Brownian Gossip: Exploiting Node Mobility to Diffuse Information in Ad Hoc Networks

Romit Roy Choudhury
University of Illinois at Urbana-Champaign
croy@uiuc.edu

Abstract— **Several network services, including routing, resource-discovery, etc., require the knowledge of a node's location, before invoking their respective algorithms. Locating a node is a challenging problem in large-scale ad hoc networks, especially when the nodes are mobile. Initial approaches to learn a node's location relied on flooding a query in the network. In view of the large overheads of flooding, *gossiping* was later proposed as a scalable alternative. With gossiping, each node probabilistically forwards the query. Research findings show that for carefully chosen gossip probabilities, the query is very likely to reach the destination.**

While gossiping is a random propagation of a specific information, a similar effect might be achieved if nodes move randomly carrying the same information in their local caches. Thus, a node's location information can diffuse into the network via node mobility, as opposed to wireless transmissions that incur bandwidth. If these mobile nodes lend their caches to be queried by its neighborhood, a distributed location estimation service can be envisioned.

Clearly, wireless transmission and physical mobility appear to be two modes of information transportation. While each have been studied individually, there has been little work on the possibility of combining them. For example, mobile nodes that gossip with each other periodically, may achieve percolation at significantly lower overheads. This paper investigates the combined potential of gossip and random (Brownian) node mobility. We argue that using such a combined strategy, it might be possible to design a scalable location service for mobile networks, over which several other applications can be developed, including geographic routing, resource discovery, etc.

I. INTRODUCTION

Location discovery is often a key service primitive in distributed systems. Several distributed system applications are built on such a location service, wherein, the location of the desired entity is determined before communication can be initiated toward it. While certain applications can query central servers for an entity's location (e.g., location of a restaurant), other applications exist where a central server is either infeasible or undesirable. For example, one may envision communication among mobile volunteers at disaster-relief operations, among soldiers in a battlefield operation, or among passengers over a vehicular network. To be more futuristic, one may envision mobile sensors communicating among themselves over a wide-spread area. In all such applications, querying centralized servers lead to serious problems of scalability and reliability. In fact, even distributed solutions like flooding prove to be inefficient [1], in view of its high-overhead, and energy-inefficient characteristics.

1-4244-0030-9/05/$20.00

As an alternative to flooding, *gossiping* has been shown to be an efficient approach for information dissemination. As opposed to flooding, where all nodes forward every packet it receives, gossiping requires nodes to forward packets with a certain probability, p_{gossip}. It has been shown that when p_{gossip} is suitably chosen, a query can reach all nodes in the network with very high probability. When the destination node receives this query, it replies back with an "*I-am-at-(x,y)-location*" message, which is routed back to the query-originator. On an average, since network nodes do not transmit all the received query-packets, the total overhead associated to gossiping is lower.

While lower, the overhead associated to gossiping is still non-marginal, and more importantly, unnecessary. For example, a gossiped query spreads far and wide over the network, even if the destination node is located close to the query-originator. On similar lines, when the destination is located north of the query-originator, the gossip unnecessarily propagates toward south. In systems that require short message exchanges between the originator and the destination, such unnecessary overheads can prove to be relatively significant, deeming the solution impractical. A solution is necessary that prunes query-propagation in unproductive directions, while allowing it to converge toward the approximate direction of the destination. In other words, the query needs to obtain an *hint* of where the destination could potentially be.

While seeking for a *hint*, we noticed the possibility of mobile nodes acting as information carriers. *We observed that while nodes move around in the network, they encounter several other nodes in the passing, and the memory of such encounters can potentially act as hints to other nodes.* For instance, if node X receives a query for destination node D, and if node X had encountered D at location (x, y), t_D time units in the past, then node X can (geographically) forward the packet toward coordinates (x, y). The value of t_D can be included in the query packet as a measure of confidence. Downstream nodes that receive this query packet, continue to forward it toward location (x, y), unless they had encountered D less than t_D time units in the past, in which case, they redirect the query toward their memorized location. Clearly, several issues arise – *Can multiple queries redundantly reach the destination over diverse paths? Are the paths traversed by queries significantly longer? Will querying incur high latency?* etc. This paper aims to address these issues, and design a protocol that offers a distributed, scalable, location service for wireless mobile networks.

