On Interest Locality in Content-Based Routing for Large-scale MANETs

Yang Zhang, Jing Zhao, Guohong Cao and Chita R. Das
Department of Computer Science and Engineering
The Pennsylvania State University
E-mail: {yangzhan,jizhao,gcao,das}@cse.psu.edu

***Abstract*—To disseminate content with content-based routing (CBR), the routing paths of subscription and publication cannot be determined *a priori* and have to be computed hop-by-hop, which brings in scalability and robustness challenges in large scale mobile ad hoc networks (MANETs). In this paper, we propose a novel two-tier content-based routing protocol called CLONE (Community and Location aware cONtEnt based routing). In CLONE, we map the human community structure of social networks to MANETs. The whole network can be self-organized into communities based on the interest locality, so that most subscriptions inside a community can be served in an intra-community fashion, reducing the communication overhead and the response delay. Community construction is self-organized and completely distributed. Analytical and simulation results demonstrate the effectiveness of CLONE in large-scale MANETs.**

I. INTRODUCTION

Content-based routing (CBR) has emerged as a popular communication paradigm in which the data flow is driven by the content rather than the explicit address of the destination [4]. By allowing consumers to define the content they are interested in, the content providers and consumers are decoupled. Content providers can simply inject the content to the network without considering what consumers will use the content. Similarly, the consumers do not need to be aware of the sources of the content they receive. This feature makes CBR especially suitable for MANETs, where the source and destination may move.

Most early implementations of CBR are on the Internet to provide publish/subscribe services [4], [6]. They rely on dedicate servers and stationary infrastructures that act as *brokers* to track the forwarding information and maintain good network connection. These solutions cannot be directly applied to large-scale and dynamic ad hoc networks consisting of a large number of mobile nodes. For example, in CBR, the destination of each subscription/publication cannot be determined *a priori*, and must be computed hop-by-hop. To route the subscription/publication to all and only those destinations, each intermediate node has to keep track of all forwarding information about its neighbors. As the network scales, both the memory requirement and the content filtering delay become major problems. Meanwhile, as no explicit node address is given in routing, if one intermediate node is disconnected from the system, there is no way to find it. The subsequent nodes behind it cannot get the subscription/publication and thus the CBR cannot proceed. All these limitations bring in scalability and robustness challenges for CBR in large-scale mobile ad hoc networks.

Content-based routing is intrinsically data-centric [17] and human-centric [25], i.e., contents are routed based on consumers' specified interests while consumers' behaviors follow some kind of social features such as mobility, community, membership, locality, etc. In [15], McPherson *et al.* identified the *homophily theory* through hundreds of case studies to prove the community and homophily feature of human society. They found that people are usually not uniformly distributed. The human society is partitioned into geographic communities based on the population density distribution. They also found that content accessing has a certain degree of location/community dependency. Individuals in the same community are more likely to have interest similarity than people in different communities and people with similar interest are more likely to stay together in the same community. These behaviors can be easily found in our daily life. For example, residents living in nearby blocks may be interested in local events (e.g., local news, weather, activity) around them; reporters in the same stadium focus on the same game; vehicles in the same area want the same traffic information ahead; soldiers in the same platoon need to wait for the same commanding officer and are interested in any information on the same surrounding area. These results from human networks motivate us to study and exploit these social characteristics to optimize content-based routing. If communities can be formed based on content consumers' interest similarity, the performance of CBR can be significantly improved.

In this paper, we propose a novel two-tier content-based routing protocol called CLONE (Community and Location aware cONtEnt based routing) for large-scale mobile networks. In CLONE, the whole network is self-organized into several communities based on the interest locality, so that nearby nodes having common interest are clustered together. A *community principal* is self-selected to buffer and forward subscriptions and publish content between content consumers and providers. We propose novel solutions to construct communities and propose solutions to optimize the content publishing. Compared with the traditional CBR [18], CLONE has the following characteristics: i) distributed community construction based on interest locality, ii) two-tier and high efficient inter-community and intra-community based routing, and iii) robust salvation mechanism to deal with dynamic topology changes and disconnections.

The rest of this paper is organized as follows. Section II

978-1-4244-5113-5/09/$25.00 ©2009 IEEE

describes the design of CLONE such as how to use interest locality to construct communities, how the two-tier CBR works and how to optimize the content publishing. The analysis of the communication overhead and the response delay is presented in Section III. Section IV presents the simulation results and Section V discusses related work. Section VI concludes the paper.

II. CLONE:COMMUNITY AND LOCATION AWARE cONTENT BASED ROUTING

In the traditional CBR [17], [18], all content consumers passively receive publications. Only when a new publication comes, they are notified and served. Therefore, consumers suffer from long response delay if the content updating rate is low. If the consumers want to get the content immediately after they submit their subscriptions in an on-demand fashion, the network will be congested with publications and subscriptions. To address this problem, CLONE groups consumers with similar interests in a community so that nodes in the same community can be managed by a local *community principal* and all subscriptions for the same content can be aggregated. Then, the content delivery becomes a two-tier process. The community principal checks its local content availability after receiving a subscription. If the content is locally available, the subscription is served immediately; otherwise it aggregates the same subscriptions and subscribe to the content provider. During publishing, the content provider only publishes content to the interested community principal, who further delivers it to the consumers in the same community. Next, we present the techniques used to construct community and the techniques for the two-tier CBR.

A. Community Construction

The metrics used for community construction affect the performance of content delivery. If nodes in the same community are highly connected and share the common interests, more benefits can be obtained. It is impossible for nodes to obtain the interest information of the whole network since they can only know their neighbors' subscription information through beacon messages. Therefore, in CLONE, community is completely self-organized, i.e., both community construction and community principal selection need to be processed in a fully distributed way based on local interest similarity. In this section, we first present the basic community construction scheme and then propose some optimizations.

1) The Basic Community Construction Scheme: If two nodes have similar interests, there is a high probability that they will access similar contents in the future. Following this observation, we measure interest similarity between two nodes as the overlap of their interests. Formally, we have the following definitions:

Definition 1 (Similarity Coefficient): Given two nodes i and j, their interest sets are $S(i)$ and $S(j)$. The *Similarity Coefficient ($\mathcal{SC}$)* between i and j is defined as the number of common interests of i and j:

$$\mathcal{SC}(i,j) = \mathcal{SC}(j,i) = |S(i) \cap S(j)|$$

where $|x|$ indicates the cardinality of the set x (i.e., number of elements in set x).

Definition 2 (Community Principal Weight): Given one node i, its *Community Principal Weight ($\mathcal{CPW}$)* indicates the benefit for it to be the community principal. $\mathcal{CPW}$ of node i is defined as the sum of the $\mathcal{SC}$ of node i with its neighbors.

$$\mathcal{CPW}(i) = \sum_{j \in Neighbor(i)} \mathcal{SC}(i,j)$$

Clearly, if one node has a larger $\mathcal{CPW}$, it has higher interest similarity with its neighbors. Hence if this node can be selected as the community principal, more benefits can be achieved.

