A Decentralized Approach towards Location Tracking of Mobile Users in Opportunistic Networks

Sudipa Batabyal[1], Apratim Mukherjee[2], Somprakash Bandyopadhyay[3],

[1] Cognizant Technology Solutions, Salt Lake, Kolkata 700091, India
[2] BPP Institute of Management & Technology, VIP Road, Kolkata 700052, India
[3] Indian Institute of Management Calcutta, Joka, Kolkata 700104, India

Abstract. In existing models of opportunistic networks, nodes are not usually topology-aware. In this paper, we have developed a distributed mechanism in order to make each node in the network topology aware, i.e. aware of the approximate location-related GPS information of other nodes. The mechanism proposed in this paper is primarily developed on mobile agent based framework. It is assumed that each node has a dedicated *satellite agent*. Task of this agent is to help exchanging GPS information between its host node and its neighboring nodes only. This nearest neighbor interaction rule will eventually enable each node to have approximate location information about other nodes. As a direct consequence of topology-awareness, data forwarding schemes becomes far more efficient. We have introduced a parameter W (degree of disconnectedness) and show that our system is robust against wide variation of W. The performance evaluation results establish the effectiveness of the proposed scheme.

Keywords: Opportunistic Networks, Multi-Agent System, Location Tracking.

1 Introduction

Opportunistic networks and opportunistic computing have drawn significant attention of researchers in the recent past. In opportunistic networks, the devices (PDA, multi-radio cell-phones and similar devices) spread across an environment form the network. In this type of networks, end-to-end route connecting any two nodes usually does not exist and a source node communicates with its destination node following hop-by-hop, store-wait-forward cycle. In this type of networks, the mobility of devices is an opportunity for communication rather than a challenge. Thus, a mobile node can communicate with other nodes even if an end-to-end route connecting them never exists; any possible node can opportunistically be used as the next hop, if it is likely to bring the message closer to the final destination(s). Designing routing and forwarding schemes is one of the main challenges in this environment [1,2].

In existing models of opportunistic networks, nodes are not usually topology-aware i.e. nodes are not aware about the geographic locations and connectivity patterns of other nodes. However, in many applications, it is important to track the location of mobile users for effective communication. For example, in the context of disaster management (e.g. flood or earthquake), let us assume that fixed infrastructure (mobile towers, etc.) is non-functional and rescue workers with their wireless personal mobile communication devices form an opportunistic network. In this context, public health department, after reaching the site with a team of medical expert, must be able to locate other health workers, who are already working in the field. Secondly, in order to distribute relief resources to the designated rescue workers, those workers must be located and the agency carrying the relief resources must be able to send a query to those designated workers in order to assess the requirements. Also, the organizers need to delegate tasks to volunteers working in the field, and therefore those volunteers need to be located. Once the approximate location information of destination node is known to a source node, message communication can be implemented using Geographic Routing [3].

In this paper, we have developed a distributed mechanism in order to make each node in the network aware of the approximate location-related GPS information of other nodes. The degree of accuracy of this information would depend on the inter-node distance and connectivity pattern of the opportunistic network. The mechanism proposed in this paper is primarily based on mobile agent based framework [4]. It is assumed that each node has a dedicated *satellite agent.* Task of this agent is to help exchanging GPS information between its host node and its neighboring nodes. This nearest neighbor interaction rule will eventually enable each node to have approximate location information about other nodes. .