II. RELATED WORK

With the emergence of mobile wireless networks, including cellular and ad hoc networks, the problem of location estimation gained prominence. While the problem was not easy in cellular networks, the possibility of having a wired backbone (for cellular backhaul) allowed much of the control signals to be transported over it [2]. However, in the absence of such a wired infrastructure, ad hoc networks are required to support location management over the wireless medium. Geographic routing protocols such as LAR, GPSR, DREAM, [3],[4],[5], assumed that the location of the destination is either known, or could be found out through a network-wide flood. Once the location is available to the source node, it can forward the packet to its neighbor who is located geographically closer to the destination's location. If each node forwards the packet similarly, the packet will eventually reach the destination. While such protocols appeared efficient, their scalability was limited by the initial flooding mechanism. The serious impact of flooding was revealed in [1], exposing the bottleneck of geographic routing protocols.

To alleviate the overheads of flooding, a gossip-based approach was proposed in [6]. While gossiping had been studied earlier in the context of distributed systems [7], the authors in [6] were among the first to exploit it for route-discovery in mobile networks. Several papers further exploited the gossiping technique, and extracted reasonable benefits in message complexity [8].

In this paper, we identify the scope to outperform gossip-based techniques, in the context of mobile networks. While [9],[10] describe message ferrying algorithms to transport packets across networks, the combination of gossip and node mobility has not been investigated in the past. We show that physical mobility can partially emulate the impact of gossip, leading to the possibility of lesser communication overhead. Based on this, we propose a protocol where information is transported via wireless transmission as well as physical mobility. We present our ideas in the form of a location estimation service in wireless ad hoc networks.

III. BROWNIAN GOSSIP: AN OVERVIEW

We begin with a high level intuition of our approach. Then, we present our proposal and discuss several issues and extensions to our basic scheme.

A. The Intuition

Transporting information bits can be achieved in multiple ways. While wireless transmission could be one form, physical transportation of an information-storage device could be another. In mobile ad hoc networks, where wireless communication is expensive, physical mobility of nodes can be exploited to transport information spatially and temporally. In particular, if nodes A and B encountered each other at a particular location in the past[1], then they can transport this location information to remote parts of the network, and provide it to queries that need it. The diagrams in Figure 1 demonstrate this effect. Of course, with increasing space and elapsing time, each piece of information becomes increasingly more noisy. However, for services that benefit from slight hints, such noisy information may suffice.

An important question arises at this point. Since the mobility of nodes cannot be controlled, the behavior of information percolation will essentially be uncontrolled too. To illustrate this, consider a worst case scenario in which all nodes in a network decide to stop moving. In such a scenario, physical transportation of an information would cease, and the information diffusion would have to rely completely on wireless transmissions. This indicates that the blend of wireless communication and physical transportation depends on the mobility patterns of the nodes. Where nodes move faster, gossiping through wireless transmissions can be replaced by physical transporta-

---

[1] Encountering entails coming in each other's communication range, and thereby learning their mutual existence.

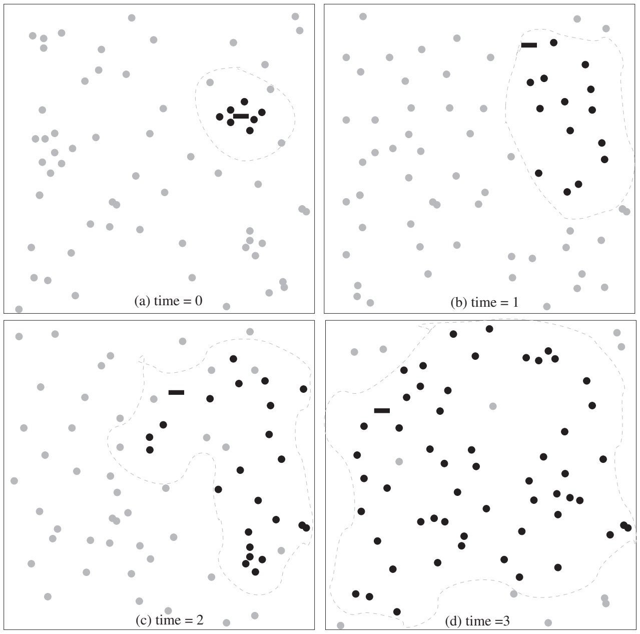

Fig. 1. We simulated a network in which the location of the rectangular node was initially known only to the black nodes (time 0). As the black nodes moved using random way-point mobility, they performed Brownian gossip with $p_{gossip} = 0.25$. Every node that came to learn of the rectangular node was colored black. The diffusion of information is shown for increasing time.

