Optimizations for Charged Service Provision in Mobile Ad Hoc Networks

Christopher N. Ververidis and George C. Polyzos
Mobile Multimedia Laboratory
Computer Science Department
Athens University of Economics and Business
Athens, Greece
chris,polyzos@aueb.gr

Abstract

Optimized service provisioning is a challenging problem in dynamic environments such as Mobile Ad Hoc Networks (MANETs). Most of the existing approaches assume an environment, where service provision is free (and dictated) and servers do not have an incentive to maximize their benefit. In this paper we consider the nodes in MANETs to be independent, rational agents trying to maximize their profits through service provision. We model this problem as a Generalized Assignment Problem (GAP). We adopt a pay-as-you-go model and introduce into the proposed profit maximization algorithm expected payoffs based on estimates of server-to-client connectivity. We define connectivity as the lifetime of the network connection between a client and a server. We experimentally study cases with non-cooperative and cooperative servers and investigate the gain of the estimate based maximization algorithm versus a classic maximization algorithm, which does not take into account the network's dynamics that affect server-to-client connectivity. The results show that our approach achieves up to three-fold improved server profits compared to the classical one and is especially suited for MANETs with high-mobility.

1. Introduction

Mobile Ad Hoc Networks (MANETs) were introduced as a networking alternative for cases where traditional infrastructure-based networks were unavailable either due to prohibiting costs (e.g. in remote and hard to reach locations) or prohibiting situations (e.g. after a devastating disaster like an earthquake). The great potential of MANETs has attracted the interest of many researchers and the evolution of MANET technology has been remarkable. Initially the main application scenarios for MANETs, included emergency response team operations and military missions. In this context a basic assumption made was that all nodes members of the network were willing to cooperate. Also, the researchers started differentiating the nodes participating in a MANET between powerful and weak nodes. Maturing further, MANET technology began to be considered also for cooperative working environments, for group communications (e.g. conferences), for peer-to-peer applications (e.g. gaming, file sharing etc.), for sensor networks and for vehicle-to-vehicle communications [6]. At the same time the assumption of unconditional cooperation began to seem not so realistic and issues arising from malicious, misbehaving and selfish (non-cooperative) node behavior came to the surface. Also, the client-server model started to take the place of the powerful-weak node model. This evolution path has led to investigations into MANETs, where mobile servers being nodes in the MANET offer their services to other MANET nodes, acting as clients, for profit. The main factors that differentiate service provisioning over MANETs from service provisioning over fixed networks are:

- Service provision in MANETs is opportunistic
  - There are no fixed, well-known service providers.
  - Any node (individual) can be a service provider for her own benefit and for as long as she participates in the MANET, or for as long as she desires to be a service provider.
- For-profit (charged) service provisioning in MANETs is in its infancy and at this stage it is better for clients to make payment offers and let servers decide whether they want to accept them or not. (Pull-based model)
- MANET communications are significantly more unreliable and error-prone than fixed network communications. Hence, service provision in MANETs (especially if services are charged and a server's goal is profit maximization) must account for this.
- Server capacity is much more constrained in MANETs (because servers are typically smaller in order to be light and mobile, on battery power etc.).
  - Communication costs are significant due to devices' energy constraints.

978-1-4244-2100-8/08/$25.00 ©2008 IEEE

- Client selection is a major issue especially if servers want to maximize their profit.

- Solutions for server profit optimization must be computationally efficient due to the processing constraints of mobile devices.

In this paper we target such commercial-purpose MANETs and investigate how mobile servers can maximize their profits by accounting for the volatility of a MANET environment. In the next Section of the paper we present related work. In Section 3 we describe in detail the system under consideration and in Section 4 we briefly describe the Generalized Assignment Problem (GAP) and formulate the problem of server profit maximization as such. In Section 5 we provide analysis and modeling of the problem using an enhanced GAP model. Finally in Section 6 we present simulation results and Section 7 provides our conclusions and plans for future work.