Fig. 1(a) gives a simple scenario, where $a \sim j$ represent 10 nodes and $C_1 \sim C_6$ represent different contents. Then, $\mathcal{SC}(a,e) = 3$. $\mathcal{CPW}(a) = \mathcal{SC}(a,e) + \mathcal{SC}(a,f) + \mathcal{SC}(a,g) + \mathcal{SC}(a,h) + \mathcal{SC}(a,b) = 3 + 4 + 3 + 3 + 5 = 18$. After nodes get their own $\mathcal{CPW}$, they send out voting messages to compete for being the community principal. A node with a larger $\mathcal{CPW}$ has higher priority to become the community principal. To save communicate overhead, if a node receives a voting message with higher priority, it will not vote anymore. Suppose the Voting Delay Time ($\mathcal{VDT}$) is the amount of time that each node should wait before sending out their voting messages. Then, a node with smaller $\mathcal{CPW}$ should send out its voting message with a longer delay, i.e., for node i, $\mathcal{VDT}(i) \propto \frac{1}{\mathcal{CPW}(i)}$.

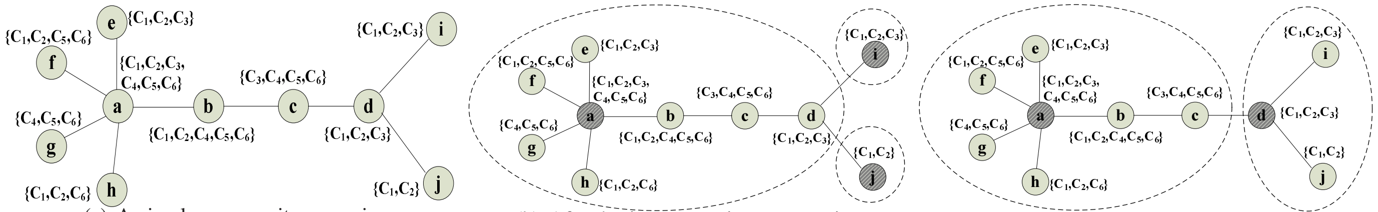
In CLONE, the community construction is performed as follows.

1) When one community construction tick comes[1], each node calculates its own $\mathcal{CPW}$ and locally sends out a voting message with its $\mathcal{CPW}$ after a delay $\mathcal{VDT}$, during which it may suppress its voting message if it has received another voting message with $\mathcal{CPW}$ larger than its own. A TTL is associated with the voting message to control the local flooding. It sets the maximum $\mathcal{CPW}$ to be its own $\mathcal{CPW}$.
2) After a node receives a voting message, it compares the received $\mathcal{CPW}$ with its own $\mathcal{CPW}$. If the newly received message has a larger $\mathcal{CPW}$, it updates its maximum $\mathcal{CPW}$ and set this node as its community principal. Then it decreases the TTL of the voting message by 1 and forwards it out. Otherwise, it drops this message.
3) Finally, the node with maximum $\mathcal{CPW}$ is selected as the community principal. It sends out confirmations to its members to finalize the community principal voting and finish the community construction.

2) Enhancements: The basic community construction approach can group nodes with similar interests together. However, there are many ways to improve the performance. For example, in the basic approach, the community radius is based on a fixed TTL value, and all nodes in the TTL range of the selected community principal will join the community. This may not always be efficient. As shown in Fig. 1(b),

---

[1] If all nodes are synchronized with the same clock time, community construction tick can be presented as some specific time; otherwise, some specific event can be used to trigger community construction. For example, in this paper, when one node receives the $10 * i$th advertisement of the content, it starts the community construction process.



(a) A simple community scenario (b) After basic community construction (c) After split and merge

Fig. 1. Community construction and enhancements

with TTL=3, neither i nor j can receive the voting message from a, and they have to construct two *orphan communities* (i.e., community with less than three members). The existence of such orphan community is not the original intention of community construction. To handle this problem, the following merge and split techniques are used. When a node finds that its community is an orphan community, it sends out a merge message to its one-hop neighbors. If a node at other orphan community receives this message (i.e., the two orphan communities are neighbors) or one node at the boundary of another community receives at least two merge messages (i.e., these orphan communities are connected by the node at the boundary of another community), these orphan communities can be merged. As shown in Fig. 1(c), when node d receives merge requests from its neighbors i and j, it splits from the community of a and forms a new community with i and j.

With community merge and split, the number of orphan communities can be reduced. It also relaxes the rule that all nodes in the community radius range need to join this community. In Fig. 1(c), when node d sets up a new community with i and j, node c can have two different choices: to join a's community or d's community. It can join a's community because a has more members and has more content. In the future, if c is interested in other content, it has a higher possibility to get its interested content from a directly. On the other hand, it is also reasonable for c to join d's community because c is closer to d than a. If c takes d as its community principal, the transmission delay and bandwidth consumption can be reduced.

Generally, suppose a node a has a choice to join community principal H_1 or H_2. The distance between a and H_1 is L_1 and the distance between a and H_2 is L_2. There exists R different types of content in the system, and the interest sets of H_1 and H_2 are $S(H_1)$ and $S(H_2)$. a accesses each content with a probability $Pr(i)$ ($1 \leq i \leq$ R). A binary function B(i, X) is defined to indicate whether i is a member of set X, i.e.,

$$B(i,X) = \begin{cases} 1 & i \in X; \\ 0 & otherwise \end{cases}$$

In a large-scale CBR system, we use L (the radius of the whole network) to estimate the average distance from the content provider to the community principal (H_1 and H_2). Then we can obtain the expected number of hops ($E(H)$) for a publication to be delivered upon each subscription request,

$$E(H_1) \approx \sum_{i=1}^{R} Pr(i) \cdot B(i, S(H_1)) \cdot L_1 + (1 - \sum_{i=1}^{R} Pr(i) \cdot B(i, S(H_1))) \cdot L$$

$$E(H_2) \approx \sum_{i=1}^{R} Pr(i) \cdot B(i, S(H_2)) \cdot L_2 + (1 - \sum_{i=1}^{R} Pr(i) \cdot B(i, S(H_2))) \cdot L$$