2 Related Work

In opportunistic networks, the notion of data forwarding and routing are merged, because routes are actually built while messages are forwarded [2]. The forwarding scheme has been primarily referred as "store, carry, and forward". Each intermediate node evaluates the suitability of encountered nodes to be a good next hop towards the destination. Another form of routing technique exploits some form of flooding. The heuristic behind this policy is that, when there is no knowledge about a possible path towards the destination or of an appropriate next-hop node, a message should be disseminated as widely as possible. The most representative protocol of this type is Epidemic Routing [5] and some optimizations of the same [e.g.,6]

However, flooding-based approach generates multiple copies of the same message. In Forwarding-based approach, though there is only one single custodian for each message, it may suffer long delays and low delivery ratios. Several schemes have been proposed considering mobility pattern / context information into account. The Haggle Project [7] has developed mechanisms for measuring and modeling pair-wise contacts between users and devices by means of two parameters: contact durations and inter-contact times. The statistical properties of these parameters are used to drive the design of forwarding policies. *Probabilistic Routing* scheme [8] calculates the *delivery predictability* from a node to a particular destination node based on the observed contact history, and it forwards a message to its neighboring node if and only if that neighbor node has a higher delivery predictability value. Leguay *et al*. [9]

have taken *mobility pattern* into account, i.e., a message is forwarded to a neighbor node if and only if the neighbor node has a mobility pattern more similar to the destination. However, in many application scenarios (such as ours), mobility patterns are largely unpredictable.

Ghosh et al [10] propose routing based on the predefined infrastructure, such as the places that device holders often visited; they call them "solar-hub". This takes the advantage of user mobility profiles to perform "hub-level"-based routing. However, in this scenario, it is necessary to know about the places visited by the receiver. Exploiting context information related to the social behavior of people is also one of the most promising research directions in the area [11].

However, data forwarding schemes becomes far more efficient, if approximate locations of nodes are known to other nodes. In this paper, we will address this issue using mobile agent-based framework. Use of mobile agents are an effective paradigm for distributed applications, and are particularly attractive in a dynamic network environment involving partially connected computing elements. Intensive research on the "Insect-like Agent Systems" has been done over the last few years. Of particular interest is a technique for indirect inter-agent communication, called stigmergy, in which agents leave information in the cache (which other agent can use) of the nodes they have visited. Stigmergy serves as a robust mechanism for information sharing [4].

As indicated earlier, we propose to use ***satellite agents*** for distributed location tracking. In a seminal paper in Physical Review Letters, Vicsek et al.[12, 13] propose a simple model of n autonomous agents moving in the plane with the same speed but with different headings. Each agent's heading is updated using a local rule based on the average of its own heading plus the headings of its "neighbors." In their paper, Vicsek et al. demonstrated that the nearest neighbor rule can cause all agents to eventually move in the same direction despite the absence of centralized coordination and despite the fact that each agent's set of nearest neighbors change with time as the system evolves. Other studies also indicate that multi-agent systems that interact through nearest-neighbor rules can synchronize their states regardless of the size of communication delays [14]. We have applied this concept in our system and the performance evaluation results indicate the effectiveness of our approach.

3 System Description

An Opportunistic Network is modeled as a time-dependent disconnected graph $G(t) = (N, L, \tau)$ where N is a finite set of nodes, L is a finite set of unidirectional links and τ is a set of time-values indicating life-span associated with the links. Each link $L_i \in L$ is associated with $\tau_i \in \tau$, indicating life-span of L_i at time t.

Since the graph represents an opportunistic network, graph G(t) should usually be disconnected. G(t) consists of multiple connected pieces called components C(t). When G(t) is fully connected, C(t) =1; when G(t) is fully disconnected, C(t) = number of nodes N.

To characterize an opportunistic network, it is important to define a parameter W that indicates the degree of disconnectedness over a period of time T. W is said to be

0%, if, for each pair of nodes, there always exists a path between them between <0..T>. W is said to be 100%, if, for each pair of nodes, no path exists between them at any point of time between <0..T>. In the first case, the network is always connected and ***ceases to be an opportunistic network***. In the other extreme, the graph is fully disconnected and the set of nodes will never form a network.