2. Related Work

As already stated in the previous section, in MANETs service breaks can be much more frequent due to increased link and path failures. In [12] and [15] it is shown that for medium and high node speeds the path (availability) duration decreases exponentially and hence only short-lived communications requiring at most a few tens of seconds will be completed before a path break occurs. In [17,18] we introduce a metric called Service Availability Duration (SAD), defined as the length of time that elapses from the moment the service is discovered until that time when access to the service is lost (as a result of mobility, congestion, or interference). It should be noted that if the path to the original service provider is lost, but there exists another provider for the same service-type in the node's routing table, then the service is still considered 'alive'. Only when all the routes from a node to all the available providers of the service are lost, this particular service is considered not to be available anymore to that node. Based on measurements using various speeds and densities and (even) assuming a high server to client ratio (i.e., 1/3), a service does not typically have a SAD larger than a couple of hundreds of seconds. If the server to client ratio is less (which is more realistic), then the SAD will drop to a few tens of seconds. Also if the service state cannot be transferred when switching servers, then the SAD actually drops to the path duration mentioned earlier. Moreover, in the most severe cases, path breaks may also lead to network partitions that separate clients from their prospective servers.

In the literature, there have been quite a few approaches trying to deal with those severe cases. Their main aim is to identify when a partition is going to happen so that the requested service is replicated in advance and connectivity to it can be guaranteed. In [13] each client monitors the set of disjoint paths between itself and the server and computes a metric. If this metric falls bellow a certain threshold then a potential partition is identified and server replication is initiated. This method assumes that services are such that client nodes may also bear to host them and also no considerations on server profits are taken. In [16] the authors propose a distributed localized algorithm for detecting critical nodes. Critical nodes are those nodes that if they get disconnected, a network partition will occur. Using this algorithm, servers can predict the partition and replicate the service to another node, which is part of the right future partition. In [10] a service backbone formation mechanism is proposed to handle network partitions. This backbone is such that every node can contact at least one of its members in at most r hops. Every node monitors the number of nodes that are in its vicinity (r-hops away) and do not have access to a server. The node with the highest number of such neighbors must get a service replica. Once again, servers and clients are considered equally powerful and service provision is free. In [19] a partition prediction model is proposed based on grouping nodes according to their position and speed. Every client sends its coordinates and velocity to the server. The server groups nodes based on a pattern-matching algorithm. Having this global knowledge, the server can predict future partitions and be replicated accordingly. The same assumptions of replication capability for any node are also made here. Replication is also used in [5], where an algorithm based on the partition detection mechanism of the TORA routing protocol is used along with an optimized replica deployment scheme. Similar schemes are also proposed in [20], taking also into account link failure probabilities during data replication and trying to balance data accessibility and query delay.

Closely following the fundamental mechanism for data replication in MANETs initially proposed by T.Hara [7], all the aforementioned approaches assume that service replication can always be carried out and do not consider servers as business entities seeking to maximizing their profits. They mainly focus on service continuation despite network partitioning. The strong assumption that a service can always be replicated cannot hold in most MANETs, either due to client device constraints, or to the nature of the services. Imagine for example that a server node is providing live stock price feeds obtained over its 3G connection. Such a service cannot be replicated to any node since a 3G connection is necessary.

In our approach, we consider non-replicable services and focus on the economic potential from providing services in MANETs. A similar point of view was adopted in [21]. However, the authors in [21] do not account for the volatility of a MANET, assuming that services will be provided in full and hence payments will be returned in full to service providers. We show that this assumption leads to suboptimal client set selection. In our approach, we take into account path failure probabilities in the proposed algorithms for enabling servers to select the optimal client set that will maximize their expected profits. Also in [21] the authors do not address cases where servers are non-cooperative and as a result more than one server selects to serve the same client (which would result in loss of profit for all but the server finally chosen by the client). We study such cases and additionally we experimentally show that in MANETs the total value for service provisioning can only be obtained if servers cooperate.

3. System Description

As already stated in the introduction, we study commercial-purpose Mobile Ad Hoc networks. Those networks are comprised of mobile clients and mobile servers acting selfishly in that they try to maximize their own profit. We assume that packet forwarding is given the right incentives by means of virtual money, or credit payments from sender nodes to forwarding nodes, as it is proposed in [22],[2],[8] and [14]. Also, virtual money (possibly exchangeable to real money) is used by clients for paying servers for the provision of a service. As it is done in [22], we assume the existence of management points acting as central-banks and controlling the flow of payments among clients, servers and forwarders. In the experiments and for the obtained results, we assume that all intermediate nodes on the path from a client to a server will not deny forwarding[1].