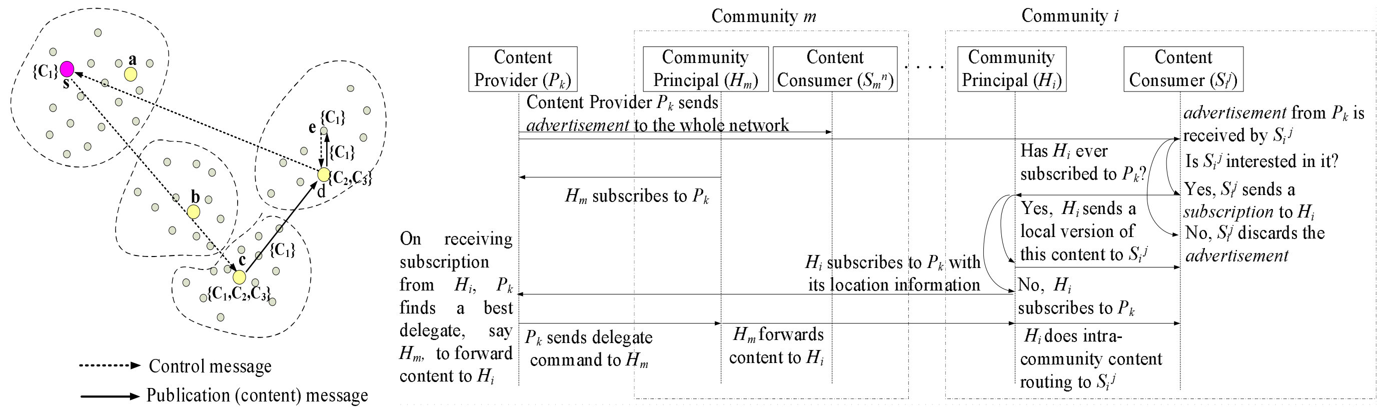
If $E(H_1) < E(H_2)$, a joins H_1's community; otherwise it joins H_2's community. This technique can also be extended to situations where several community principals compete.

B. The Two-Tier Content-based Routing

In CLONE, subscriptions and publications traverse two tiers to reach the destinations. Since there is no node address or identification in content-based routing, we need to decide how the content provider publishes the content to the interested communities and how the community information and subscriptions be routed to the content provider in an efficient and reliable way. In CLONE, geographical location information is used to assist routing. More specifically, the content provider inserts its location information into advertisements. Community principals use location information to route its subscriptions back with greedy geographical forwarding [3], [14]. At the same time, community's location information is also inserted into subscriptions. During the content publishing phase, the content is routed towards the community head. With geographical routing, messages can be delivered to the destinations quickly even if intermediate topology changes. Next, we look into the details of the three operations widely used in pub/sub services: (1) routing of advertisements, (2) routing of subscriptions and (3) routing of publications (content).

TABLE I
CLONE DATA STRUCTURES: ADSTABLE AND SUBSTABLE.

AdsTable	Store ads from neighbors
t.ad *	The content abstraction of ad
t.pbr *	The content provider
t.pbl *	The estimated location of provider
t.in *	Node that sent the advertisement
t.out *	Node that forwarded the subscription
t.expire	Time to expire this entry
...	
SubsTable	Store subs from neighbors
t.ad *	The content abstraction of ad
t.subl *	The estimated location of content consumer
t.in *	Node that sent the subscription
t.out *	Node that forwarded the advertisement
t.expire	Time to expire this entry
...	



(a) An example

(b) The protocol

Fig. 3. The delegate-assisted publishing

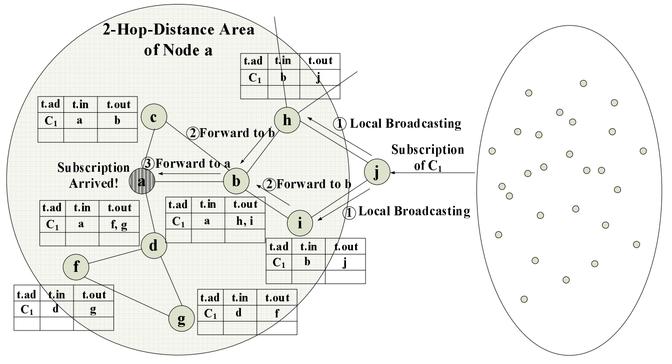


Fig. 2. Forwarding in the two-hop-distance area

1) Routing of Advertisement: Advertisements are used by content providers to announce the contents they publish. In the traditional CBR, each node maintains two data structures called *AdsTable* and *SubsTable* [12] to store advertisements from neighbors and create routing paths for the content consumers to route their subscriptions back to the providers. In CLONE, the advertisement routing is similar except that it uses geographical location information to assist routing. Therefore, intermediate nodes do not need to maintain the routing information. However, it is possible that the message arrives at its destination, but the expected destination is not there due to mobility. At this time, AdsTable can help route the message to one specific node. The fields in the data structure are outlined in Table I. In CLONE, only the nodes within two-hop distance area of the content provider or the community principal need to record this extra routing information in their AdsTable. Fig. 2 shows how this design saves communication overhead. In this example, for the content provider a, only nodes in its two-hop-distance area keep the AdsTable for it. In subscribing, when one subscription is routed to a node whose distance to the point where the content provider is expected to be less than its communication radius, it is broadcast locally. Therefore, one subscription for content C_1 is routed to node j where a is supposed to be at. However node a is out of its communication range, so it broadcasts the subscription to its neighbors. Then, node h and i who are in the two-hop-distance area of a receive the subscriptions. After finding the C_1 entry in their AdsTable, they know that the content provider of C_1 (node a) is close to them. Then the subscription can be forwarded to node b according to the AdsTable and finally it arrives at node a through b.

2) Routing of Subscriptions: In CLONE, the community principal forwards subscriptions for its community members. Because location information can be obtained from advertisements, greedy geographical forwarding is used for inter-community routing of subscriptions. Also, the location information of the community principal is added to the subscriptions so that it can be used by the content provider for routing the content back. The intra-community routing of subscriptions is simple. When one node wants to subscribe for one content, it subscribes to the community principal with the same subscribing procedure used in the traditional CBR.

3) Routing of Publications: The content provider needs to publish the new content after it is generated. Based on the received subscriptions, the content provider knows who should receive the content. As a result, it can send the content to all interested community principals directly with separate streams. Although this solution has low delay, it creates much redundant traffic. An alternative solution is to use the multicast-based solution, where the location information of the community principal can be used to build the multicast tree.

Besides multicast-based publishing, a delegate-assisted publishing mechanism is used to serve new content consumers quickly. It is possible that one member of a community subscribes some content that the provider has already distributed. In the traditional CBR, the consumer has to wait until the next content distribution time. With delegate-assisted publishing, when the content provider receives this subscription from a community principal a, it chooses one best community principal from its service list as the publishing delegate. This delegate should be close to a and has already received the latest content. The content provider sends one delegation command to the delegate which forwards the content to the subscribing community principal a. For example (see Fig. 3(a)), node e subscribes for content C_1, but its community principal d only has content C_2 and C_3. Node d subscribes to the provider of C_1, which is s. Node s checks the location of d and finds that c has a local copy of C_1 and c is closer to d. Then, s sends its delegation to c which forwards the publication to d. Since the delegation message is much smaller than the publication, it reduces the message overhead. At the same time, d can get the content quickly instead of waiting for the next publishing time. The procedure of delegate-assisted forwarding is described in Fig. 3(b).