In order to quantify W, we need to take a set of snap-shots of G(t). $G(t_i)$ is snap-shot of G at $t=t_i$. W_i, the degree of disconnectedness for $G(t_i) = \{C(t_i)-1\}/(N-1)$. When $G(t_i)$ is fully connected, $W_i = 0$; When $G(t_i)$ is fully disconnected, $W_i = 100\%$;

W is the average of W_i over the number of snap-shots taken. So, if number of snap-shots taken is α,

$W = [\Sigma_{i=1 \text{ to } \alpha} W_i] / \alpha$

While designing and testing the robustness of any algorithm designed in the context of opportunistic networks, it is important to consider the parameter W. Specially, while testing and validating algorithms designed for opportunistic network in a simulated network environment, disregarding the parameter W may result in a network condition where nodes are always forming a network or forming a network with a few numbers of disconnected components. Any algorithm designed for opportunistic network should work well for a wide range of W.

We define the ***physical neighbors*** of node n at time t as $N_n(t) \in N$, where $N_n(t)$ is the set of nodes within the transmission range of n at time t. It is assumed that each node knows its position, velocity and direction of movement using Global Positioning System (GPS). It is also assumed that each node periodically broadcast a beacon with its id to all its physical neighbors at that instant of time.

The mechanism proposed in this paper is primarily based on a mobile multi-agent based framework. We assume that each node n_i has a dedicated ***satellite agent*** S_i. Task of S_i is to help exchanging information between its host node n_i and each of the neighboring nodes of n_i. To do this, the satellite agent S_i periodically hops from n_i to one of its neighbors with all location-related information as perceived by n_i. The neighboring node has a different perception regarding location-related information of other nodes. S_i and the neighboring node mutually exchange this information, forms a "consensus view" regarding the location related information of other nodes and S_i then comes back to the host node n_i with this "consensus view". This would then change the perception of n_i about location-related information of other nodes. In the next time-slots, S_i visits other neighboring node of n_i and the process is repeated.

4 Agent-based Mechanism for Location Tracking

In our mechanism, each ***satellite agent*** interacts with their respective neighboring nodes only and come back to the respective host node with a localized "consensus view" about location-related information of other nodes. The technique used here for indirect inter-agent communication is ***stigmergy***, in which agents leave information in the cache (which other agent can use) of the nodes they have visited.

Structure and Behavior of a Node

In order to facilitate agent-based distributed location tracking, each node is assumed to have the following structure:

- Node Id (n_i)
- Current Location (x_i, y_i)
- Neighborhood List : <for all k : list of n_k>
- Information about all nodes (n_p) → (for all p : n_p, x_p, y_p , $timer_p$ ↑)

Each node is assumed to know its id and current location (through GPS). Each node broadcast a periodic beacon to its neighbors to inform its id. This would help a node to form Neighborhood List. Initially, the Location Table of a node would contain only the location information of itself only. Location Table of node n_i would be augmented by : (i) visits of satellite agents from other neighboring nodes containing the neighborhoods' perception (ii) returning of satellite agent of node n_i from a neighboring node containing the "consensus view" of n_i and that neighboring node.

It is to be noted that the entire scheme is based on multi-agent interaction via Location Table of nodes. Since navigation of satellite agents is asynchronous and there is an obvious time gap between the updation of information by one satellite agent in one node and carrying this information to another node by another satellite agent, there is a notion of timer↑ with each entry, depicting the ageing of information. The information is aging as agents percolates from one node to another and the nodes will therefore have new information about close neighbors and old information about remote nodes.The symbol ↑ indicates that each timer is counting up locally, unless overwritten by more recent information about that entry.

This is illustrated in figure 1. For the sake of simplicity, we are assuming unidirectional communication and we only show percolation of location information of node n1 to node n4. In figure 1, current location of node N_1 is x_1, y_1. Since this information is always recent, no aging factor is associated with it and timer value is 0. Let us assume, this information is taken to node N_2 by N_1's satellite agent S1 at t_1=0. The timer associated with x_1, y_1 now starts getting incremented at node N_2, indicating that the location information of N_1 at N_2 is aging. At t_2=10, say, N_2's satellite agent S_2 carries this information to N_3. The timer associated with x_1, y_1 at N_3 now starts getting incremented at node N_3 with a starting value of 10. Let us assume further that after 25 time-unit, i.e. at t_3=25, N_3's satellite agent S_3 carries this information to N_4. The timer associated with x_1, y_1 at N_4 now starts getting incremented at node N_4 with a starting value of 35. This implies that the perception of node N_4 about the location of N_1 is 35 time-unit old and it is aging, unless overwritten by some more recent location information of N_1.