We also assume that servers participating in the network have a finite capacity. In our context capacity refers either to server memory or processing capacity constraints or both. Periodically, servers announce their presence and wait for client requests. After their announcement, servers begin collecting client requests (including the client's bid for the requested service) for a given period. Upon the end of this period, the servers construct a schedule for serving those clients that maximize their profit while not exceeding their capacity. We should note here that there are the following two cases:

- If there are more than one servers belonging to the same owner, then it is natural to assume that there is some form of communication (e.g. using a side channel) so that the servers can collaboratively decide on the best client allocation among them (common knowledge required).

- If there are more than one servers, but they belong to different providers, then each one would try to maximize its own profits. Here, two servers may select the same client in their respective optimal client sets. In this case, the client chooses the server that it estimates to be the most stable (reachable longer).

Once they have finished providing the requested services to the current client set, the servers enter their next announcement period. A last assumption is that clients will only pay for the amount of service they have received. This means that if a service provision is terminated earlier than its expected normal termination, then the client will have paid only for the amount of the service received until the termination happened.

For example consider the scenario, depicted in Fig.1, of a Vehicular Ad Hoc Network (VANET) where a car is capable of offering a navigation service to other cars in the network. A car using this service will be paying for the service as long as it can reach the serving vehicle via the multihop network (e.g. cents/packet).

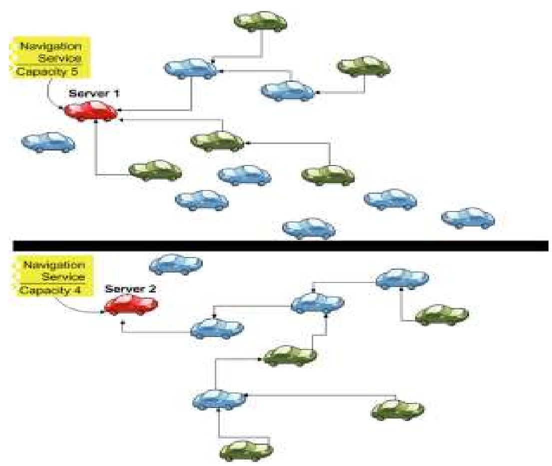

Figure 1. VANET with mobile servers.

4. Problem Formulation and Analysis

Based on the description given, the problem of server profit maximization can be modeled as a Generalized Assignment Problem (GAP). The GAP is defined as: There are n items x_1 through x_n and m bins. Each item has a weight a_{ij} (weight of item j if assigned to bin i) and a value c_{ij} (value of item j if assigned to bin i) and every bin has a capacity b_i. The problem is to find the optimal assignment of items into the bins such that the capacity constraints of the bins are not violated and the total value obtained is maximized. The mathematical formulation of the GAP is the following:

$$maximize \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} * c_{ij} \tag{1}$$

subject to:

$$\sum_{j=1}^{n} x_{ij} * a_{ij} \leq b_i, i = 1, ..., m \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} \leq 1, j = 1, ..., n \tag{3}$$

$$x_{ij} \in \{0, 1\}, i = 1, ..., m, j = 1, ..., n \tag{4}$$

This model is directly applicable to our problem if we assume that items are clients, that a_{ij} is the amount of resources consumed at server j if it selects to serve client i

[1] To some extend we could address some forms of stochastic denials of forwarding in our modeling of a probability of path breaks.

(also called requested capacity), that c_{ij} is client i's payment to server j and that bins are servers with serving capacities b_i.

Solving the GAP is NP hard, and it is even APX-hard to approximate it [3]. However, there exist polynomial time approximation algorithms (having approximation guarantee equal to $(1 - 1/e - \varepsilon)$, for any $\varepsilon > 0$), further analysis of which is out of the scope of this paper (the interested reader is reffered to [11] and [4]). Especially since the size of commercial MANETs cannot grow more than a few tens of nodes and the instances of GAP are small, the approximation algorithms (and even greedy algorithms) perform satisfactorily.

Solving the classic GAP problem without considering profit estimation would lead to a solution vector x_{ij}, where:

$$x_{ij} = \begin{cases} 1, & \text{if client } j \text{ has been assigned to server } i \\ 0, & \text{if client } j \text{ has not been assigned to server } i \end{cases}$$

It is true that given the c_{ij} and solving the GAP, the x_{ij} vector is selected so that we obtain the maximum $\sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} * c_{ij}$.

