Community Aware Content Retrieval in Disruption-Tolerant Networks

You Lu, Mario Gerla
Department of Computer Science
University of California, Los Angeles
Los Angeles, USA
{youlu, gerla}@cs.ucla.edu

Tuan Le, Vince Rabsatt, Haik Kalantarian
Department of Computer Science
University of California, Los Angeles
Los Angeles, USA
{tuanle, rrabsatt, kalantarian}@cs.ucla.edu

Abstract—A major challenge in the design of a sparse mobile ad-hoc network (MANET) is the support of efficient content retrieval. Information-centric network (ICN) is an emerging network technology which provides a viable underlying architecture for efficient content distribution. However, current mobile ICN approaches generate excess control overhead that cause scalability issues as network and content sizes grow. In this paper, we propose a community-based content retrieval architecture which is highly scalable in disruption-tolerant mobile information-centric networks. Simulations in the NS-3 environment shows that our system requires less control overhead while maintaining comparable performance for content retrieval applications.

Keywords—social network forwarding; information-centric network; disruption-tolerant mobile ad-hoc network

I. INTRODUCTION

Mobile ad hoc networks (MANETs) are most effective in dynamic environments where network infrastructure is either not readily available or not adequate. Examples include coalition military operations, disaster recovery and emergency operations, and various other scenarios of vehicular communications. In many cases, MANETs will provide various services such as communication, storage, and computing for a range of applications. Sparse MANETs are a subclass of ad hoc networks in which the node population is sparse, and contact between nodes in the network are infrequent. As a result, message delivery in sparse MANET must be disruption-tolerant. One challenge in sparse MANET design is the ability to manage and serve the large amount of content resources distributed among different nodes. Thus, the sparse MANET architecture must support critical services, e.g., internal data storage, content search and sharing, etc.

In some cases, disruption-tolerant content delivery crosses different communication groups/communities. Each group may form their own connection/topology based on their social relationships. Multiple communities must be connected by a bridge node that exists in the border of its own community, and that is easily connected to other communities’ nodes. In this case, the efficient content retrieval among communities will be a great challenge, especially in a disruption-tolerant environment.

Regarding content search and retrieval services, information-centric network (ICN) has been drawing increased attention in both academia and industry. ICN is designed for content data search and retrieval, which is an alternative approach to the architecture of IP-based computer networks. In ICN, users only focus on the content data they are interested in, but do not need to know where these content data are stored or carried. Each content datum is identified by a unique name from the hierarchical naming scheme. The content retrieval follows the query-reply mode. A content consumer spreads his Interest packets through the network. When matching content is found either in the content provider or intermediate content cache server, the content data will trace its way back to the content consumer using the reverse route of the incoming Interest.

Several existing ICN proposals have been studied and implemented in Internet and MANET test beds. CCN [7] and NDN [20] are two popular examples of ICN designs in the Internet. Vehicle-NDN [19] and MANET-CCN [13] are two examples of the ICN architecture design in MANET, which can address the dynamic topology change for content retrieval. All of them are designed for the connected real-time network for the Internet or MANETs, not for the disruption-tolerant mobile ICN.

One major design challenge in disruption-tolerant mobile ICN is to design an efficient content retrieval scheme. In sparse MANETs, network connectivity is highly dynamic and the duration of the connection varies significantly. A common approach to deliver messages in a disruption-tolerant network is social network routing, which attempts to cover the gap of disconnection between nodes using the store-carry-and-forward method to deliver the message to the proper next hop until it reaches the destination. However, it cannot be deployed directly in mobile ICN, since the destination location, where the content is carried, is not exposed during the content search period.