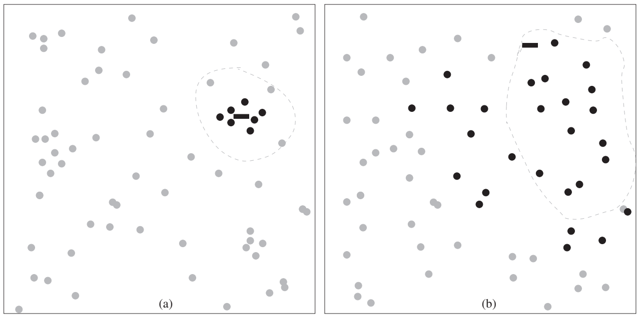

Fig. 2. With $p_{gossip} = 0.25$, regular gossip diffuses the information to a significantly smaller fraction of the nodes. Moreover, the information is not persistent with time, and thus unavailable for others to query.

tion. However, note that when nodes slow down, the overhead of wireless transmission does not increase heavily. This is because, as nodes move slower, their location becomes increasingly more predictable, and thus, a desired level of location estimation can be achieved with fewer gossip messages. In other words, we indicate the possibility of a self-sufficient network, that copes with high and low mobility, at reasonably low bandwidth overheads. Our proposal on Brownian gossip aims to realize this possibility. We are currently evaluating this scheme through exhaustive simulation.

B. Protocol Description

- Caching Encounters

We assume that as nodes move into the communication ranges of other nodes, beacons are exchanged between them[2]. The GPS location of the nodes are included in the beacons. As a result, nodes learn about their mutual locations, and update their local caches with this information – we call this operation an *encounter*. As an example, if nodes A and B encounter each other at time t_i, while they are at locations (x_A, y_A) and (x_B, y_B) respectively, then node A updates its cache with the following tuple.

$$\{node\ B,\ t_i,\ (x_B, y_B),\ IsNeighbor\ TRUE\}$$

Node B updates its own cache similarly. While nodes A and B remain in each other's communication range, they update the same entry with newer (more recent) timestamps and locations. This update is performed periodically, based on the frequency of beacon exchanges. The updation terminates once the nodes have moved out of their mutual communication ranges.

Once A and B are not immediate neighbors, they set $IsNeighbor$ to $FALSE$, indicating that the encounter occurred in the past. The tuple in node A's cache is now of the form

$$\{node\ B,\ t_j,\ (x_B^{'}, y_B^{'}),\ IsNeighbor\ FALSE\}$$

where $t_j \geq t_i$. Now, as these nodes encounter new nodes in the network, they mutually gossip their memory of encounters. For example, if node A meets node P at a later point of time, t_k, then node A informs P about its memory of encounter with node B. If P has not met B any later than t_j, then P updates its own cache with the memory of A's encounter with B[3]. Thereafter, P pretends that P met B at time t_j in the past, and continues to gossip this information with subsequently encountered nodes. The choice of gossip probability, p_{gossip}, determines the overhead from wireless transmissions. Moreover, the value can be adapted based on the mobility pattern of nodes, and the traffic pattern in the network. We discuss the choice of p_{gossip} later in this section.

- Querying the Destination's Location

A source node, S, that intends to determine the location of a destination node, D, initiates queries directed toward K chosen directions. To be precise, node S first chooses a

---

[2] Several protocols require such beacon exchanges, including neighbor discovery protocols, time synchronization protocols, etc.

[3] However, if P had happened to meet B later than t_j, then node A updates its own cache with the memory of P's encounter with B.

random location in the network, say (x_{rand}, y_{rand}). Now based on this location, it chooses $K-1$ other locations, such that the lines joining node S with each of these locations are angularly separated by $\frac{2\pi}{K}$ radians. Now, S directs K copies of the query packet, one in each direction. The diagram in Figure 3(a) illustrates the idea with $K = 3$, i.e., 3 copies of the packet radially propagate in 3 diverse directions.