However, if we consider that service breaks may occur and that clients pay providers only for the part of the service received until the break happened, then the produced x_{ij} vector may not lead to the actual profit maximizing solution. The proof is given in the following:

Assume:

x_{ij} is the computed "optimal" solution vector,

$\sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} * c_{ij}$ is the "optimal" profit obtained,

p_{ij} is the portion of the service actually received from client j when being served by server i.

If there is one client l, allocated to server k, in the optimal allocation, and another allocation of one client n to server k, not included in the optimal allocation, for which:

$$a_{kn} = a_{kl}, c_{kn} \leq c_{kl} \text{ and } p_{kn} \geq p_{kl} \tag{5}$$

such that:

$$c_{kn} * p_{kn} \geq c_{kl} * p_{kl} \tag{6}$$

then using a pay-as-you-go model and taking into account that link failures may happen we get:

$$\sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} * c_{ij} * p_{ij} \leq \sum_{i=1}^{m} \sum_{j=1}^{n} x^*_{ij} * c_{ij} * p_{ij} \tag{7}$$

where:

$$x_{kl} = 1, x_{kn} = 0 \text{ and } x^*_{kl} = 0, x^*_{kn} = 1 \tag{8}$$

Hence the computed solution using the classic GAP model is not always the optimal one considering that service failures may occur. For getting close to optimality in an error prone environment as a MANET, we enhance the GAP model with estimates for the profits (instead of fixed profits) of service provisioning. The model can then be reformulated as follows:

$$maximize \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} * c_{ij} * p_{ij} \tag{9}$$

subject to:

$$\sum_{j=1}^{n} x_{ij} * a_{ij} \leq b_i, i = 1, ..., m \tag{10}$$

$$\sum_{i=1}^{m} x_{ij} \leq 1, j = 1, ..., n \tag{11}$$

$$x_{ij} \in \{0, 1\}, i = 1, ..., m, j = 1, ..., n \tag{12}$$

Since we do not consider the case where a server can over-schedule we do not replace equation (10) with:

$$\sum_{j=1}^{n} x_{ij} * a_{ij} * p_{ij} \leq b_i, i = 1, ..., m\ , \tag{13}$$

since if a server under-estimates p_{ij} it may admit more clients than it can serve. The parameter p_{ij} is an estimate of the proportion of service that can be delivered to client j by server i. This parameter is directly related to the path failure probability from the client to the server due to channel conditions or mobility. It is worth noting here that any algorithm for solving the GAP problem can be directly applied to this model if instead of c_{ij}'s, the $c_{ij} * p_{ij}$ values are taken into account. Also, the p_{ij} can be based on sophisticated estimates like those proposed in [9].

It is obvious that solving the GAP requires server cooperation and can be applied into MANETs where service providing nodes belong to the same owner. Actually every server must inform her team-member servers about her connectivity estimations (p_{ij}s) to all clients, the client's bids (c_{ij}s) and their requested capacities (a_{ij}s) along with the server's own maximum capacity. All this information can be encoded into small vectors and be efficiently distributed over a side-channel (e.g. 3G connection) among servers.

In case that competitive teams of service providers (i.e teams of service providers belonging to different owners) exist in the MANET, then each team can solve a GAP and the solutions among teams may have overlaps. An overlap means that a given client is chosen by more than one server teams. In this case the client will select to get the service from the team whose server is estimated to have the best connectivity to it. The connectivity is measured as the proportion of the service that can be delivered while the client is in contact with the server. In this case there will be a loss of profit for all the other teams that have also chosen that particular client.

5. Approximation of Connectivity

As described in section 3 servers announce their presence in the network at regular intervals and wait for clients' bids for the next serving period. Then, based on those bids, the requested capacities and the estimation of the proportions of service that each client will receive, the servers make their selection about which clients to serve. In this section we will propose an algorithm for estimating the connectivity (i.e. the expected lifetime of network connection) between a client and a server.

