"TellMe": A Novel Protocol and Location Prediction Scheme exploiting the "One For All" Framework for Location Management

Amal ElNahas
Faculty of Media Engineering and Techn.
German University in Cairo, Egypt
Amal.ElNahas@guc.edu.eg

Omar H. Karam
Faculty of Computer and Information
Ain Shams University, Cairo, Egypt
Ohkaram@asunet.shams.edu.eg

Ahmad Hamad
Faculty of Computer and Information
Ain Shams University, Cairo, Egypt
Amhamad13@yahoo.com

Ingy Ramzy
Faculty of Computer and Information
Ain Shams University, Cairo, Egypt
IngyRamzy@usa.com

Abstract

Motivated by the surge of location prediction challenges and inspired by the mutual advancements in knowledge discovery and location management, this paper opts for exploiting the merits of the "One For All" (OFA) framework for moving objects databases(MOD) to address the crucial challenge of extrapolating objects' positions and balance the mainstay "bandwidth-precision" tradeoff. The proposed "TellMe" protocol and prediction scheme utilizes the OFA mobility groups to provide individual objects with accurate foreknowledge of their motion paths. Equivalently it diminishes the OFA's overheads of groups' formation. Simulation results have proven that the "TellMe" supercedes the cost savings of the OFA's current protocol and prediction scheme.

1. Introduction

The emergence of location based services has created a surge of research activities that address the various challenges of situational awareness exposed by the specially featured location information stored and maintained in Moving Objects Databases (MOD). Applications of situational awareness enabled through MOD extend to both the civilian and military fields [1][2]. MOD challenges involve devising efficient location models and update policies to cope with the dynamic nature of location information and improve the scarce bandwidth utilization yet maintain acceptable data precision.

For a moving object equipped with a positioning device reporting its position to a location server, a location model defines what information to be stored at the server and exchanged between the object and the server. Equivalently an update policy defines when an object should send a location update to optimize the bandwidth /precision tradeoff. The proposal cited in this paper is based on the "One For All" (OFA) location management framework which exploits the fact that several objects tend to move in clusters (ex: fighters and soldiers in a battlefield), hence experience the same conditions and varying bandwidth availability which finally converges them to similar deviation and update behavior [3]. Accordingly, OFA considers mobility groups, where objects with similar movement pattern are grouped into clusters and a cluster leader (the one) sends updates on behalf of other objects within its group (the all), which significantly reduces the total number of updates while maintaining acceptable precision.

In This paper we extend the OFA through the "TellMe" protocol and prediction scheme, which de-centralizes the task of mobility groups' formation, hence diminishing the incurred overheads. Furthermore, it significantly improves the ongoing location prediction process at the moving objects and the location server by relying on what the group leader has communicated regarding the adopted path.

The outline of this paper is as follows. In section 2 we survey existing location models and policies. The motivation behind the proposal is introduced in section 3. Section 4 elaborates the proposed "TellMe" protocol and prediction scheme which is evaluated through

Proceedings of the 12th International Symposium on Temporal Representation and Reasoning (TIME'05)
1530-1311/05 $20.00 © 2005 **IEEE**

experimentation presented in section 5. Finally section 6 concludes the proposed work.

2. Related Work

The Moving-Objects Spatio-Temporal model (MOST) introduced in [4] exposes the concept of dynamic attributes which change as a function of time without being explicitly updated. An explicit update is required to control the deviation between the actual and predicted object's position. In [5] an information cost model is proposed that quantifies the tradeoff between the deviation and communication costs as an information cost, upon which two proposed update policies were based. The MOST extension in [1] and [2] introduced the concept of deviation threshold (uncertainty) which resulted in extending the information cost model in [5] to account for the uncertainty. The cost model derived three dead reckoning policies: Speed (SDR), Adaptive (ADR) and Disconnection Detection (DTDR) Dead Reckoning. These policies employed the deviation threshold (uncertainty) to determine when an object should update its location information. The Trajectory Location Management cited in [6], [7], [8] and [9] utilizes prior knowledge related to object's routes and destinations to reduce the overall costs. Finally, the "One For All" (OFA) framework introduced in [3] extended the MOST model to handle objects' states and upkeep clusters/groups identifying data. OFA equivalently extended the policy by having the group leader send updates on behalf of the group members. OFA designates an analogy to the trajectory location management with no prior knowledge of objects' trips.

3. Motivation

The "One For All" (OFA) framework designates a new perspective for location management that considers mobility groups to enhance individual moving objects' behavior. The OFA consolidates research efforts in knowledge discovery and location management by having the location server deploy a dynamic clustering algorithm that attempts to attach single moving objects to their nearest centroid/leader satisfying a clustering criteria. During the objects' trip, the centroid broadcasts its location information and each follower updates its database position accordingly. Nonetheless, the followers continuously compute their deviation as the difference between the predicted position (at the location server) and its' real position. A follower detaches from the cluster, by sending an update message whenever its deviation exceeds its computed threshold [3].