In this paper, we propose a social-tie and community aware content retrieval scheme (STCRC: Social-Tie based Content Retrieval among Communities) in a disruption-tolerant mobile ICN, which addresses scalable content retrieval in large-scale sparse MANETs. STCRC allows users to request the content name toward the higher social level nodes, which are more popular in the network. If the Interest cannot be resolved in the requester’s community, it will be forwarded to other communities for the content query. After the Interest is matched by one of the content digests, the search process will forward the request to the next hop that has a higher possibility to reach the content provider based on the encounter history of the node. Finally, content data will be forwarded back from the content provider to the original requester.
The rest of this paper is organized as follows. Section II reviews the related work. Section III describes the design of the proposed scheme in detail. Section IV presents the results from performance experiments. Section V concludes the paper.

II. RELATED WORK

In this section, we introduce the general idea of information-centric network (ICN), and review the content retrieval methods in Internet and mobile ad hoc networks (MANETs). We also discuss disruption-tolerant networks in sparse MANET, and its routing schemes.

A. Information-Centric Network

Information-centric networking is an alternative architecture to IP-based computer networks. In ICN, the user focuses on their data of interest, rather than the physical location of the data. ICN differs from IP-based routing in three aspects. First, all contents are identified or named by the hierarchical naming scheme. A name becomes the object of routing or interest. Second, there is a designed caching system through the entire network that helps with the content distribution and provides the native features to help many applications, e.g., multicast. Third, the packet communication follows the form of query-reply mode. User (content consumer) spreads his interested content name as the “Interest” packet through the network. When one “Interest” packet hits the content name in an intermediate cache or the media server (content provider), the content data packets will be forwarded back to the content consumer along the reverse Interest forwarding path.

A number of previous studies focused on ICN with high level architectures, and provided sketches of the required components. Content-centric network (CCN) [7] and named data network (NDN) [20] are two implemented proposals for the ICN concept in the Internet. Their components including FIT, PIT, and Content Store form the caching and forwarding system for the content data in Internet applications. Several mobile ICN architectures have also been proposed for the mobile environment, e.g., Vehicle-NDN [19] for the traffic information dissemination, and MANET-CCN [13] for the tactical and emergency application.

There are clearly many potential challenges that remain to be appropriately analyzed and integrated into ICN architectures. One prominent example is the need for delay/disruption tolerance, a function that is increasingly important, considering the emerging dominance of easily disrupted mobile communications.

B. Disruption-Tolerant Network

A disruption-tolerant network (DTN) is a type of network that supports the existence of significant delays or disruptions between sending and receiving data [4]. Using the store-carry-and-forward method, DTN will temporarily store and carry the data during network disruptions until an appropriate next hop can be reached in a sparse MANET [16]. These disruptions and delays can be caused by a number of reasons such as low density of nodes, network failures, and wireless propagation limitations. One typical type that has received much research attention is the pocket switched network (PSN) [14]. PSNs are formed from opportunistic human contacts, typically by creating ad hoc links between mobile phones [15].

The routing protocol in sparse MANET has been discussed for decades, and different researchers have proposed many potential routing protocols. The observation is that the encounters between nodes in real environments do not occur randomly [2], and that nodes do not have an equal probability of encountering a set of nodes. Hsu et al claim that nodes never encountered more than 50 percent of the overall population [6]. As a consequence, not all nodes are equally likely to encounter each other, and nodes need to assess the probability that they will encounter the destination node. It was found that node encounters are sufficient to build a connected relationship graph, which is a small-world graph. Therefore, the social network routing is one of the most popular routing protocols in a disconnected delay-tolerant MANET. A node does not send messages to the next node randomly, but sends messages to a node they perceive might be a good carrier for messages based on their own local information [3].

Geographic routing exploits the fact that nodes usually reside in the plane. This enables the nodes to make local routing decisions based solely on the destinations’ geographic coordinates [12]. But geographic routing requires an efficient location service, i.e., a distributed database recording the location of every destination node. The transmission cost to maintain location state therefore depends directly on the amount of mobility, or the rate at which the network topology changes. Last encounter routing [5] is one example which utilizes position information to assist message delivery in disconnected delay-tolerant MANETs, but also attempts to reduce the cost of the location service. Each node maintains a local database of the time and location of its last encounter with every other node in the network. This database is consulted by packets to obtain estimates of their destination’s current locations. As a packet travels toward its destination, it is able to successively refine an estimate of the destination’s precise location, because node mobility has “diffused” estimates of that location.