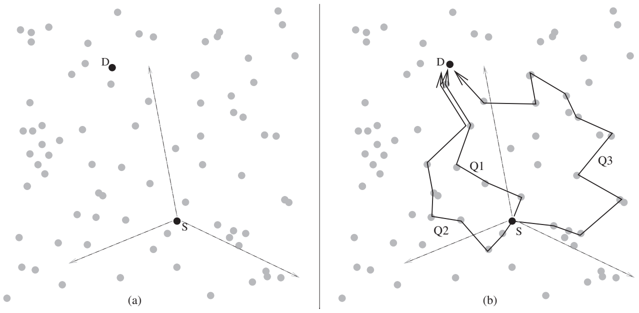

Fig. 3. (a)Choosing 3 directions for geographically forwarding the same query. (b) Tracing the path of 3 queries while they converge toward destination D.

As the query (geographically) propagates in the specified direction, it inquires nodes about their encounters with D. A node that has a memory of its encounter with D, say from time t_m, guides the query toward its cached location. The value of t_m is included into the query, so that nodes that receive this query, do not redirect the packet unless they have encountered D later than t_m. Thus, at this point, the query packet has the following form.

$$\{Source\, S,\ Location_S\,(x_S, y_S),\ Target\, D,\ t_m,\ Hops\, h\}$$

As the query closes in toward the destination, it gains more and more accurate information about the destination's location. This is because of spatial locality – nodes that are located near D, are likely to have encountered D more recently, and thus may offer better estimates of the destination's location. Figure 3 (b) illustrates the convergence of multiple query packets to the destination, D. Once the query reaches node D, node D replies to the source node S with a reply packet that contains its present location, (x_D, y_D). This reply packet can then be geographically routed back to S, based on the location of S included in the query.

IV. PROTOCOL DETAILS

A tradeoff arises as a consequence of sending K copies of the same query. Observe that all the K copies may eventually reach the destination, at different points in time. While this is redundant, notice that a single query might take significantly longer to reach the destination. One way to handle this *overhead-latency tradeoff* could be through more informed choice of K. For example, a source node S, that already has a hint of D's location, can choose a smaller value of K, depending on how stale the hint is. One way of choosing this value of K can be

$$K = min(\frac{(t_{current} - t_{last-encounter})}{C}, K_{max})$$

where C and K_{max} are appropriately chosen constants. Thus, a node that knows of a recent encounter with D, spawns very few copies of the query to propagate into the network. Moreover, these queries are not transmitted in random directions, but are guided to envelope the expected location of D. The expected location of D is the region over which D is likely to be, given that D was encountered at a location (x_D, y_D), at $t_{last-encounter}$. The expected region can be calculated as a circular region centered at (x_D, y_D), and with a radius of $v_{max}(t_{current} - t_{last-encounter})$, where v_{max} is the maximum speed with which a node might move. Guiding queries toward the expected region can significantly reduce the search space for the query.

Another design choice with Brownian gossip relates to the frequency with which a mobile node must gossip. Clearly, a node that has its neighbor-set changing continuously (i.e., more mobile), must gossip more frequently, as opposed to another node, whose neighbor-set hardly changes with time (almost stationary). This might seem surprising because our proposal argues that mobility reduces the amount of gossip necessary in the network. We intend to observe a subtle point here. Observe that a mobile node that gossips, spares many other static nodes from gossiping, and thereby the aggregate overhead reduces. If the mobile nodes did not undertake the responsibility of diffusing the information, static nodes would need to gossip at substantially higher probabilities. Thus, our observation that fast-moving nodes are required to gossip more, is not in conflict with the key intuition of this paper.

We now discuss one way of choosing the interval at which a node must gossip. A suitable value of gossip-interval is one that separates two consecutive gossip transmissions in time, such that these two transmissions are received by totally different sets of nodes. To achieve this, we choose the gossip-interval to be $\frac{2R}{v}$, where R is the communication range and v is the speed of the node that intends to gossip. Our justification is that, on an average, a node would substantially change its neighbor set every $\frac{2R}{v}$ time units.

While we have chosen the gossip-interval, we still need to choose the precise value of p_{gossip}. A tradeoff arises – while higher values of p_{gossip} leads to quicker percolation, it also increases the overhead. Moreover, an ideal value of p_{gossip} should probably be on a per-node basis. More precisely, a node that is almost stationary, or does not have too many queries destined to it, need not be gossiped about frequently, and vice versa. Our proposal handles this in the following manner. For every other node, i, node X maintains a value p^i_{gossip}. When node X is required to gossip every "$gossip - interval$" time units, it picks each of the nodes based on its corresponding p^i_{gossip}. Nodes that are selected are included in a gossip packet and sent out as a wireless transmission. If none of the nodes are selected at this time, node X does not gossip, and waits till the next scheduled time.