Though demonstrated significant cost savings regarding the number of updates, the OFA's novelty mandates numerous challenges. Nonetheless, its trend is entailed by the nature of MOD applications either in the military or civilian fields. Moreover it promises synergistic advances in query processing, particularly for the nearest neighbor and reverse nearest neighbor queries, a tribute to its pre-defined groupings [10].

3.1. Groups/Clusters Formation Overheads

The OFA employs two types of control messages sent from the location server to the moving objects for the sake of clusters/groups formation, namely: "Attach_Follower" and "Detach_Follower" messages [3]. These are utilized by the location server to acknowledge a group centroid/leader of an incepted follower, or simply to detach a follower.

Upon cluster formation the location server sends two "Attach_Follower" messages, one to the anticipated centroid, and the other to the moving object itself. Depending on the object's state, the server may send a "Detach_Follower" request to detach an object from a previous centroid. Accordingly a moving object joining a group penalizes the server an extra cost of three update control messages two for attachment and the one for detachment.

Although the OFA entails significant cost savings, its merits could be severely challenged in case of frequent detachments introduced by hesitant moving objects or simply enforced by the underlying environment. Moreover, the OFA protocol of clusters/groups formation may result in bad groupings, since upon clustering, it utilizes prediction data to evaluate the clustering criteria. The need for prediction stems from the fact that the location server attempts to cluster an object upon the reception of its location update, meanwhile, to elect the best centroid/leader relative to this object, the server has to quantify the relative distance and speed extracted from the position attribute of the anticipated leader, which may be outdated at the time of clustering. This mandates employing prediction to extrapolate the leader's position, hence evaluate the clustering criteria at the time of group formation. Intuitively, bad formations multiply the chances of detachments and penalize the overall information cost. Accordingly, the clustering overheads should be practically diminished to enable the OFA to cope with the various movement patterns and achieve the anticipated cost savings.

Proceedings of the 12th International Symposium on Temporal Representation and Reasoning (TIME'05)
1530-1311/05 $20.00 © 2005 **IEEE**

3.2. Managing highly dynamic objects

The OFA underlying dead reckoning update policy employs a linear prediction function for extrapolating moving objects positions during their update-to-update interval. This prediction method and other mathematical methods fail to provide accurate foreknowledge of objects' motion. The challenge is an extension of prediction challenges in general that focus on how to communicate future experience.

Unfortunately, for the OFA, inaccurate prediction could highly penalize the overall cost. Consider a centroid *C* and a follower *F*, where *C* attached *F* at a relative distance of d_r and a relative speed of v_r. Assume t_1 time units after the attachment, *C* decelerated to a stop in a traffic jam causing it to send a location update reporting its zero speed v_c and position (x_c, y_c). Relative to *C's* reported position, *F's* database speed would be v_r. Shortly after *C, F* decelerates to a stop in the same traffic jam. The centroid *C* trapped in the jam for t_2 time units would cause *F* to detach, since its predicted position, t_2 time units after *C's* last update, reports a position that overshoots its real position and speed resulting in a fast deviation as a function of elapsed time. Accordingly if *F* could induce this relative distance or speed change it would have maintained its group membership since it is evident that *F* and *C* exhibited the same movement behavior but at different times.

4."TellMe": The Protocol and Prediction Scheme

The proposed protocol and prediction scheme, namely: "TellMe", exposes the full potential of the OFA framework through an enhanced de-centralized group formation protocol as well as an efficient prediction scheme. The novelty of the "TellMe" lies in grouping objects that tend to exhibit the same behavior but at different time points. This core concept enables the OFA to handle moving objects with high dynamicity, by assuming a semi-identical shifted behavior relative to the group leader. This resolves the possibly high rate of change of the relative distances and speeds. And better suits the motion of moving objects in a cluster/group, by assuming a consummate leader incepting the environmental incidents and communicating it to its followers who are expected to undergo the same experience afterwards. Accordingly the leader acts as a traffic reporter, generating anticipated best routes of motion for its followers based on real life experience.

The strength of the "TellMe" is particularly exploited by applications demanding the province of recommended routes/motion paths for a particular service member, which mandates identifying the current traffic patterns and continuously updating the member with up-to-date recommended motion paths. In this regard the "TellMe" realizes the full portfolio of benefits offered through the trajectory location management but with no prior knowledge and no overheads (for example: a traffic server, route composer,…etc). Moreover it excels to provide a practical real-life experience rather than a proposed computed route.