Haggle [17] is another example which delivers the message among sparse MANET during encounters. It allows separating application logic from the underlying networking technology. Applications delegate the task of handling and communicating data to Haggle, which in turn adapts to the current network environment using the best available connectivity and protocol for the situation, and user-specified policies that allow trading speed, cost and power constraints. Haggle’s forwarding algorithm contains “direct” and “epidemic” modes. The “direct” mode runs in the connected network in order to deliver messages to the reachable destinations. The “epidemic” mode floods messages to all immediately reachable nodes in order to improve the delivery ratio in sparse MANETs.

According to the design of ICN applications, the content data is divided into several chunks in order to fit the congestion control and packet retransmission features. Chunks may be
required by interests separately, and mobile users may generate a huge amount of interest packets in a large-scale sparse MANET. All of these imply that the naive epidemic method is not suitable for the mobile ICN application in a large-scale network, especially between different communication groups/communities. We need a scalable content retrieval scheme to save the network control overhead, and enhance the delivery ratio in group communication for disruption-tolerant mobile ICN.

Our work differs from all these studies in several aspects. First, we consider the social hierarchy in different communities in the encounter relationship to direct packet forwarding. Second, we converge the request and content digest toward the higher social hierarchy to avoid the pure flooding announcement in sparse MANETs. Third, the content data is forwarded back to the requester based on the social graph between communities instead of the reverse route in the original ICN design.

III. Protocol Design

In this section, we first give the common assumptions that drive some of the design decisions of the protocol. After that, we describe the design of our STCRC scheme in detail.

A. Assumptions

In this paper, the following assumptions are made. Such assumptions are common for delay-tolerant mobile ICN.

- We assume the connection associated with each encounter is bi-directional, thus facilitating two way communications during the period of encounter.
- In the given topology, node-id is used as the unique identifier for a given node.
- We follow the naming scheme in NDN [20].
- Requests can be made randomly for any content from any node.

B. Basis Work

As the basis of STCRC scheme, we describe its basic content retrieval method in a single community [11].

When receiving a Hello message from an encountered node, a data structure, called the encounter-vector, which includes encountered node-id and timestamp of this encounter event, will be created after each encounter event. Every node also maintains a data structure called the encounter table which stores the encounter-vectors created by the node at the encounter time. During the encounter period, the encounter table will be exchanged and stored among the nodes in the same group. After receiving a peer node’s encounter table, each node will merge it into its own encounter table.

1) Compute Social-Tie Relationship: The frequency metric is used to evaluate how frequently two nodes meet each other. We think two nodes have a strong relationship if they meet frequently. The freshness metric is used to evaluate the encounter’s timestamp distribution, reflecting how recently nodes have met each other. A strong social relationship stems from recent rather than remote encounters. Thus, we value recent encounter events higher than older ones. Combining the concepts of frequency and freshness, we define the social-tie concept that will be used to evaluate two nodes’ social relationship.

The encounter event’s contribution to this value is determined by a weighing function $F(x)$, where $x$ is the time span from the encounter event to the current time. Assume that the system time is represented by an integer and is based on $n$ encounter events of node $i$. The social-tie value of node $i$’s relationship with node $j$ at time $t_{base}$, denoted by $R_i(j)$, is defined as in (1).

$$R_i(j) = \sum_{k=1}^{n} F(t_{base} - t_{jk})$$

where $F(x)$ is a weighing function and $\{t_{j_1}, t_{j_2}, ..., t_{j_n}\}$ are the encounter time when node $i$ met node $j$ and $t_{j_1} < t_{j_2} < ... < t_{j_n} \leq t_{base}$. We take $F(x) = \left( \frac{1}{2} \right)^{\lambda x}$ where $\lambda = 1e^{-4}$ is a control parameter which has been proved in [10].

