller domains of ad hoc networks.

V. Simulation Framework and Results

In order to evaluate the performance of our trajectory control algorithm, we have completed extensive simulations. We used the *ns-2* network simulator, release 2.1b6 [12] as our primary simulation platform, with code extensions for the 802.11 MAC and DSDV routing [9], as implemented by Carnegie Mellon University for wireless ad hoc networks. We also conducted simulations with appropriate modifications to the Dynamic Source Routing (DSR) algorithm [11]. Our results were similar and we omit a discussion of this due to space limitations. The LP algorithm itself was integrated into the *ns-2* simulation framework by means of a C function call from the main program code. To implement the optimization algorithm, we used a modified version of PCx [13]. We were interested only in the inter-domain networking performance and consequently, all the data packets that the nodes generated in each domain were always deterministically addressed to the CCA[8], and the range extension network beyond the domain of the CCA was itself considered as a 'black-box'.

The first case that we consider is that of equally loaded nodes generating packets of the same priority. In this case, $f(\rho_i, \tau_i) = 1$, for all i, and the problem is now equivalent to minimizing the sum of the distances from the CCA to the nodes, subject to the boundary and blockage constraints.

The results shown in Figure 3 through Figure 5 are for the following system parameters: a rectangular of size 10,000 units by 10,000 units; mobile ad hoc nodes numbering from 10 to 100 per domain; nodes assigned random velocities chosen uniformly between 0 units/s (stationary) and 25 units/s, moving in accordance with the random waypoint model. The CCA velocity is chosen to be at most one and a half times the maximum speed of the ad hoc nodes. All the nodes are equally loaded and generate traffic 50% of the time. The mobile nodes transmit their coordinates to the CCA once every 0.5 seconds, and the optimization calculations are also repeated with this frequency. The simulations are run for a total of 5000 seconds.

In Figure 3, simulation data was collected for three different scenarios: a CCA that is placed statically at the center of its domain; a CCA that is moving according to a random waypoint model; and finally, a CCA, the locus of whose trajectory is being updated using our optimization method.

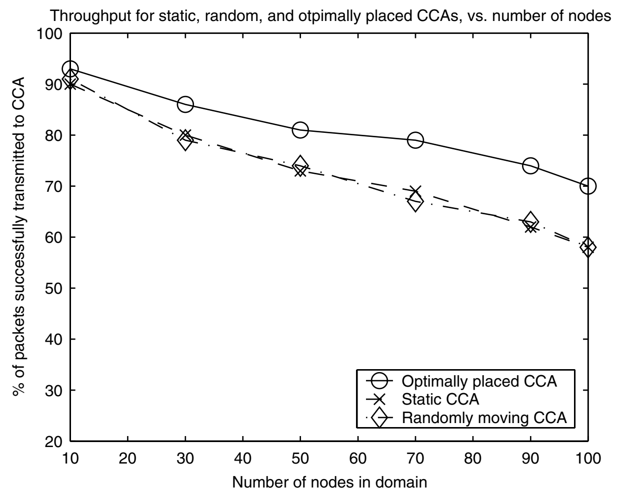

Fig. 3. Comparison of network throughput with optimally placed CCA versus statically placed or randomly moving CCA.

Note that when our algorithm is implemented, the improvement in throughput is as high as 10% per domain (ignoring interdomain interference effects), and hence can be very significant for the network as whole when there are multiple CCA domains.

The advantage of the dynamic CCA placement technique is much more evident when we increase the area covered by the blockages (Figure 4).

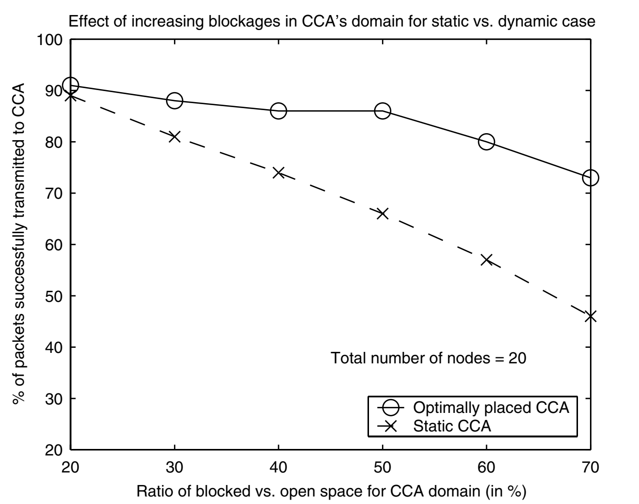

Fig. 4. Effect of blockages on the performance of the static vs. optimally placed CCAs.

In Figure 4, initially, with very few blockages in the domain, the number of packets that are successfully transmitted to the

[7] http://www.intel.com

[8] Intra-domain traffic in the MANET does not affect the CCA trajectory.

CCA is roughly the same for the two cases, i.e., the case in which the CCA is placed statically at the center of the domain, and the case in which the CCA is optimally positioned. However, as the numbers of blockages are increased causing the area covered by the blockages to increase, the open space in the domain decreases as a result, and the throughput drops dramatically if the CCA is statically placed. When the CCA is optimally placed, we noticed the improvement in throughput to be as high as 60%.