C. Fault-Tolerance and Mobility Issues

In an MANET, nodes may move and thus leave or join the network. In this subsection, we discuss the mobility effects of the content provider, the consumer, and the community principal.

From the content provider point of view, the mobility of the provider does not have any negative effect on CLONE. Since all publications are initiated by the content provider who records the location information of each community, even if the provider moves, it can still route its publication to the interested communities. From the content consumer and community principal point of view, if the content provider moves, the subscriptions can still reach the destination as long as the content provider stays within a two-hop area (as discussed in Section II-B1).

Content consumers may disconnect or move away from the network. Their movements affect the interest pattern of the community and hence may change the community topology and community principal. Therefore, each community needs to update its topology and select its community principal periodically.

Since the community principal has the best connectivity and interest similarity with its surrounding nodes, it should have less mobility or have similar mobility with the rest of its community members. However, if the community principal moves out of its community, the performance of this community will be affected, which can be addressed by selecting a new community principal. At the next community construction tick, a new community principal can be selected and the system returns to normal.

III. PERFORMANCE ANALYSIS

In this section, we compare CLONE with the traditional CBR [18] in which all subscriptions and publications are sent between content providers and consumers directly.

A. The Model and Notations

We consider a square field in which N nodes are uniformly distributed; thus there are approximately $\sqrt{N}$ nodes on each side. Among these N nodes, there are M content providers with an average content publishing interval of T_P. The arriving time interval of each subscription is T_S. Each advertisement and subscription has a message size of S_A and S_S respectively. The publication packet size is S_P.

In CLONE, k communities are used. Each community has an average of $n = \frac{N}{k}$ nodes and $\sqrt{\frac{N}{k}}$ nodes on each side of the community. S_C denotes the message overhead for community construction and h denotes the radius of the community.

B. The Communication Overhead

We first analyze the communication overhead, which is the total number of hops that the control messages (such as voting message, advertisement and subscription) and content travel. In CLONE, the communication overhead consists of four parts:

- Overhead of *community construction*: Since voting messages are flooded locally (a $h * h$ square), the overhead of voting messages for each community construction is $4h^2 \cdot S_C$. The same overhead is applied for the confirmation messages. Given the community construction interval T_C, the overhead for community construction is
$$\frac{2k}{T_C}(4h^2 \cdot S_C) \tag{1}$$
- Overhead of *advertising*: In the advertising phase, advertisement is flooded to the whole network. Its overhead is
$$\frac{1}{T_A} \cdot N \cdot S_A \tag{2}$$
- Overhead of *subscriptions*: For inter-community subscribing, each community principal interested in the advertisement of some content will subscribe to the content provider. Then, the average overhead of inter-community subscribing is $\frac{k}{2} \cdot \frac{\sqrt{N}}{2} \cdot S_S$, where $\frac{\sqrt{N}}{2}$ is the average hops from the community principal to the content provider. Similarly, for each intra-community subscribing, the overhead is $\frac{N}{2k} \cdot \frac{1}{2} \cdot \sqrt{\frac{N}{k}} \cdot S_S$. The total overhead of subscription is
$$\frac{\alpha}{T_S} \cdot (\frac{k}{2} \cdot \frac{\sqrt{N}}{2} \cdot S_S) + \frac{1}{T_S} \cdot (\frac{N}{2k} \cdot \frac{1}{2} \cdot \sqrt{\frac{N}{k}} \cdot S_S) \tag{3}$$
where α is the coefficient to represent the portion of subscriptions that cannot be serviced locally.
- Overhead of *routing publications*: Similar to (3), the overhead of publishing is
$$(\frac{1}{T_P} + (1-\alpha) \cdot \frac{1}{T_S}) \cdot (\frac{k}{2} \cdot \frac{\sqrt{N}}{2} \cdot S_P) + \frac{1}{T_S} \cdot (\frac{N}{2k} \cdot \frac{1}{2} \cdot \sqrt{\frac{N}{k}} \cdot S_P) \tag{4}$$

The total communication overhead of CLONE (O_{CLONE}) is the sum of Expression (1) (2) (3) and (4).

In the traditional CBR, the communication overhead (O_{Trad}) is the sum of the overhead in advertising, subscribing and publishing. It can be calculated as
$$\frac{1}{T_A} \cdot N \cdot S_A + \frac{1}{T_P} \cdot \frac{N}{2} \cdot \frac{\sqrt{N}}{2} \cdot S_P + \frac{1}{T_S} \cdot \frac{N}{2} \cdot \frac{\sqrt{N}}{2} \cdot S_S \tag{5}$$

Comparing CLONE to the traditional CBR, we have:
$$(I): \frac{O_{CLONE}}{O_{Trad}} \approx \frac{\frac{1}{T_S} \cdot \frac{S_S}{k\sqrt{k}} + \frac{1}{T_S} \cdot \frac{S_P}{k\sqrt{k}}}{\frac{1}{T_P} \cdot S_P + \frac{1}{T_P} \cdot S_S}, \quad if\ N \gg k$$

$$(II): \frac{O_{CLONE}}{O_{Trad}} \approx \frac{(\frac{1}{T_P} + (1-\alpha) \cdot \frac{1}{T_S}) \cdot k + \frac{1}{T_S} \cdot \frac{N}{k\sqrt{k}}}{\frac{1}{T_P} \cdot N}, \quad if\ S_P \gg S_S, S_A, and\ S_C$$

Since the message size of subscription is significantly smaller than that of publication and the number of nodes is significantly larger than the number of communities, both expression (I) and (II) hold. Then combine (I) and (II) we get

$$(III): \qquad \frac{O_{CLONE}}{O_{Trad}} \approx \frac{T_P}{k\sqrt{k} \cdot T_S}, \\ if N \gg k \ \& \ S_P \gg S_S, S_A, and\ S_C$$

Expression (III) indicates that in a large-scale network ($N \gg k$), CLONE has less communication overhead than the traditional CBR as long as $k > (\frac{T_P}{T_S})^{\frac{2}{3}}$. For example, as used in the simulation, $T_P = 10sec$ and $T_S = 7sec$, when $k > \lceil \frac{T_P}{T_S}^{\frac{2}{3}} \rceil = \lceil \frac{10}{7}^{\frac{2}{3}} \rceil = 2$, $O_{CLONE} < O_{Trad}$, which means CLONE is more efficient than the traditional CBR as long as there are more than two communities in the network. When nearby nodes that share common interests are grouped into communities, more subscriptions can be served by the local community principal which saves the overhead to route the subscriptions and publications between the content consumers and providers. Expression (III) also shows that, given T_P and T_S, $\frac{O_{CLONE}}{O_{Trad}} \propto \frac{1}{k\sqrt{k}}$. Thus, when k increases, the benefit of CLONE over the traditional CBR increases. Expression (II) illustrates the communication overhead in two extreme cases. First, when $k \to N$, every node becomes a community and $\alpha \to 1$. CLONE becomes the traditional CBR. Therefore, $O_{CLONE} \approx O_{Trad}$. Second, when $k = 1$, $\frac{O_{CLONE}}{O_{Trad}} \approx \frac{T_P}{T_S}$. In this case, the whole network becomes one single community, CLONE is like the on-demanded traditional CBR.