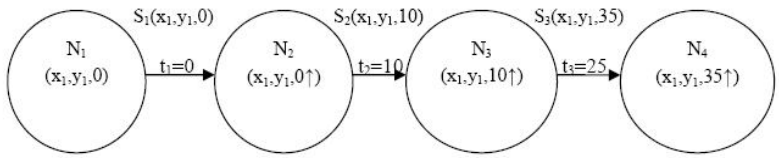

Figure 1 Percolation of Location Information

Structure and Behavior of a Satellite Agent

As explained earlier, satellite agent is a dedicated agent for any node. Its task is to carry information from its host node to neighboring nodes in a time-sequenced fashion (i.e. one neighbor at a time) and return to its host node with information of that neighboring node. For example, if node N_2's satellite agent has not visited N_1 in the recent past, the satellite agent of Node N_1 visits Node N_2, updates N_2's table to form a consensus view between N_1's and N_2's perception, then comes back to N_1 with this information. After some time, it again collects the latest information from N_1, and moves to another neighboring node, say N_3, provided N_3's satellite agent has not visited N_1 in the recent past. This process occurs for every node since each node has a dedicated satellite agent.

If a satellite agent loses its host node due to mobility of host node, it kills itself. The host node, on the other hand, generates a new satellite agent if the satellite agent doesn't come back after a certain time. The structure of a satellite agent is given below:

- Host Node Id (n_i)
- Current Location of Host Node (x_i, y_i)
- Location of Target Neighbor to be visited now
- Information about all nodes (n_p) $\rightarrow$ (for all p : n_p, x_p, y_p , $timer_p \uparrow$)

5 Performance Evaluation

5.1 Evaluation Criteria

***Average Perception Deviation of a node*:** We have developed a metric Average Perception Deviation of a node i, denoted by P_i (t), to quantify the deviation of actual node position of each node with the node-position of corresponding node perceived by node i at any instant of time t. Let us assume that (x_k, y_k) is the actual co-ordinates of node k at time t. Let (x_k^i, y_k^i) be the coordinate of node k as perceived by node i.

$$P_i(t) = \Sigma_{k=1 \text{to } n} [(x_k - x_k^i)^2 + (y_k - y_k^i)^2]^{1/2} / n$$

Average Perception Deviation of a network N with n number of node, denoted by $P_N(t)$, is defined as

$$P_N(t) = \Sigma_{i=1 to n} P_i(t) / n$$

Wait-before-Migrate (WbM): In order to control agent-traffic in the network, a satellite agent, after finishing its first visit to a neighboring node of its host node , is not allowed to migrate immediately to another neighboring node. A satellite agent will be forced to wait in its host node for a pre-specified period of time, termed as *Wait-before-Migrate (WbM)* before migrating to another neighboring node. By controlling WbM, the network congestion due to satellite agent traffic can be controlled. For example, if WbM = 200 msec, and an agent takes approximately 4 msec. to physically migrate from one node to another, the agent traffic (going from and coming back to host node) would occupy 8 ms out of 200 msec. i.e 4%. So, any host node would be free to communicate 98 percent of the time. On the other hand, increasing WbM reduces information percolation efficiency. Thus, the trade off is between congestion and convergence.

Degree of Disconnectedness: In section 3, we have introduced a parameter W that indicates the degree of disconnectedness over a period of time T. In our simulation, by controlling the transmission range, we have controlled W and evaluated the performance.