It is worth mentioning that the centroid/leader communicates its experience to its followers, accordingly it is mandated that it precedes the followers in behavior. Hence we shall denote the centroid as a "leader" and its cluster as a "group" to better communicate the logical concept.

4.1. "TellMe": The Protocol

The "TellMe" protocol addresses the OFA challenge of group formation overheads as well as the crucial aspect of the groupings' quality. It adopts a de-centralized approach for composing the mobility groups instead of relying on the location server to exploit these groups (centralized approach). The merit is realized by enabling moving objects to broadcast its' location information which are captured by other objects (anticipated leaders) that are contending to win appropriate followers by advertising their motion plans. The location server considers a leader offer of membership a winner if the object replies back the offer through an attachment approval. Other objects receiving the approval, discard the sender as a potential candidate for group membership. Noticeably, the objects demand to be acknowledged of other objects' states or motion paths ("TellMe") which decentralizes the group formation task and diminishes the grouping overheads. Moreover, the groups are formed based on real-time data, rather than extrapolated prediction data which reduces the chances of bad formations.

The MOST model underlying the OFA has been extended to enable the "TellMe" protocol. Firstly, a moving object should maintain a local trace of its positions and update thresholds to be able to communicate its motion plan to win followers.

Definition1. An Object Position Trace *OPT(O)* of a moving object *O* at time *t*, defines a polyline in three dimensional space *(X,Y, Time)* represented as a sequence of points (x_i,y_i,i), $(x_{i+1}, y_{i+1}, i+1),\ldots,(x_t,y_t,t)$.

Denote the object's tracked history length by *j, j:1→+∞, then i:(t-j)→t*

Definition2. An Object Threshold Trace *OTT(O)* of a moving object *O* at time *t* is represented as a sequence of values :K_{ti} , K_{ti+1} ,..,K_{tq-1}**.** Threshold K_{ti}, *i:0→q-1*,where *q* is the number of update points over *[t-j,t[* and j is the object's tracked history length where *j:1→+∞*.

Secondly, the moving object should continuously evaluate other objects as followers. Accepted objects are anticipated group candidates that are maintained in a Candidates List which is dynamically updated.

Definition3. The Candidates List CL(O) of a moving object O at time t, represents anticipated followers as a sequence of pair values: *(*O_j *,* β_j*), (* O_{j+1} *,* β_{j+1}*),.., (* O_{n-1}*,* β_{n-1}*),* where O_j , *j:0→n-1,*denotes the anticipated follower (Candidate) object id, and n, the number of O's potential candidates at t. β_j,j:0→n-1 denotes the attachment time shift of anticipated follower (Candidate) j from object O(leader).

Thirdly, the moving object should be able to formulate four types of location update messages, namely: Standard Update, Leader Advertisement, Member Advertisement and Leader Update Messages. Initially, the moving object should utilize the Standard Update to communicate its position to the server and other objects (through message broadcast).

Definition4. A Standard Update Message $M_S(O)$ of a moving object *O*, broadcasted to the location server and other moving objects, reports its position attribute (x_t,y_t), speed (v_t) and Threshold (K_t) at update time *t*.

Afterwards, the object begins receiving Leader Advertisement messages, constituting membership proposals. The message encompasses a trajectory proposal (extracted from the sender's position trace), and a proposed time shift through which the sender acknowledge the server of the shift parameter to enable the prediction scheme. It is worth noting that a Leader Advertisement message received by a follower constitutes an update rather than a membership proposal, since it is already a follower to the sender.

Definition5. The Followers List *FL(O)* of a moving object *O* at time *t*, represents actual current followers of O as a sequence of pair values: *(* O_i *,* β_i*), (* O_{i+1} *,* β_{i+1}*),.., (* O_{n-1}*,* β_{n-1}*),* where O_i,*i:0→n-1,*denotes the follower object id, β_i,*i:0→n-1* denotes the attachment time shift of follower *i* from the leader *O* and *n* denotes the number of *O* followers at *t*.

Definition6. A Leader Advertisement Update Message $M_{LA}(O)$ of a moving object *O* at *t*, Multicast to the location server and objects in *CL(O)* and *FL(O)*, reports its speed v_t, trajectory $TR_t(O) \equiv OPT_{[t-\beta,t]}$, Thresholds $TH_t(O) \equiv OTT_{[t-\beta,t]}$, and Candidates $CD_t(O) \equiv CL_t$, where the time shift $\beta \geq \beta_i$, *i:0→n-1,*β_i*:* $-\infty \rightarrow +\infty$ and n is the number of candidate and follower objects.

If the values included in the leader's advertisement message prove to be better than linear prediction, a non-follower object sends a Member Advertisement message, communicating its leader id and its position attribute, the message constitutes an approval to the leader's proposal of membership. The Leader, hearing the message, attaches it to its Followers List, while other objects remove it from their Candidates Lists.