It is widely known that the duration of a path between two nodes in a MANET is dependent on network density, on node speed, on the number of hops separating the two nodes and on the transmission range. Returning to our case, a server must be able to obtain an estimate of the path duration with any client that has requested to be served for the next serving period. It can compute this estimate as follows:
$p_{ij} = \frac{\text{Expected Lifetime of Connectivity between client } j \text{ and server } i}{\text{Serving period duration}}$
Regarding the impact of the number of hops between client and server on the expected path duration, it is intuitive that the fewer the hops, the better the connectivity between the two nodes. In the following we show how density impacts the path duration for paths of up to 4 hops. Our simulation-based measurements show that average path duration follows certain patterns and can be accurately estimated. The results were obtained assuming serving periods of 100 seconds. For node mobility we use the Random WayPoint (RWP) mobility model with constant speed (maximum speed takes the values 3.5m/sec, 7m/sec and 14m/sec) and no pause time. We used the Qualnet simulator [1] for obtaining the path duration measurements. Under these setting we have tested several scenarios (see Table I) by altering the number of participating nodes and the terrain sizes (node range was fixed to 400 meters). We compute node density using the following formula:

$$D = \frac{N*\pi*R^2}{TerrainSize},$$

where N is the number of clients and R is the transmission range.

Terrain Size	Number of Nodes	Density
2000x2000	11	1,3816
2000x2000	17	2,1352
1500x1500	11	2,4561
2000x2000	22	2,7263
1250x1250	11	3,5368
1500x1500	17	3,7959
1500x1500	22	4,9123
1250x1250	17	5,4661
1000x1000	11	5,5264
1250x1250	22	7,0737
1000x1000	17	8,5408

Table 1. Density Values.

In Fig. 2 we present the results. Each point in the diagram represents an average path duration obtained from experiments with 90 seeds and having duration of 4000 seconds each (this corresponds to 3600 serving periods). As it is obvious from Fig. 2 for a certain speed and serving period the average path duration of 1, 2, 3 and 4-hop paths can be well approximated as a logarithmic function of density that increases as density increases (either due to node number change, terrain size changes or node range changes).

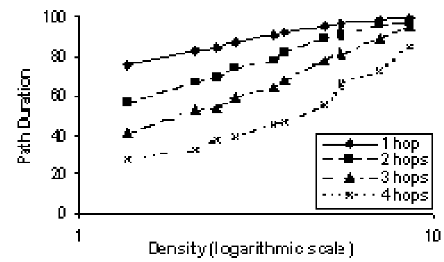

Figure 2. Path Duration vs Density and Number of Hops.

In Fig. 3 we present how those logarithmic trends change a s speed increases. Fig. 3 presents the results for 1 hop paths but the impact on the pattern is similar for 2, 3 and 4-hop paths also (omitted due to space limitations). If we model how speed impacts the coefficients of the logarithmic functions used to approximate the path durations we actually see that they can be well approximated by another set of logarithmic functions.

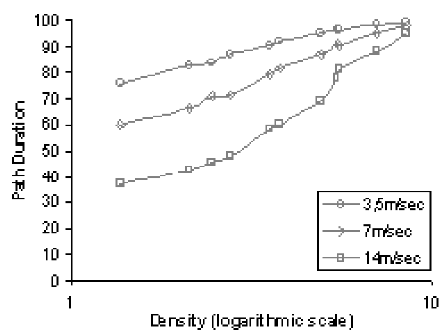

Figure 3. Path Duration vs Density and Speed.

Hence we can derive analytical formulas, which take into account node speed and density and estimate 1-hop, 2-hop, 3-hop and 4-hop average path durations. The formulas are of the following form: $F_x = (a_x * ln(speed) + b_x) * ln(Density) + (c_x * ln(Speed) + d_x)$, where x is 1 to 4 and corresponds to number of hops.

Hence, if we assume that nodes also include their position coordinates and speed along with the bids they sent to servers, servers can actually compute node density and average speed. Servers also know the number of hops required to reach every client and hence by using the approximation formulas derived from the experimental results, they could estimate the p_{ij} values.

6. Performance of Client Selection Algorithms

For computing the solution to the GAP when servers are cooperative we have used a branch and bound algorithm based on linear programming that always finds the optimal solution. We compare the results of the algorithm when it is run:

C-GAP : Without considering payment estimates for solving the classic GAP (C-GAP algorithm),

OE-GAP: Considering payment estimates and accurate knowledge of client-server connectivity (E-GAP Oracle algorithm),

AE-GAP: Considering payment estimates with approximations obtained for client-server connectivity based on density and speed (E-GAP Approximation algorithm).