5. 2 Simulation Setup

The proposed schemes are evaluated on a simulated environment under a variety of conditions to estimate average perception deviation against time. In the simulation, the environment is assumed to be a closed area of 1500 x 1500 square meters in which mobile nodes are distributed randomly. We present simulations for networks with 40 mobile hosts, operating at a transmission range from 150 to 250 meters. In order to study the time-related parameters, every simulated action is associated with a simulated clock. The speed of movement of individual node ranges from 2 m/sec (walking) to 10 m/sec (vehicle). Each node starts from a starting location, selects a random direction and moves with a uniform, predetermined velocity along that direction. We define a set of 10 waiting-zones within the area. Once a node reaches a waiting-zones, it waits there for a pre-specified amount of time, selects randomly another direction of movement and moves towards that.

5.3 Results and Discussions

Figure 2 shows the average perception deviation $P_N(t)$ w.r.t time at different Wait-before-Migrate where mobility is randomized between of 2 m/ sec and 10 m/ sec. With 40 nodes in the system, transmission range is adjusted to get an approximate average degree of disconnectedness W= 28%, indicating that the network is having around 10 to 12 disconnected components (fig. 3).

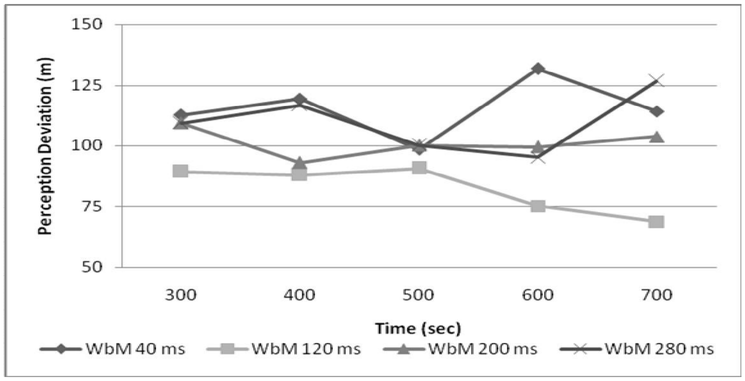

Figure 2 Average Perception Deviation with time at different WbM

At WbM=120 to 200 msec., average perception deviation is around 75 to 100 meters. So, considering the maximum possible perception deviation to be 2100 meters in a 1500 m x 1500 m area, the perception deviation is around 3.5 % to 5 %. However, when WbM is low (=40 msec)., perception deviation is higher because of congestion due to agent traffic. Perception deviation is also higher, when WbM = 280 msec, although the value is not significantly high even at WbM = 280 Msec ($P_N(t) <$ 125 meters i.e. 6%).

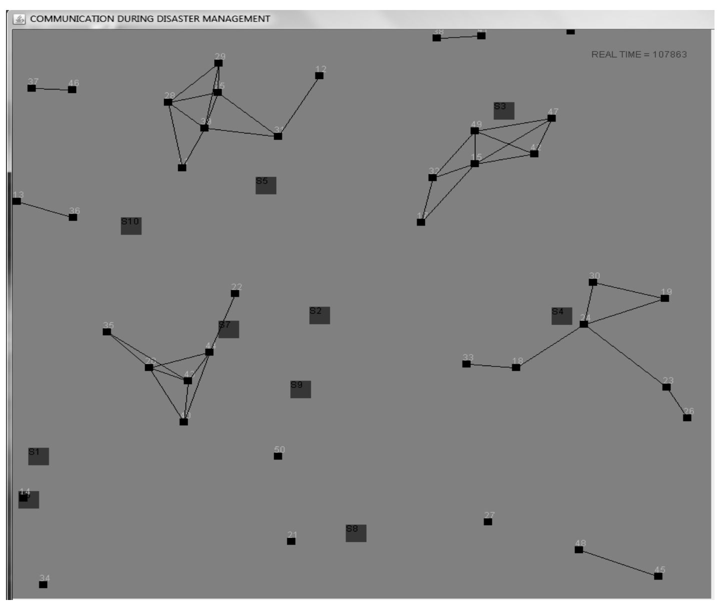

Figure 3 A snap-shot of the opportunistic network in our simulator: squares are the waiting-zones and dots are the nodes