Definition7. A Member Advertisement Update Message $M_{MA}(O)$ of a moving object O, broadcasted to the location server and other moving objects, reports its position attribute (x_t,y_t), speed (v_t), Threshold (K_t) and Leader id $LID_t(O)$ at update time *t*.

Finally, the moving object receives Leader Update Messages containing a new trajectory, and utilizes the provided trajectory to significantly decrease the deviation cost by building a local prediction trajectory that it utilizes for prediction instead of the mathematical linear prediction method. Since the group formation involves no initial overheads, objects' detachments do not constitute a threat to the overall information cost, and an object can simply detach by sending a Standard Update Message.

Definition8. An Object Prediction Trajectory OPrT(O) of a moving object O at time t, defines a polyline in three dimensional space *(X,Y, Time)* represented as a sequence of points: *(*x_i*,* y_i*, i), (*x_{i+1}*,* y_{i+1}*, i+1) ,…, (*x_r*,* y_r*,r)***.** where *i:(t+1)→r.* Denote O's leader's last update message time by t', then r≡ t'+β where β is O's attachment time shift. And each *(*x_i*,* y_i*)* designates O's leader's position at time *(i-*β*).*

Definition9. A Leader Update Message $M_{LU}(O)$ of a moving object *O*, Multicast to the location server and follower objects in FL(O), reports its speed v_t, trajectory $TR_t(O) \equiv OPT_{[t-\Delta,t]}$ and threshold $(K_t(O))$. let β_i denote the time shift of follower *i* relative to its leader *O*, and n denotes the total number of O followers, while T denotes the length of O's last update-to-update interval. Thus Δ could be expressed as follows.

$$\Delta = \begin{cases} T & \beta > T \text{ and } \beta > \beta_i \text{ , } i:0 \rightarrow n-1 \\ \beta & \beta < T \text{ and } \beta > \beta_i \text{ , } i:0 \rightarrow n-1 \end{cases} \quad (1)$$

Table 1. lists the main/basic routine running at each moving object. A leader could receive Leader advertisement messages, as indicated in lines 7 through 17. These messages advertise leaders' positions and thresholds up to the maximum potential candidate shift from the leader's update time.

Table 1. Moving Object Basic Routine

```
1  Begin
2    while (true)
3    Begin
4      Sender=Receive(UpdateMess)
5      If(MyState not "Follower")
6      Begin
7        If (UpdateMess type is Lead_Adver)
8        Begin
9          Cand=UpdateMess. CD_t
10          If(Cand  Contains(Me))
11          Begin
12            Accepted=Eval_Proposal
13        (Cand[Me]. β, UpdateMess. TR_t)
14            If(Accepted)
15            Accept_Proposal(UpdateMess,Cand)
16          End
17       End
18       Else
19       Begin
20         My_Cand_List=CL_t
21         If(UpdateMess type is Member_Adver)
22           If(My_Cand_List contains Sender)
23                Remove(Sender)
24         Else //standard message received
25         Begin
26         If(My_Cand_List not contain(Sender))
27           Begin
28             Potential_Cand=Eval_Member
29               (UpdateMess.Position)
30             If(Potential_Cand)
31               Add_To_Cand_List(Sender, β)
32           End
33           Else
34             Update_Shift (
35                    My_Cand_List[Sender])
36         End
37       End
38     End
39     Else  //I'm a follower
40     Begin
41       Update_Prediction_Trajectory
42        (UpdateMess.Trajectory, CurrentTime)
43     End
44     Send=Compute_Dev_And_Act
45                (Curr_state, Pred_state)
46   End
47  End
```

Table 2 lists the leader advertisement evaluation routine through which an object evaluates a leader proposal by comparing the deviation of the linearly predicted position ("PredState") and the proposed position ("Trajectory[Shift]").

As listed in Table 3, once a Leader is approved by an object, the object state is altered to be "follower" and it sends a Member Advertisement message declaring the proposal acceptance, then it caches the leader's trajectory ("Cash_My_Trajectory") and clears its candidates list. Denote the current time by t and proposed shift by β, then the object copies the trajectory's data from t down through to t-β.

As indicated in lines 20 through 23 in Table 1, if the object received a member advertisement message, it should remove this member from its candidates list, as it designates a declaration of fellowship to another object/leader. This action excludes this candidate from the object's next leader advertisement message. An object receiving a standard update message [Lines 24 through 36 of Table1] should evaluate the sender as a potential candidate.