For the non-cooperative server case we consider single server non-cooperative teams. In this case each server solves a specific version of GAP where there is only one server (no info on other servers). This is actually a Single Knapsack (SK) problem, which we also solve with a branch and bound algorithm that always finds the optimal solution. Once again we compare the results of the algorithm when it is run:

C-SK : Without considering payment estimates for solving the classic SK (C-SK algorithm),

OE-SK: Considering payment estimates and accurate knowledge of the path failure probability (E-SK Oracle algorithm),

AE-SK: Considering payment estimates with approximations obtained for client-server connectivity based on density and speed (E-SK Approximation algorithm).

Using Qualnet [1], we have simulated a MANET consisting of 2 servers and 20 clients. The node range has been set to 400 meters. Server cooperation and non-cooperation scenarios have been investigated. In order to take into account the effects of node speed on our algorithms, all nodes move following the random waypoint mobility model with constant speed (experiments for different speeds have been conducted). We also investigated the effects of density by using various terrain sizes ranging from 1250x1250 square meters to 2000x2000 square meters.

For the first set of experiments and for simplifying the analysis we assume that a_{ij}=a_{ij}=1 unit (requested capacity and payment unit respectively) for all i's and j's[2] , and that the 2 servers have equal capacities (either 5 or 25 units). When servers have 5 units of capacity this means that each server can provide only part of the requested services (since there are 20 clients), while when servers have 25 units of capacity this means that each server may serve all requests on his own.

[2] This way we actually can talk about client allocations to servers, which is more comprehensible. Experiments with a_{ij} following a uniform distribution led to the same conclusions and are omitted due to space limitations and for the shake of better readability.

6.1. Cooperative Servers

Fig.4 presents simulation results considering cooperative servers having capacity of 5 units (hence each server may serve up to 5 clients in every serving period). Nodes move according to the Random Waypoint model with constant speed of 3.5m/sec. The results are average values obtained over 20 experiments with different seeds, each having duration of 4000 seconds (40 serving periods per experiment). The y-axis presents the percentage of total profit gain/loss using the E-GAP solving algorithm (taking into account client-server connectivity) instead of the classic GAP solving algorithm. The x-axis presents different densities.

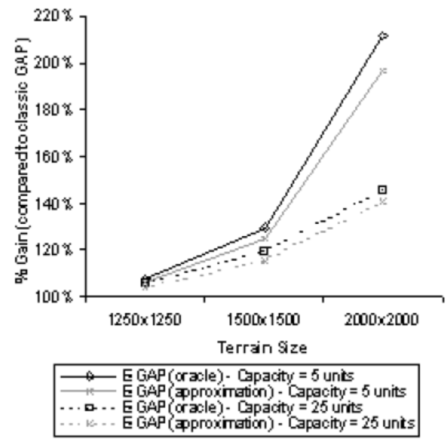

Figure 4. Capacity and Terrain Size vs Profit.

In this simple case, since all clients have the same requests and pay the same amount of money to any server, if servers solve the classic GAP problem then the client to server assignment is actually done randomly by the C-GAP algorithm. However, since path failures occur, such a random assignment cannot lead to profit maximization for the 2 servers due to the fact that clients may not be assigned to the server they are best connected (i.e. the server for which the client-server path has the maximum lifetime among all paths to other servers). Using the E-GAP algorithms (oracle or approximation), path failures are taken into account and the allocation of clients to servers is done optimally (i.e. every client is allocated to the server it is best connected to) hence leading to a maximization of the total profits obtained. The sparser the network and the higher the mobility the greater is the need for carefully choosing the clients to be served, since path failures are more prevalent and the random assignment will most likely lead to frequent path failures and to decreased profits (assuming the pay-as-you-go model). This fact is validated by the results presented in Fig. 4, where it is evident that the profit gain increases with decreasing density. Also, the profit gains are larger when the servers cannot satisfy all the demand, since then it is even more crucial to select to serve only the best connected subset of clients. Finally, it is shown that the approximation algorithm closely follows the oracle algorithm, which has perfect knowledge of the connectivity between any client-server pair, hence leading also to much better performance than that of the C-GAP algorithm.