Next, we have considered $P_N(t)$ vs time at different W (Degree of Disconnectedness) with mobility randomized between of 2 m/ sec and 10 m/ sec. (figure 4). For the sake of simplicity, we have mentioned approximate value of W in the graph. With increase in W, $P_N(t)$ will increase as expected. But even with W=50% (number of disconnected components is around 20 in a 40-node system), average perception deviation is around 10%.

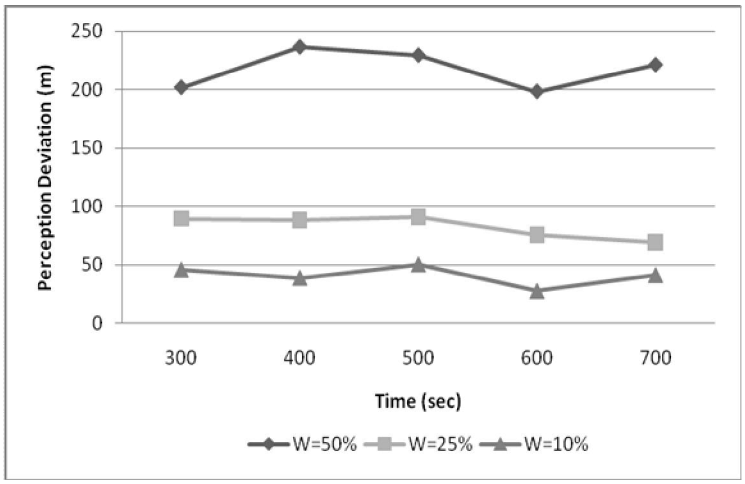

Fig 4 Perception Deviation with Time at different W

Next, we have adjusted the transmission range to get a degree of disconnectedness W=50% (approximately). i.e. number of average disconnected components = 20 in a 40-node network. We have studied the average perception deviation against time at different mobility. In the graph (figure 5), we can see that the perception deviation generally increases when velocity is increased from 2 m/s to 10 m/s. It is mainly due to increased rate of change of actual position with increased velocity. So due to this, the perception deviation tends to be high. But as we increase the velocity even higher to as high as 20 m/s, the perception deviation seems to decrease. The reason is that, although the rate of change of actual positions is high (as it was with velocity 10 m/s), at the same time, due to increased mobility, information is percolated faster over longer distances. The up-down trends in the graph indicate the information convergence-divergence pattern. Because of mobility, the information stability would never happen, but at the same time, it would also never diverge away beyond a certain point; information will again start converging. This up-down pattern is more pronounced when mobility is high.

6 Conclusion

In this study, we have designed a mobile agent based mechanism to make the nodes position-aware about other nodes in the network**.** It has been assumed that each node knows its position and velocity at any instant of time using GPS and this information is getting distributed to other nodes through agents. We have assumed that agents do not get lost in transit nor suffer from any kind of errors in transmission and reception. We have introduced a parameter W (degree of disconnectedness) and show that our system is robust against wide variation of W. Even when the mobility is as high as 20m/ sec (72 km/hr), average perception deviation is 10 to 12 % (figure 5), indicating the success of the proposed mechanism.