Table 2. Eval_Proposal Routine

```
1  Eval_Proposal (Shift,   Trajectory)
2  Begin
3      Dev=Check_Dev(CurrState,PredState)
4      Dev1=Check_Dev(CurrState,Trajectory
5          [Shift])
6      If(Dev1<Dev)
7          return true
8      Else
9          return false
10 End
```

Table 3. Accept_Proposal Routine

```
1  Accept_Proposal(UpdateMess, Cand)
2  Begin
3      SendUpdate("Member_Adver")
4      Cash_My_Trajectory(Cand[Me].Shift,
5                  UpdateMess. TR_t)
6      Clear_Candidates_List()
7      MyState="Follower"
8  End
```

Table 4 lists the membership evaluation routine, where the object looks up the sender's position in its local position trace. This method guarantees that the sender's behavior follows the leader, since it is subset of the leader's past experience stored in its Position Trace.

An object receiving a standard update from a candidate in the candidates list, should update the shift

as indicated in Lines 33 through 35 of table 1. On the other hand, a follower should update its local trajectory, upon the inception of a leader update message as indicated in lines 40 through 43 in table 1. Finally, tables 5 and 6 detail the ongoing deviation computation routine that identifies the type of message to send if the deviation threshold was exceeded.

Table 4. Eval_Member Routine

```
1  Eval_Member (SenderPos)
2  Begin
3     MinDev=Value //user defined
4     Flag=false
5     for each time t_i in Position Trace (OPT)
6     Begin
7      Dev=Check_Dev(SenderPos,OPT[t_i])
8       If(Dev<MinDev)
9       Begin
10        MinDev=Dev
11        Flag=true
12        β =CurrentTime-t_i
13       End
14    End
15    Return Flag
16 End
```

Table 5. SendUpdate Routine

```
1  SendUpdate(Type)
2  Begin
3     If( Type is "Leader_Update")
4        Send(LeaderUpdateMessage)
5     Else
6        If( Type is "Member_Adver")
7           Send(MemberAdvertisement)
8        Else
9           If(Type is "Lead_Adver)
10          Begin
11             Send(LeaderAdvertisement)
12             Clear_Candidates_List()
13          End
14          Else
15          Begin
16             Send(StandardUpdateMessage)
17            MyState= "Leader"
18          End
19    End
```

Table 6. Compute_Dev_And_Act Routine

```
1  Compute_Dev_And_Act(CurrState, PredState)
2  Begin
3     Send=Check_Dev(CurrState,PredState);
4     If(Send)
5     Begin
6        If( Cand_List is not empty)
7           SendUpdate("Lead_Adver")
8        Else
9           If(Followers_List is not empty)
10             SendUpdate("Leader_Update")
11          Else
12             SendUpdate("Standard")
13       End
14 End
```

Moving Object States: Noticeably, three main moving objects' states, namely: Single, Leader and Follower, derive the messages exchange scenarios of the proposed "TellMe" protocol. As depicted in the state transition diagram of figure 1, a single object *O* alters its state from "Single" to "Leader" upon the reception of a Member Advertisement update message ($M_{MA}(O_i)$) from a "Single" moving object O_i confirming the acceptance of O's membership proposal. Alternatively, an empty Followers List *(FL(O))* at a leader *O*, causes state alteration to "Single". An object *O* switches to a "Follower" state upon sending a Member Advertisement update message ($M_{MA}(O_i)$), whereas sending a Standard Update messages switches a "Follower" object to the "Single" state.

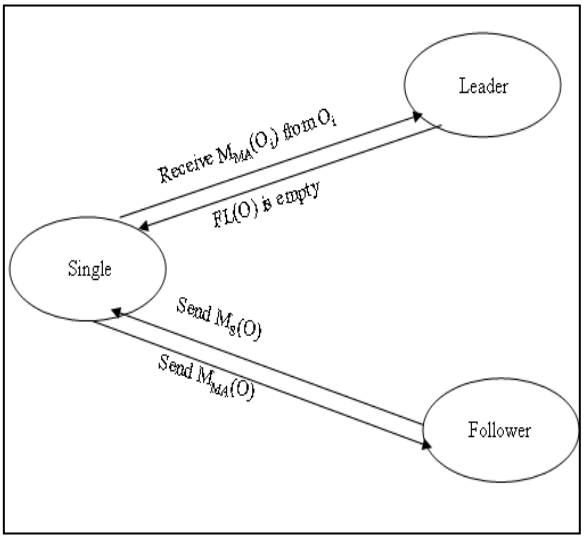

Figure 1. Moving Object States

The protocol diagram of figure 2 demonstrates the messages exchange scenarios of the "TellMe" protocol. The diagram denotes single and leader objects as "Free Objects", while the location server and follower objects are represented as separate entities.