To determine p^i_{gossip}, we consider the mobility of node i, and the traffic destined to it. To achieve this, we require each node i to perform two book-keeping tasks. Each node maintains the number of times it was queried within a T time interval in the past – called $QueryCount_i$. Also, each node maintains the average speed with which it has traveled over the past T interval – called $AvgSpeed_i$. When node X encounters node i, it updates its cache with these values as well. Later, p^i_{gossip} is chosen as follows

$$p^i_{gossip} = max\{\ f(QueryCount_i),\ g(AvgSpeed_i)\}$$

where $f(.)$ and $g(.)$ are discrete, monotonically increasing functions, with range $[0, 1]$. The exact nature of $f(.)$ and $g(.)$ can be selected based on specific application requirements.

V. DISCUSSION

While we are currently evaluating the proposal through exhaustive simulation, we are also engaged in addressing some of the issues that arise from the protocol. Observe that nodes that are more mobile (and thereby gossip more often), spend more of their energy. For example, if a mobile node passes through a static region in the network, it diffuses the information regarding this region far and wide into the network. However, nodes in this static region gossip much less about this single mobile node. A heterogeneous distribution in energy consumption arises. We intend to address this issue as a part of our future work.

The mobility pattern of nodes will also influence the quality of diffusion, and in turn the gossip probabilities. Several challenging problems arise – given that nodes move in groups, or in some other correlated form, not all mobile nodes may need to gossip. Making distributed decisions on which nodes should gossip is a difficult problem to solve, and remains open. Further-more, other mobility models like *Manhattan Grid*, *Gauss-Markov*, etc. may influence the percolation behavior. We intend to work on these aspects as well, as part of our future work.

VI. CONCLUSION

Location service in mobile networks has been viewed as a challenging problem. To cope with the uncertainty of a node's constantly changing location, previous solutions have invested bandwidth in the form of flooding or gossiping a query. In this paper, we observe that physical mobility of a node, along with its storage caches, can be an alternative to wireless transmissions. We propose the possibility of caching a particular event, and gossiping the memory of that event with other nodes encountered in the network. We qualitatively argue that a large fraction of the wireless overhead can be alleviated by exploiting Brownian node mobility. We present a scheme in which mobile nodes gossip about the location of nodes that they have encountered in the past, and thereby forms a distributed, scalable, location service. Several applications, like geographic routing, resource discovery, etc. may significantly benefit from our proposed scheme of Brownian gossip. We are currently evaluating our proposal through exhaustive simulation.

REFERENCES

[1] Yu-Chee Tseng, Sze-Yao Ni, Yuh-Shyan Chen, and Jang-Ping Sheu, The broadcast storm problem in a mobile ad hoc network, *Wireless Networking*, 2002.

[2] Jingyuan Zhang, Location management in cellular networks, 2002.

[3] Y. Ko and N. H. Vaidya, Location aided routing (lar) in mobile ad hoc networks, in *Mobile Computing and Networking*, 1998.

[4] Brad Karp and H. T. Kung, Gpsr: greedy perimeter stateless routing for wireless networks, in *Proceedings of Mobicom*, 2000.

[5] Stefano Basagni, Imrich Chlamtac, Violet R. Syrotiuk, and Barry A. Woodward, A distance routing effect algorithm for mobility (dream), in *Proceedings of Mobicom*, 1998.

[6] Zygmunt J. Haas, Joseph Y. Halpern, and Li Li, Gossip-based ad hoc routing, in *Infocom*, 2002.

[7] David Kempe, Alin Dobra, and Johannes Gehrke, Gossip-based computation of aggregate information, in *IEEE Symposium on Foundations of Computer Science (FOCS'03)*, 2003.

[8] Xiang-Yang Li, K. Moaveninejad, and O. Frieder, Regional gossip routing for wireless ad hoc networks, *Mobile Networks and Applications*, vol. 10, no. 1-2, 2005.

[9] W. Zhao, M. Ammar, and E. Zegura, A message ferrying approach for data delivery in sparse mobile ad hoc networks, in *Proceedings of Mobihoc*, 2004.

[10] M. Grossglauser and M. Vetterli, Locating nodes with ease, in *Proceedings of Infocom*, 2003.