C. The Response Delay

In this subsection, we compare the response delay of CLONE and the traditional CBR. The response delay is defined as the average delay from initiating the subscription to receiving the publication. In CLONE, if the local community principal has the requested content, it can send the content to the interested nodes immediately; i.e., the response delay is the transmission delay of the content from the community principal to the content consumer. We denote such delay as D_{hit}. If the local community principal does not have the content, the community principal needs to forward the subscription to the content provider and wait for the next publishing. Then the response delay D_{miss}, equals to $D_{hit} + D_{pub} + \frac{1}{2} \cdot T_P$, where D_{pub} is the delay of content based routing from the content provider to the community principal. With the delegate-assisted publishing, when the content provider receives a subscription from one community principal, it appoints one community head with its latest publication as its delegate to forward the content. With fewer hops of data transmission, the community principal can get the publication more quickly and the response delay can be further reduced. Therefore, with delegate-assisted publishing,

$$\begin{aligned} D_{CLONE} &= p \cdot D_{hit} + (1-p) \cdot D_{miss} \\ &= p \cdot D_{hit} + (1-p) \cdot (D_{hit} + \beta \cdot D_{pub}) \end{aligned}$$

where $\beta (0 < \beta < 1)$ is the improving coefficient from delegate-assisted publishing.

In the traditional CBR, the response delay is the average waiting time for publishing.

$$D_{Trad} = \frac{T_P}{2}$$

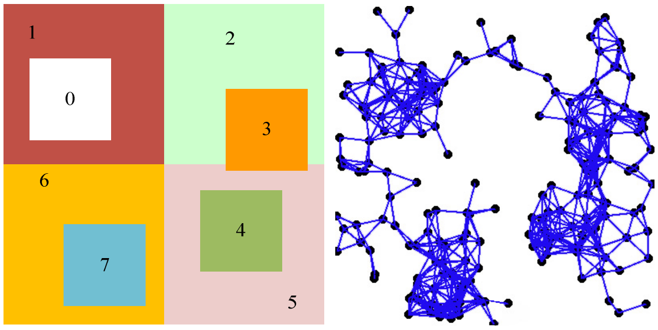
Since T_P is much larger than D_{pub}, D_{CLONE} is much shorter than D_{Trad}, i.e.,

$$D_{CLONE} \ll D_{Trad}$$

From the analysis of this section, we can easily see that CLONE outperforms the traditional CBR in (1) it significantly saves the communication overhead and (2) it greatly shortens the response delay.

IV. PERFORMANCE EVALUATIONS

In this section, we evaluate the performance of CLONE through simulations. We first compare CLONE with the traditional CBR [18] where content providers and consumers communicate directly, and the *grid-based CBR*. In the grid-based CBR, the network is divided into grids of nodes, and each grid has a grid head. Similar to CLONE, the content provider only publishes the content to the interested grid head, which delivers the content to the interested nodes in its grid. However, grids are fixed based on location. They are different from communities in CLONE which group nodes based on interest locality. Grid-based CBR does not exist in the literature. We use it only for comparison purpose to see the benefit of two-tier community aware content-based routing in CLONE.



(a) Area partition (b) Node deployment

Fig. 4. Simulation Scenario

A. The Simulation Setup and Performance Metrics

The simulation is based on *ns*-2 with the CMU wireless extension [10]. GPSR [14] is used as the routing protocol.

The node deployment model: Due to privacy and cost issues, there is no real-world pub/sub traces available for the public. In this paper, we try to set up a scenario that can capture the general characteristics of node mobility and content accessing in a real city-wide pub/sub system. The experiment is conducted in a 2000m*2000m square area. To model the node distribution in a city, we randomly select 4 deployment points in this simulation area. Each deployment point is the center of a 500m*500m square, which is the specific place of social interest. Here we call it *hub*. As shown in Fig. 4(a), the squares 0, 3, 4, 7 are four non-overlapping hubs. 25 nodes are randomly deployed in each hub and other 100 nodes are randomly distributed in the whole area. The topology is generated with the `setdest` tool in *ns*-2. Fig. 4(b) shows the topology of node deployment.

To see the performance of the content-based routing, 5 nodes are randomly selected as content providers and other nodes are consumers. Nodes move following the random way point movement model. To remove the negative effect of *speed decay* [26], we use the first 500 seconds simulation time as *warm-up*, which is long enough for the simulation to reach its steady state. After the warm-up, simulation results are recorded. Further, 100 nodes only move within their hubs at the speed of 2m/sec (walking) while the other 100 nodes move freely in the whole simulation area at the speed of 15m/sec (driving).

The content generating/consuming model: Each content provi-der generates four types of publications, and hence there are a total of 20 types of publications. Each content consumer generates a stream of subscriptions. The publication and subscription generating time follow exponential distribution with mean value of T_P and T_S respectively. Similar to [24], the content access pattern is based on the *Zipf* distribution. But to simulate the location-dependent property of content access pattern (i.e., nodes around the same location tend to access similar content), we make some changes to the original *Zipf* distribution. We divide the whole simulation area into 2 by 2 grids. Adding to the 4 deployment areas around the four deployment points, we get 8 non-overlap areas denoted as 0, 1, 2, ..., 7 (see Fig. 4(a)). Nodes in the same area follow the same *Zipf* pattern, while nodes in different areas have different offset values. For example, if the generated subscription should access publication id according to the original *Zipf* access pattern, in area i, the new id would be $(id + n \bmod i)$ mod n, where n is the total number of publication types. This subscription generating pattern ensures that nodes in the same area have similar, although not the same, content access pattern. Most of the simulation parameters and their default values are listed in Table II.

The evaluation metrics: We use three metrics to evaluate the performance of the protocols: the communication overhead, the response delay and the delivery ratio. Communication overhead and response delay have been defined in section III. The delivery ratio is the ratio of the number of successfully received publications to the total number of subscriptions at an interested node. Since the publication is much larger than the advertisement and subscription, we also measure the *normalized communication overhead*, which assumes the overhead to transmit one publication is γ times of other messages.