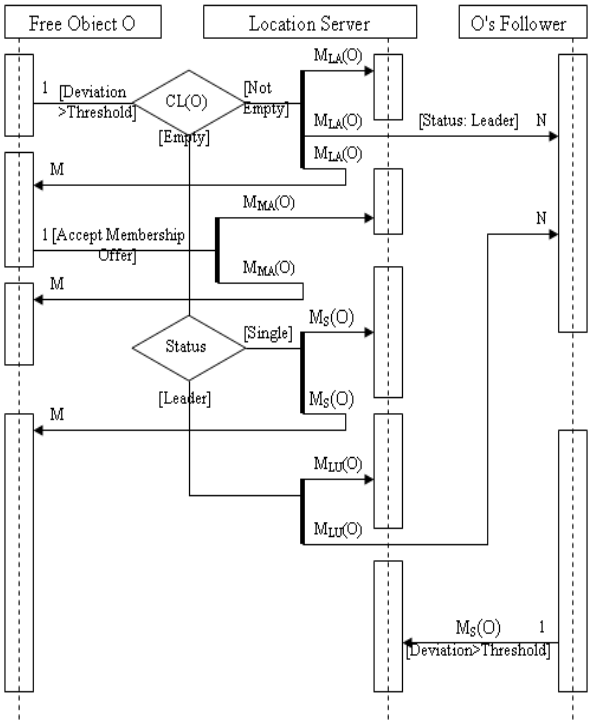

Figure 2. The TellMe protocol

4.2. "TellMe": The Prediction Scheme

Consider a moving object O, moving at speed v_t and is at position (x_t,y_t) at time t. Assume that O's current deviation threshold is K_t. Accordingly, O should send a location update if and only if its current deviation exceeds K_t as dictated by the Adaptive Dead Reckoning Policy (ADR) ([1], [2]). O's current deviation is computed by differencing O's current position (x_t,y_t) and its predicted position $(x_p(t),y_p(t))$, which is obtained using a linear predictor that utilizes the current speed v_t and time elapsed T since the last update to estimate the traveled distance (v_t*T) since the last update time $(t\text{-}T)$ ([1], [2]). Other nonlinear prediction functions can be employed for the OFA to provide better prediction.

The "TellMe" prediction scheme utilizes the model introduced by the "TellMe" protocol. It makes use of the moving object's local Prediction Trajectories to accurately predict the current object location. For a moving Object O, with Prediction Trajectory $OPrT[t_1,t_2]$, its predicted position at t is as follows.

$$P(O)=\begin{cases} OPrT(t)+\varepsilon & t\in [t_1,t_2] \\ P(O)+v_t\check{T} & t\notin [t_1,t_2], \\ & \check{T}=t-t_2 \end{cases} \quad (2)$$

ε denotes the attachment deviation error and $\check{T}$ denotes the time elapsed since updating the position by $OPrT(t_2)$. The "TellMe" prediction scheme utilizes the local prediction trajectory for prediction, but if the object consumed all the prediction trajectory entries $([t_1,t_2])$ before being updated by new entries, the object should utilize linear or other mathematical prediction function to extrapolate the position after t_2.

It is evident that as the attachment time shift β increases the prediction trajectory entries increase, which improves the ongoing prediction process. Meanwhile, increased β entails more chances of environmental changes which deviates the follower from its leader. Nonetheless, β does not affect the length of the leader update message, since it is bounded to the length of the leader's update-to-update interval, whereas it has an insignificant effect on the initial Leader Advertisement message since it is sent only once.

5. Performance Evaluation

Dr. Brinkhoff's Framework for generating Network-Based moving objects has been utilized for generating the required spatiotemporal data [11], [12]. The framework generates objects moving on road networks and exposes them to the influence of external events, and time-scheduled traffic. Moreover it considers important aspects of movement in a network including the maximum speed and the maximum capacity of the connections resulting in a data that combines the features of both real and statistically generated data. A model-policy simulator was built on top of the framework to process the generated data and evaluate the various location update policies and models. The model-policy simulator has been utilized to compare the proposed protocol and prediction scheme, namely: "TellMe", and the "One For All"(OFA) as well as the Adaptive Dead Reckoning Policy (ADR). The policies are evaluated based on the average total information cost of the objects' trips. The cost is expressed in terms of the total update, deviation and uncertainty costs introduced in [1] and [2]. It should be noted that in order to test the OFA or the proposed "TellMe" protocol, the objects should have tendency to move in

groups, accordingly the generator behavior has been modified to expose the required group mobility patterns.

Figures 3 through 6 demonstrate the results of two different experiments. Note that the experiments did not include the OFA's control messages' cost, although it penalizes the OFA's information cost as a result of numerous attachments and de-attachments. The reason is that it was required to compare the policies based on the cost savings realized for the moving objects during their trips, and being an overhead of the location server rather than the objects, the control messages are not included in the figures 3 through 6.