In Fig. 5 we show the experimental results regarding the investigation of the impact of mobility on the profit gains obtained using the E-GAP algorithms for cooperative servers. The general simulation setup is the same but now nodes are placed on terrain sizes of different density. It is evident that speed impacts the profit gains especially for higher density scenarios. In lower density scenarios, speed does not play a significant role since most connections are 1-hop connections and the speed affects only two participants, the client and the server. In denser networks, where paths may span from 1 to 4 hops between servers and clients, the speed affects all nodes participating in the path and hence its impact is more evident.

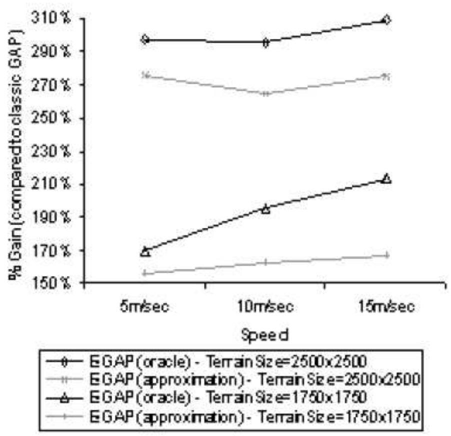

Figure 5. Speed vs Profit.

6.2. Non-Cooperative Servers

The results of the simulation when the servers are non-cooperative are actually the same as the ones observed in figure 3 and 4, the only difference being that in cases where servers have capacities of 25 units (each server can satisfy the whole demand on its own) the profit gains for using E-GAP instead of GAP are zero. This is explained by the fact that either GAP or E-GAP will result in the solution that each server selects to serve every client in order to maximize its profit. The same conflicts will exist and hence the total profits will be the same irrespectively of the algorithm.

What is more interesting is that the total profit obtained in the non-cooperative case is on the average lower. This is due to the fact that when servers are non-cooperative the client-to-server allocations may include conflicting sets of clients, since the servers do not try to optimally "share" clients. A conflict means that a client has been selected by more than one server. Since this client will be finally served by one of the servers, this results to loss of profit for the other server that had selected that same client in its allocation.

In Fig. 6 we show that being non-cooperative is bad for the total profit obtained especially for high density scenarios. In such scenarios there are more chances for servers to select the same clients and hence have conflicts, which result in loss for the total profits. However, in sparser networks servers will most likely select non-conflicting client sets consisting of nodes in their respective vicinities. Hence, the total profits obtained will not differ compared to the case of having cooperative servers.

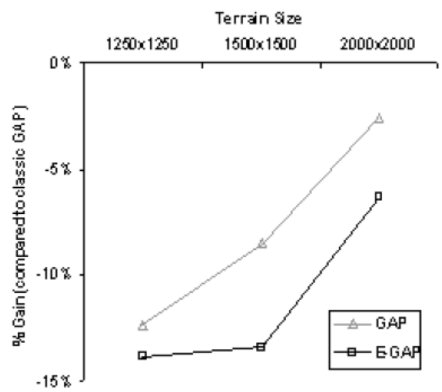

Figure 6. Server Capacity = 5 units, Speed = 3.5m/sec.

7. Conclusions and Future Work

In this paper we have extensively studied the problem of optimized service provisioning Mobile Ad Hoc Networks (MANETs). We argued that when service provision is not free and capacity constrained service providers seek to maximize their profits by providing services to other nodes in the MANET, it is crucial that optimization mechanisms exist for selecting the best client set to be served. We have modeled this problem as a Generalized Assignment Problem (GAP) and showed the benefits from taking into account server-to-client distances and estimates for the path failure probabilities in the algorithms for solving it.

We have studied cases with non-cooperative and cooperative servers. Our simulations showed that the proposed estimate based algorithms are valuable, especially for sparse mobile ad hoc networks. Their performance compared to a classic GAP algorithm was better even by a factor 3 in terms of accumulated profits for the servers. Further experiments revealed also that for a specific ratio of server capacity to the maximum possible requested capacity, competitive servers can acquire greater profits if they cooperatively select their respective client sets.

In our future work we plan to investigate the relation between density, speed and client-to-server connectivity under serving periods of longer or shorter duration.

References

[1] Qualnet simulator, http://www.scalable-networks.com.

[2] L. Buttyán and J.-P. Hubaux. Enforcing service availability in mobile ad-hoc WANs. In *MobiHoc*, pages 87–96. ACM, 2000.