B. The Community Construction

We first study the community construction. We use the `ad-hockey` visualization tool [11] developed by Rice University to display the community construction results from the *ns-2* simulation trace. Fig. 5(a) shows a snapshot of the community construction result with the basic community construction approach. As can be seen, although nodes are grouped with their interest similarity, there are still some orphan communities, With the optimizations such as merge, split, and re-organization, those orphan communities are eliminated, as shown in Fig. 5(b).

TABLE II
SIMULATION PARAMETERS

Parameter	Default Value
Simulation Time	3500s
Warm-up Time	first 500s
Publication Type	20
Radio Range	200m
Moving Speed	2m/s (walking); 15m/s (driving)
Mean Subscription Generating Time (T_S)	7s
Mean Publication Generating Time (T_P)	10s
Mean Advertisement Generating Time (T_A)	50s
Community Radius (TTL)	3 hops
$Zipf$ Parameter θ	0.8
Normalization Coefficient for communication overhead γ	10

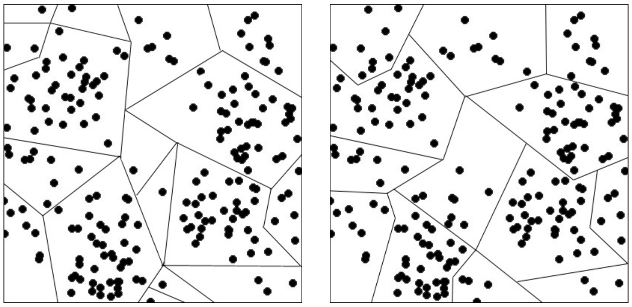
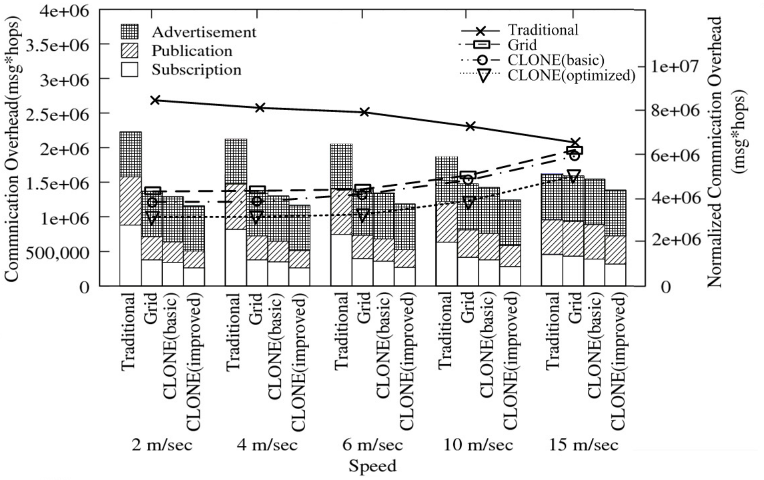

(a) Basic community construction with interest similarity
(b) The community construction with enhancement

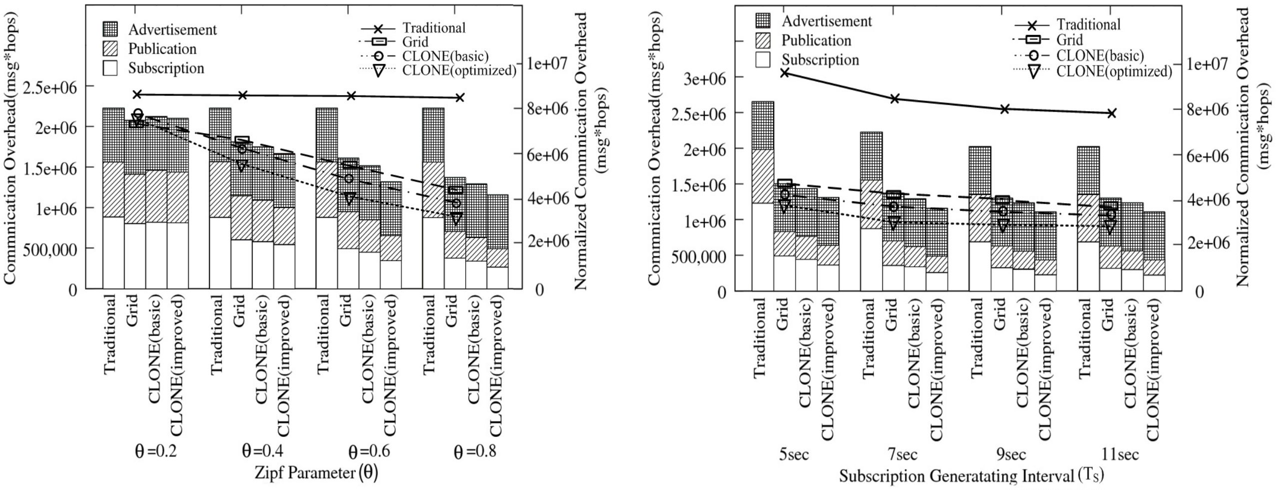
Fig. 5. Results of Community construction

C. The Communication Overhead

Fig. 6 evaluates the communication overhead of the traditional CBR, CLONE and the grid-based CBR, where the simulation area is evenly divided into $4 * 4 = 16$ grids. The bar-graph (Y axis on the left) shows the basic communication overhead and the line-graph (Y axis on the right) shows the normalized communication overhead. Fig. 6(a) compares the communication overhead at different moving speed of the nodes outside hubs. Mobility affects the stability of the network connection. Since the traditional CBR needs to delivery messages between content consumer and provider individually. It brings much more communication overhead. In the grid-based solution and CLONE, the hop-by-hop computation is only executed near the destination and messages in the same community can be aggregated, which can effectively avoid the negative effect of mobility and save communication cost. Similarly, even in a highly dynamic network, community construction based on interest locality and related optimization techniques are very effective. Here, we notice that in Fig. 6(a), as mobility increases, the total message cost of the traditional CBR decreases. This is not to say that the mobility helps save content delivery cost in the traditional CBR. The overhead decrease comes from the drop of delivery ratio (as show in Fig. 8, the delivery ratio is 99.4% when $2m/sec$ while 50.3% when $15m/sec$). Since more subscriptions cannot arrive their destinations, lots of publications are not issued