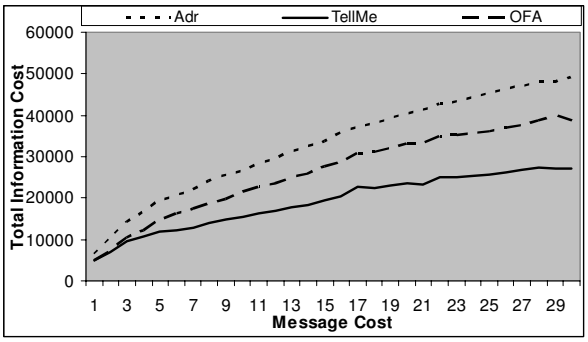

Figure 3. Total Information Cost versus the Message Update Cost of Experiment 1

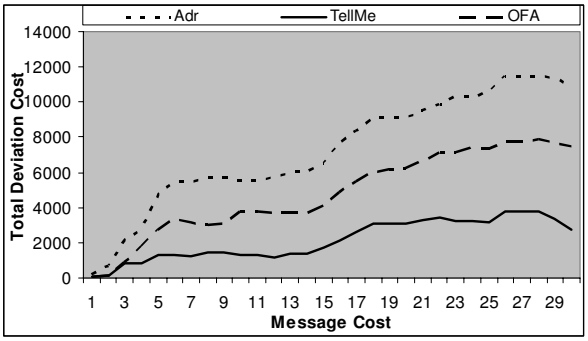

Figure 4. Total Deviation Cost versus the Message Update Cost of Experiment 1

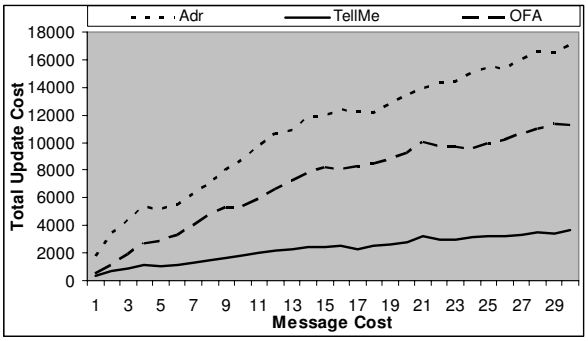

Figure 5. Total Update Cost versus the Message Update Cost of Experiment 1

The first experiment considered five objects for 400 time units using an uncertainty relative cost of 0.75, measuring the total information cost, against different message update relative costs. The relativity is to the deviation cost which is considered 1. The generator's speed divisor was set to 50 to generate objects that are neither too fast, nor too slow. Simulation results proved that the "TellMe" protocol and prediction scheme reduces the total information cost and its components beyond the cost savings introduced by the OFA. It should be noted that the OFA experienced several detachments as a result of its adopted protocol and prediction scheme, and that accounts for its cost increase versus the ADR. In fact in the absence of detachments and with objects that exhibits higher identicality, the OFA recorded significant cost savings [3]. Though subjected to the same conditions, the "TellMe" survived the challenge and recorded a cost decrease of 40% in contrast to the OFA that resulted in a maximum of 18% decrease in the total information cost. Figure 4 demonstrates that the "TellMe" decreased the deviation cost by 66% while the OFA decreased it by a maximum of 27%. Concerning the update cost, and as demonstrated in figure 5, the "TellMe" decreased the cost by 80% while the OFA decreased it by a maximum of 32%.

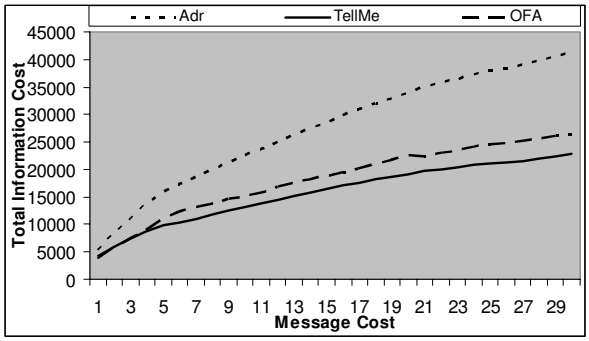

Figure 6. Total Information Cost versus the Message Update Cost Of Experiment 2

The second experiment considered the same settings but the speed divisor was set to 250 to generate slow moving objects. The generated objects exhibited movement patterns with a slow rate of change resulting in a small number of updates as a result of the slow deviation patterns. As evident in figure 6 the "TellMe" approached the OFA cost savings, since the linear prediction was then used to extrapolate the object's positions due to the lack of entries in the Prediction Trajectory as a result of the seldom location updates. Moreover the OFA exposed better results as a result of the decreased objects dynamicality. Finally, it should

be noted that the cost savings are a function to a great extent in the variance of the objects' behavior within the groups.