[3] C. Chekuri and S. Khanna. A PTAS for the multiple knapsack problem. In *SODA*, pages 213–222, Philadelphia, PA, USA, 2000. Society for Industrial and Applied Mathematics.

[4] R. Cohen, L. Katzir, and D. Raz. An efficient approximation for the generalized assignment problem. *Inf. Process. Lett.*, 100(4):162–166, 2006.

[5] A. Derhab, N. Badache, and A. Bouabdallah. A partition prediction algorithm for service replication in mobile ad hoc networks. In *Proceeding of the 2nd IFIP Wireless On-Demand Network Systems Working Conference, WONS 2005*, pages 236–245, 2005.

[6] M. Gerla. From battlefields to urban grids: new research challenges in ad hoc wireless networks. *Pervasive Mob. Comput.*, 1(1):77–93, 2005.

[7] T. Hara. Effective replica allocation in ad hoc networks for improving data accessibility. In *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM-01)*, pages 1568–1576, Los Alamitos, CA, Apr. 22–26 2001.

[8] Z. Ji, W. Yu, and K. J. R. Liu. An optimal dynamic pricing framework for autonomous mobile ad hoc networks. In *Proceedings of the 25th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM-06)*, 2006.

[9] S. Jiang. An enhanced prediction-based link availability estimation for MANETs. *IEEE Transactions on Communications*, 52(2):183–186, 2004.

[10] B. Liang and Z. Haas. Virtual backbone generation and maintenance in ad hoc network mobility management. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM-00)*, volume 3, pages 1293–1302, 26-30 Mar 2000.

[11] V. S. M. Lisa Fleischer, Michel X. Goemans and M. Sviridenko. Tight approximation algorithms for maximum general assignment problems. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithm (SODA '06)*, pages 611–620, New York, NY, USA, 2006. ACM.

[12] A. McDonald and T. Znati. A path availability model for wireless ad-hoc networks. In *In Proceeding of the IEEE Wireless Communications and Networking Conference, (WCNC-1999)*, volume 1, pages 35–40, 1999.

[13] D. S. Michael Hauspie and J. Carle. Partition detection in mobile ad hoc networks. In *Proceedings of the Second Mediterranean Workshop on Ad-Hoc Networks (MED-HOC-NET-03)*, Mahdia, Tunisia, June 2003.

[14] M. J. Neely. Optimal pricing in a free market wireless network. In *Proceedings of the 26th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM-07)*, pages 213–221, 2007.

[15] N. Sadagopan, F. Bai, B. Krishnamachari, and A. Helmy. PATHS: analysis of PATH duration statistics and their impact on reactive MANET routing protocols. In *Proceedings of MobiHoc 2003*, pages 245–256. ACM, 2003.

[16] M. Sheng, J. Li, and Y. Shi. Critical nodes detection in mobile ad hoc network. In *Proceedings of the 20th IEEE International Conference on Advanced Information Networking and Applications (AINA-06)*, pages 336–340, 2006.

[17] C. N. Ververidis and G. C. Polyzos. Impact of node mobility and network density on service availability in manets. In *Proceedings of the 14th IST Mobile and Wireless Communications Summit*, 2005.

[18] C. N. Ververidis and G. C. Polyzos. A routing layer based approach for energy efficient service discovery in mobile ad hoc networks. *Wireless Communications and Mobile Computing, WCMC*, to appear.

[19] K. Wang and B. Li. Efficient and guaranteed service coverage in partitionable mobile ad-hoc networks. In *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM-02)*, volume 2, pages 1089–1098, 2002.

[20] L. Yin and G. Cao. Balancing the tradeoffs between data accessibility and query delay in ad hoc networks. In *Proceedings of the 23rd International Symposium on Reliable Distributed Systems (23rd SRDS'04)*, pages 289–298, Florianopolis, Brazil, Oct. 2004. IEEE Computer Society.

[21] J. Zhai, Q. Li, and X. Li. Data caching in selfish MANETs. In X. Lu and W. Zhao, editors, *ICCNMC*, volume 3619 of *Lecture Notes in Computer Science*, pages 208–217. Springer, 2005.

[22] S. Zhong, J. Chen, and Y. R. Yang. Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM-03)*, 2003.