(a) Impact of mobility



(b) Impact of content access pattern

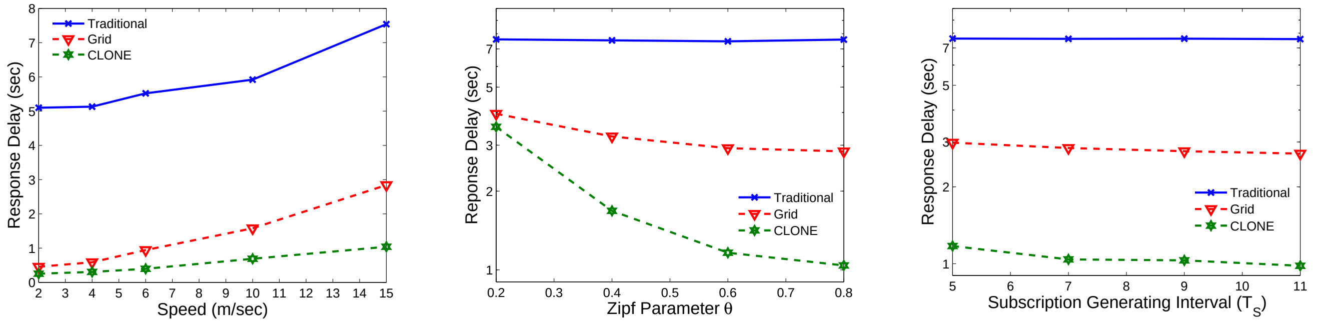
(c) Impact of subscription load

Fig. 6. The communication overhead

by the content providers. Therefore, in Fig. 6(b) and (c), we evaluate the communication overhead of different protocols at similar delivery ratio (i.e., all nodes moves in an average speed of $2m/sec$ so as all protocols can achieve a high delivery ratio of 99%), which can help gain a more accurate comparison of the real communication overhead for each content delivery with different protocols.

Fig. 6(b) illustrates the communication overhead with different interest patterns. Since the traditional CBR does not take sociological community into consideration, the change of interest pattern does not have much impact on its performance. However, the community-based solutions (grid and CLONE) can benefit from the skewness of interest pattern. As the figure shows, when the $Zipf$ parameter θ increases, the communication overhead of the grid-based solution and CLONE quickly reduces. This trend is more obvious in the normalized communication overhead analysis. This is because as θ increases, accessed content becomes more concentrated and hence more nodes can be served from their community principals. From the figure, we can also see that CLONE outperforms the grid-based approach due to considering the interest locality. Also, the optimized CLONE outperforms the basic CLONE by using merge, split and re-organization techniques and the delegate-assisted publishing. The normalized communication overhead has the same trend as the communication overhead, but the difference is a little bit higher. This is because the control overhead in CLONE is relatively smaller compared to the saved publication communication overhead, which is about ten times higher than the normal control packet.

Fig. 6(c) presents the communication overhead with different subscription load. When the subscription generating interval increases, fewer subscriptions are generated and thus the publication overhead decreases. Since community-based solutions can make use of the two-tier structure, their communication overhead is much less than that of the traditional CBR. Moreover, since CLONE takes the interest locality into account, the nodes in the same community have more interests similarity than that of the grid-based solution. and more subscriptions can be served locally. Therefore, CLONE saves more communication overhead compared with grid-based solution. In the following comparisons, CLONE is the optimized CLONE.



(a) Impact of mobility (b) Impact of access pattern (c) Impact of subscription load

Fig. 7. The response delay

D. The Response Delay

Fig. 7 compares the response delay. Since all interested nodes need to wait for the periodical publishing from content providers in the traditional CBR, they suffer from long response delay. However, in CLONE and the grid-based approach, subscriptions for the same content can be served with intra-community publishing, and hence reducing the response delay. Fig. 7 confirms the result.

As shown in Fig. 7(a), the grid-based solution and CLONE have better response delay even in a highly dynamic network. Since CLONE organizes communities based on interest locality instead of fixed grid, it can tolerate some network mobility.

Fig. 7(b) shows when the content access pattern becomes skewer, the response delay of the grid-based solution and CLONE reduce. When the content accessing is not very skew ($\theta = 0.2$), CLONE has similar response delay as the grid-based solution. As θ increases, the advantage of CLONE becomes more obvious. This is because CLONE organizes communities with interest locality and nearby delegates can be used to forward publications. However, these benefits can only be realized when the content access is skewed. Fig. 7(c) studies the response delay under different subscription load. Again, CLONE outperforms grid-based solution and the traditional CBR greatly (30% delay of grid-based solution and 13% delay of the traditional CBR).

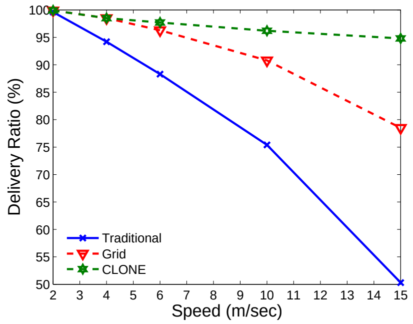


Fig. 8. The delivery ratio

E. The Delivery Ratio

The delivery ratio is studied in Fig. 8. We can see that as mobility increases, the delivery ratio of the traditional solution drops dramatically; i.e., the delivery ratio is cut by half as the moving speed increases from $2m/sec$ to $15m/sec$. This is because the hop-by-hop routing path computation limits its applicability in mobile networks. As network mobility increases, it becomes more difficult to find the next routing hop.

However, in the grid-based solution and CLONE, intra-community/grid publishing and location assisted content routing lead to a fast and relatively reliable delivery. Many subscriptions can be satisfied by local community/grid heads. Therefore, as mobility increases, their performance degradation is not as high as that in the traditional CBR. For similar reasons discussed earlier, CLONE outperforms the grid-based solutions.

V. RELATED WORK

Content distribution can be achieved by various means. The simplest approach is *flooding*, where all advertisements and contents are flooded into the whole network and unwanted contents are filtered out by the consumer. This approach can quickly lead to network congestion. An alternative approach is to use CBR in which content is only routed to interested consumers instead of all nodes in the network. Many pub/sub services rely on some form of CBR [4]. However, most existing works on CBR are studied in the context of wired and static networks, which assume that both publishers and subscribers do not move. Since these solutions do not consider the dynamic network topology changes and the limited wireless bandwidth, they may not be suitable for mobile networks.

Mobility is an important issue for CBR in MANETs. The works that are related to CLONE are the Mobile-ToPSS Project [17], [18] and the reconfiguration algorithm [1], [2]. They study the CBR fault tolerance and reliability issues in a dynamic environment. [19] also discusses the topology reconfiguration problem. However, these works can only be used in a small scale network as the hop-by-hop routing path computation limits the routing scalability and performance. Further, none of them considers interest locality, which is the major contribution of CLONE. CLONE also uses location information for fast and reliable message routing. All these improvements can reduce the computation overhead at intermediate nodes and mitigate the negative effect of node mobility. Other works such as [6] and [5] study the semantic and content representation issues in CBR, which are complement to CLONE and can be used by CLONE for intra-community routing.