To test the performance of the network when the cost function is altered to incorporate different priorities for different nodes[9], 100 nodes were deployed and the load generated by each node was progressively increased from 10% to 90%. For a specific value of the offered load, half the nodes (chosen randomly) generated high priority traffic ($f(\rho_i, \tau_i) = \tau_i = 10$), whereas the remaining nodes generated low priority data traffic ($f(\rho_i, \tau_i) = \tau_i = 1$). For each of these cases, the average message latency was measured while using the CCA trajectory update algorithm with, and without, the priority class as a parameter in the cost function. The results are depicted in Figure 5. The CCA now favors the nodes generating the higher priority

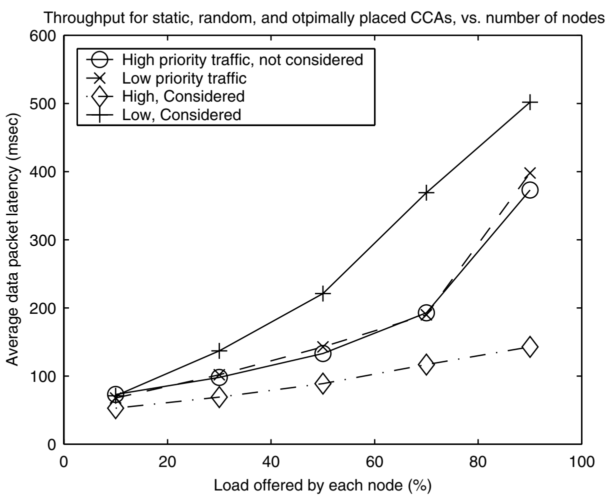

Fig. 5. Effect of including the effect of priority in the optimization cost function.

traffic. These packets are delivered more efficiently–directly instead of via multiple hops–and with improved latency. Since the system capacity is fixed, the price that is paid, however, is that the lower priority traffic suffers increased latency and reduced throughput as a consequence. We also note that, as expected, when the cost function in the optimization formulation does not take packet priorities into account, but considers only the position and offered load, there is no significant difference in the latency incurred by the different classes of traffic.

VI. Conclusions and Future Work

In order to support scalability in ad hoc networks, one can envision the deployment of a range extension network that consists of airborne nodes interfaced with the ad hoc network. One approach could be to deploy gateways that we refer to as CCAs to relay data traffic from/to an ad hoc group to/from the range extension network. The objective of this work is to determine where the CCA is to be placed relative to the ad hoc group of nodes such that certain network performance metrics are optimized. This objective can be formulated as a set of convex optimization problems. By means of suitable modifications, we simplify these convex formulations such that they can be very efficiently solved by numerical methods. We enforce the CCA to follow the computed optimal trajectory, and evaluate the improvements in network performance that is achieved.

Simulation results indicate that the network throughput improves by about 10-15% per CCA domain if the CCA moves in accordance with the optimally computed trajectory, as opposed to being static or moving randomly. A similar improvement is seen in terms of a reduction in latency that the data packets experience. The cost function in the optimization formulation can also be appropriately modified to support better performance for prioritized traffic. We also show that the operations that are required in order to thus define an optimal trajectory for the mobile CCAs can be performed efficiently with current hardware technologies, and with little additional overhead.

One particular extension of interest is to consider the case wherein multiple CCAs are present in a particular domain, where the nodes have the ability to choose different CCAs for relaying their inter-domain traffic. This structure can be exploited to balance the loads in different domains, thereby improving performance further. This is currently being investigated.

[9]Note that a node's priority depends on the priority of the packets that it generates at the given time. This is dynamic, akin to the offered load, and changes with time.

VII. Acknowledgements

The authors wish to thank Gregory Pottie (UCLA), and Yongguang Zhang (HRL), for valuable discussions and input.

References

[1] Z. Haas, et. al., "Guest Editorial on Wireless Ad Hoc Networks", *IEEE Journal on Selected Areas in Communications*, Vol. **17** No. 8, August 1999.

[2] B. M. Leiner, "Goals and challenges of the DARPA GloMo program (global mobile information systems)," *IEEE Personal Communications Magazine*, Vol. **3**, No. 6, pp. 34-43, Dec. 1996.

[3] Y. Ko, N. Vaidya, "Using Location Information in Wireless Ad Hoc Networks", *Proceedings of the 49th IEEE Vehicular Technology Conference*, Vol. **3**, pp. 1952-1956, 1999.

[4] G. Karumanchi, et. al., "Information Dissemination in Partitionable Mobile Ad Hoc Networks", *Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems*, pp. 4-13, 1999.

[5] K. Amouris, et. al., "A Position-Based Multi-Zone Routing Protocol for Wide Area Mobile Ad Hoc Networks", *Proceedings of the 49th IEEE Vehicular Technology Conference*, Vol. **2**, pp. 1365-1369, 1999.

[6] IEEE Standards Committee, "Medium Access Control (MAC) and Physical (PHY) Specifications," IEEE P802.11/D10, Jan. 1999.

[7] E. Royer, C.-K. Toh, "A Review of Current Routing Protocols for Ad Hoc Wireless Networks", *IEEE Personal Communications Magazine*, pp. 46-55, April 1999.

[8] D. Johnson and D. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth, chapter 5, pp. 153-181, Kluwer Academic Publishers, 1996.

[9] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", *Computer Communication Review*, pp. 234-44, Oct. 1994.

[10] D. Bersimas, J. Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific, Massachusetts, 1997.

[11] J. Raju, J. J. Garcia-Luna-Aceves, "A comparison of on-demand and table driven routing for ad-hoc wireless networks," *IEEE International Conference on Communications (ICC2000)*, Vol. **3**, pp. 1702-1706, 2000.

[12] **ns-2** Network Simulator, *USC ISI*, http://www.isi.edu/nsnam/ns.

[13] http://www.cs.anl.gov