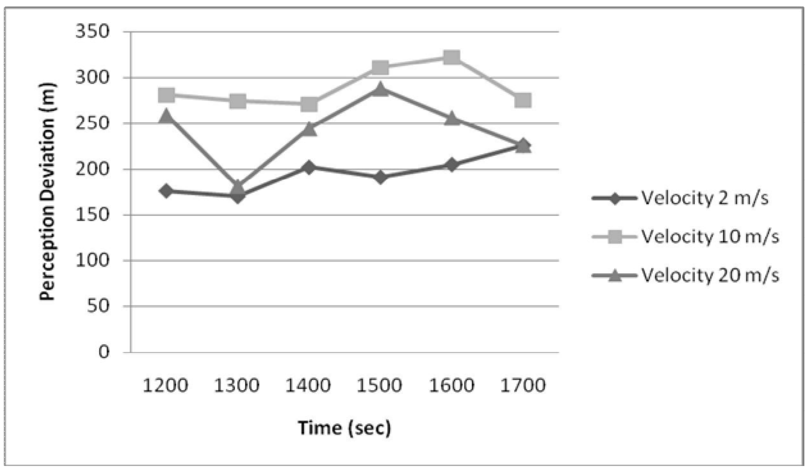

Figure 5 Effect of Mobility on Perception deviation at W=50%

References

1. Marco Conti, Mohan Kumar: Opportunities in Opportunistic Computing. IEEE Computer 43(1): 42-50 (2010)
2. Conti, M.; Giordano, S.; May, M.; Passarella, A.; From opportunistic networks to opportunistic computing, IEEE Communications Magazine, IEEE , Volume: 48 Issue:9, Sept. 2010
3. Somprakash Bandyopadhyay and Krishna Paul, "Evaluating The performance of mobile agent based message communication among mobile hosts in large ad hoc wireless network", Proc. of the Second ACM International Workshop on Modeling and Simulation of Wireless and Mobile Systems (In conjunction with IEEE/ACM MobiCom'99), Seattle, Washington, USA, August 15-19, 1999.

4. Romit RoyChoudhury, Krishna Paul, and Somprakash Bandyopadhyay, "MARP: A Multi-Agent Routing Protocol for Mobile Wireless Ad Hoc Networks" Autonomous Agents and Multi-Agent Systems (Kluwer), 8 (1): 47-68, January 2004.
5. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. Technical Report CS-2000-06, Duke University, 2000.
6. K. A. Harras, K. C. Almeroth, and E. M. Belding-Royer. Delay tolerant mobile networks (dtmns): Controlled flooding in sparse mobile networks. In IFIP Networking, 2005.
7. Augustin Chaintreau,Pan Hui, Jon Crowcroft,Christophe Diot, Impact of Human Mobility on the Design of Opportunistic Forwarding Algorithms, IEEE INFOCOM 2006, April 23-29, Barcelona, Spain
8. Lindgren and A. Doria. Probabilistic routing protocol for intermittently connected networks. Technical report, draft-lindgren-dtnrgprophet- 01.txt, IETF Internet draft, July 2005.
9. J. Leguay, T. Friedman, and V. Conan. Dtn routing in a mobility pattern space. In ACM SIGCOMM Workshop on Delay Tolerant Networks, 2005.
10. Ghosh, J., Sumesh J. Philip, & Chunming Qiao (2007). Sociological Orbit aware Location Approximation and Routing (SOLAR) in MANET. ELSEVIER Ad Hoc Networks Journal, Vol. 5, Issue 2, Pgs 189-209, March 2007
11. Hui, P., Crowcroft, J., & Yoneki, E. (2008, May). Bubble rap: Social-based forwarding in delay tolerant networks. Paper presented in the Proceedings of the 9th ACM international symposium on Mobile ad hoc networking & computing (MobiHoc'08).
12. T. Vicsek, A. Czirok, E. Ben Jacob, I. Cohen, and O. Schochet. Novel type of phase transitions in a system of self-driven particles. Physical Review Letters, 75:1226– 1229, 1995.
13. Czirok, A. L. Barabasi, and T. Vicsek. Collective motion of self-propelled particles:kinetic phase transition in one dimension. Physical Review Letters, 82:209–212, 1999.
14. Herbert G. Tanner and Dimitrios K. Christodoulakis, "The stability of synchronization in local-interaction networks is robust with respect to time delays," 44th IEEE Conference on Decision and Control, 2005.