Clustering [20], [27] has similar ideas to the community construction in CLONE, and hierarchical based routing has been widely used to improve the system scalability. Ye *et al.* [23] have designed a two-tier data dissemination scheme for wireless sensor networks, where the network is partitioned into grids based on node locations. Several clustering algorithms, like K-Means, G-Means, or hierarchical clustering [9] have been proposed for partitioning datasets based on some specific parameter. However, none of these techniques considers interest locality and human community behavior. CLONE targets at the interest similarity and locality. The nodes which are close and have high content access similarities are grouped in the same community. Also, in CLONE, the community construction process is completely distributed and self-organized. This makes CLONE more flexible, scalable and data-oriented.

Recently, researchers start to look into the social network and human behavior characteristics in computer networks and use them to optimize the network performance. These works focus on the mobile contact issues, such as pattern [7], [21], probing [22], data diffusion [28] and exchange [13], multicasting [8] and etc. Mohomed *et al.* [16] propose solutions to group users into communities and adapt content based on the usage semantics. However, it is still based on the "mobile-nodes, stationary broker" model, which is different from the peer-to-peer MANET model.

VI. CONCLUSIONS

Content-based routing (CBR) is considered to be very efficient and flexible for content distribution in mobile networks. One fundamental challenge for CBR in a large-scale mobile network comes from the fact that the routing paths cannot be determined *a priori* and have to be computed hop-by-hop. Existing CBRs are either based on wired and stationary networks with fixed infrastructure or only for small scale networks. In this paper, we proposed a novel two-tier content-based routing protocols called CLONE (Community and Location aware cONtEnt based routing) for large-scale mobile networks. In CLONE, we apply social network concepts to MANETs considering location and interests of the user. The whole network is self-organized into communities, so that nearby nodes having similar interests are grouped together. A community principal is self-selected to buffer and forward subscriptions and publications between content consumers and providers. Different from previous work, community location information is used to route content towards the community principal quickly and the hop-by-hop routing path computation and local broadcasting are only performed in areas close to the destination. Analytical results and simulation results demonstrate that CLONE can effectively route subscriptions and publications between content providers and consumers in large-scale MANETs.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation under grant CNS-0721479.

REFERENCES

[1] R. Baldoni, C. Marchetti, A. Virgillito, and R. Vitenberg. Content-based publish-subscribe over structured overlay networks. In *Proceedings of ICDCS'05*, pages 437–446, 2005.

[2] R. Baldoni1, R. Beraldi, L. Querzoni, and A. Virgillito. Efficient publish/subscribe through a self-organizing broker overlay and its application to siena. In *The Computer Journal*, pages 444–459, 2007.

[3] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *ACM Wireless Networks*, 2001.

[4] F. Cao and J.P. Singh. Efficient event routing in content-based publish-subscribe service networks. In *Proceedings of INFOCOM'04*, 2004.

[5] A. Carzaniga, M. Rutherford, and A. Wolf. A routing scheme for content-based networking. In *Proceedings of INFOCOM'04*, 2004.

[6] A. Carzaniga and A. Wolf. Forwarding in a content-based network. In *Proceedings of SIGCOMM'03*, pages 163–174, 2003.

[7] P. Costa, C. Mascolo, M. Musolesi, and G.P. Picco. Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, pages 748–760, 2008.

[8] W. Gao, Q. Li, B. Zhao, and G. Cao. Multicasting in delay tolerant networks: A social network perspective. In *Proc. MobiHoc*, 2009.

[9] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2002.

[10] http://http://www.isi.edu/nsnam/ns/. The network simulator.

[11] http://www.monarch.cs.rice.edu/cmu ns.html. Rice university.

[12] M. Petrovic I. Burcea H. Jacobsen. S-topss: Semantic toronto publish/subscribe system. In *Proceedings of VLDB'03*, 2003.

[13] T. Karagiannis, J. L. Boudec, and M. Vojnović. Power law and exponential decay of inter contact times between mobile devices. In *Proceedings of MOBICOM '07*, pages 183–194, 2007.

[14] Brad Karp and H. T. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. In *Proceedings of MOBICOM'00*, 2000.

[15] M. Mcpherson, L. Lovin, and J. Cook. Birds of a feather: Homophily in social networks. In *Annual Review of Sociology*, 2001.

[16] I. Mohomed, A. Chin, J. C. Cai, and E. de Lara. Community-driven adaptation: Automatic content adaptation in pervasive environments. In *Proceedings of the Sixth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '04)*, pages 124–133, 2004.

[17] Vinod Muthusamy, Milenko Petrovic, and Hans-Arno Jacobsen. Effects of routing computations in content-based routing networks with mobile data sources. In *Proceedings of MOBICOM '05*, pages 103–116, 2005.

[18] M. Petrovic, V. Muthusamy, and H. Jacobsen. Content-based routing in mobile ad hoc networks. In *Proceedings of the The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services(MOBIQUITOUS '05)*, pages 45–55, 2005.

[19] G. Picco, G. Cugola, and A. Murphy. Efficient content-based event dispatching in the presence of topological reconfiguration. In *Proceedings of ICDCS'03*, pages 234–243, 2003.

[20] L. Ramaswamy, B. Gedik, and L. Liu. A distributed approach to node clustering in decentralized peer-to-peer networks. *IEEE Trans. Parallel Distrib. Syst.*, 16(9):814–829, 2005.

[21] V. Srinivasan, M. Motani, and W. Ooi. Analysis and implications of student contact patterns derived from campus schedules. In *Proceedings of MOBICOM '06*, pages 86–97, 2006.

[22] W. Wang, V. Srinivasan, and M. Motani. Adaptive contact probing mechanisms for delay tolerant applications. In *Proceedings of MOBICOM '07*, pages 230–241, 2007.

[23] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang. A two-tier data dissemination model for large-scale wireless sensor networks. In *Proceedings of MOBICOM'02*, pages 148–159, 2002.

[24] L. Yin and G. Cao. Supporting cooperative caching in ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(1):77–89, 2006.

[25] E. Yoneki, P. Hui, S. Chan, and J. Crowcroft. A socio-aware overlay for publish/subscribe communication in delay tolerant networks. In *Proceedings of MSWiM'07*, 2007.

[26] J. Yoon, M. Liu, and B. Noble. Sound mobility models. In *Proceedings of MOBICOM '03*, pages 205–216, 2003.

[27] O. Younis and S. Fahmy. Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach. In *Proceedings of INFOCOM'04*, 2004.

[28] Y. Zhang and J. Zhao. Poster: Social network analysis on data diffusion in delay tolerant networks. In *Proc. MobiHoc*, 2009.