6. Conclusions and Future Work

Moving Objects Databases (MOD) possesses several challenges, particularly for modeling and updating location information. In this paper we have proposed a protocol and prediction scheme as an extension to the "One For All"(OFA) framework introduced in [3]. The OFA considered mobility groups to realize significant cost savings per individual moving object, nonetheless the novelty of the OFA framework exposed numerous challenges. The proposed "TellMe" protocol and prediction scheme opts for addressing the challenges of cluster formation and managing objects of high dynamicality. The proposal migrated the group formation protocol to a de-centralized approach through an enhanced dialogue among the objects. Moreover, it enabled the group leader to communicate its past experience, to enhance the prediction of its followers, so the leader is "Telling" the objects what happened which significantly reduces the overall costs as proven by experimentation.

New challenges are added up by the introduction of the "TellMe" protocol and prediction scheme. One of the main challenges is to enable moving objects within a group to develop a sense of ongoing membership evaluation versus the individual behavior. The concept is vital to discover false memberships that may escalate the information cost and result in a definite detachment. Even if an object exhibited a true fellowship to its leader, it should be allowed to switch to the singular mode as needed to maneuver the environmental challenges that may give a false indication of its fellowship. Although the "TellMe" diminishes the control messages' overheads that penalized the group formation process, the mode switching should not restart the group membership evaluations, instead it should be seamless and transparent to the main three tiers involved, namely: the object, the location server and the leader. The constraint is crucial to enable the moving object to switch its mode at will and whenever needed with no considerations or overheads. Moreover, moving objects with hesitant movement patterns should not be considered in leaders' advertisement messages, accordingly a crucial challenge is to discover these hesitant objects and exclude them from the advertisements. Another challenge is quantifying the maximum optimum attachment time shift that exploits the protocol and scheme's strength, since a small shift, entails less prediction points, and a big one, threats the objects' motion similarity. Furthermore, quantifying the optimum length of the positions and thresholds traces maintained at the objects exposes tangible benefits to the flexibility and practicability of the whole framework.

7. References

[1] O. Wolfson, P. Sistla, S. Chamerlain, Y. Yesha, "Updating and Querying Databases that Track Mobile Units", *Distributed and Parallel Databases*, *7*(3), 1999, pp. 257-387.

[2] O. Wolfson, B. Xu, S. Chamberlain, L. Jiang, "Moving Object Databases: Issues and Solutions", *Proc. 10th International Conf. on Scientific and Statistical Database Management,* Capri, Italy, 1998, pp. 111-122.

[3] O. Karam, A.Nahas, A.Hamad, I.Ramzy, "One For All: A new perspective for Modeling and Updating Location Information in Moving Objects Databases", *The 2nd International Conference on Communication Systems and Networks*, Spain, Sept 2003, pp. 175-180.

[4] P. Sistla, O. Wolfson, S. Chamberlain, S. Dao, "Modeling and Querying Moving Objects", *Proc. 13th International Conf. on Data Engineering,* Birmingham, UK, 1997, pp. 422-432.

[5] O. Wolfson, S. Chamberlain, S.Dao, L. Jiang, G. Mendez, "Cost and Imprecision in Modeling the Position of Moving Objects", *Proc. 14th International Conf. on Data Engineering,* Orlando, FL, 1998, pp. 588-596.

[6] O. Wolfson, "Moving Objects Information Management: The Database", *Proc. 5th Workshop on Next Generation Information Technologies and Systems*, Caesarea, Israel, 2002, pp. 75-89.

[7] G. Trajcevski, O. Wolfson, F. Zhang, S. Chamberlain, "The Geometry of Uncertainty in Moving Objects Databases", *Proc. 8th International Conf. on Extending Database Technology,* Berlin, Germany, 2002, pp. 233-250.

[8] G. Trajcevski, O. Wolfson, C. H. Lin, F. Zhang, N. Rishe, "Managing Uncertain Trajectories of Moving Objects With DOMINO", *Proc. 4th International Conf. on Enterprise Information Systems,* Spain, 2002, pp. 217-224

[9] G. Trajcevski, O. Wolson, B. Xu, P. Nelson, "Real-Time Traffic Updates in Moving Objects Databases", *Proc. 13th IEEE International Workshop on Database and Expert Systems Applications,* 2002, pp.1529-4188/02.

[10] Ritmantas Benetis, Christian S. Jensen, Gytis Karciauskas, Simonas Saltenis, "Nearest Neighbour and Reverse Nearest Neighbour Queries for Moving Objects", TimeCenter Technical Report, 2001.

[11] T. Brinkhoff, "A Framework for generating network-based moving objects", *Geoinformatics, 6*(2), 2002, pp. 153-180.

[12] T. Brinkhoff, "Generating Network-Based Moving Objects", *Proc. 12th International Conf. on Scientific and Statistical Database Management,* Berlin, Germany, 2000, pp. 253-255