2) Compute Centrality: Each node maintains a social-tie table that contains the social distances from the current node to all other encountered nodes, and each social-tie comes with a timestamp $t_{base}$ when computed. During the encounter period, the social-tie table is exchanged and merged into the other node’s social-tie table.

Based on the social-tie table, a node can compute each node’s centrality. Centrality measures the average social distance from a given node to all other encountered nodes, which can be computed as in (2), where $N$ is the number of nodes observed from the social-tie table, and $R$ is the social-tie value from the given node to each of other nodes.

$$\frac{\sum_{k=1}^{N} R_i(k)}{N}$$

We propose a technique for centrality estimation which considers both the average social-tie values and their distribution to achieve a higher degree of centrality. We adopt Jain’s Fairness Index mechanism [8] to evaluate the balance distribution of social-tie values. As in equation (3), Jain’s Fairness Index is used to determine whether users or applications are receiving a fair share of network resources.

$$Jain's\ Fairness\ Index:\ balance = \frac{(\sum x_i)^2}{n \times \sum x_i^2}$$

Jain’s equation rates the balance of a set of values. The result ranges from 1/n (worst case) to 1 (best case). Jain’s metric identifies underutilized channels and is not unduly sensitive to a typical network flow pattern. In our approach, Jain’s fairness index is used to evaluate the balance of social-tie connection. The centrality metric is defined in (4), where $N$ is the encountered node count in the encounter table.

$$C_i = \alpha \frac{\sum_{k=1}^{N} R_i(k)}{N} + (1 - \alpha) \frac{(\sum_{k=1}^{N} R_i(k))^2}{N \times \sum_{k=1}^{N}(R_i(k))^2}$$

Here $\alpha$ (set in our experiment as 0.5) is a parameter decided by the user according to the specific scenario and network conditions.
3) Content Digest Convergence: A node can compute each observed node’s centrality from its social-tie table and forms a sequence of social relationship among the observed nodes in the network. A higher centrality a node has, a higher probability to meet other nodes than lower level nodes. In the content query phase, in order to avoid pure flooding, we design a content digest convergence process. The basic idea is that each content provider actively announces its content name digest to higher centrality nodes. When a node encounters another higher centrality node, it will send its content name digest with the timestamp to that node. Each node maintains a local data structure called digest table to store the received digests from lower centrality nodes.

Then, the digest table will be sent to another higher centrality node encountered later. If a node receives multiple copies of the content name digest from the same content provider, the digest table is updated according to the freshest timestamp. In this way, each node collects content name digests from lower centrality nodes, and reports the collected digests to higher centrality nodes. Thus, the content name digests from each content provider are converged toward higher centrality nodes in the network. Higher centrality nodes have wider knowledge of the content name digests, and know which content provider contains which content.

4) Content Request: When a content requester requests certain content, an Interest packet will be generated containing the requester’s node-id and content name. This is forwarded to a higher centrality node to avoid naive flooding, because a higher centrality node has more knowledge on content name providers. Each node can compute the centrality of the newly encountered node from its local social-tie table. If we compute the centrality of each node in the social-tie table and sort them in order, we will find that the interval of centrality is not even, as shown in Fig. 1. If the relay node has a similar centrality to the current node, they may have a similar knowledge on the content name digests. Thus, we may not get much benefit from this forwarding. Intuitively, we prefer a relay node whose centrality has enough difference than that of the current node, to further reduce transmission cost. Inspired by K-means clustering algorithms, we periodically divide nodes into clusters according to their centrality distribution, and forward the Interest packet to a newly encountered node that belongs to a higher centrality cluster, as shown in Fig. 2. The Interest packet is only forwarded from cluster A to cluster B. There is no Interest forwarding within a cluster.

We employ Lloyd’s K-means clustering algorithm [9], which is proven to have polynomial smoothed running time [1]. In the K-means clustering algorithm, K is a parameter determined according to the specific scenario and network scalability. A larger K value will benefit the packet delivery ratio but cause higher transmission cost.

![Fig. 1. Centrality sequence](image)

5) Interest Packet Forwarding: As described above, we use the K-means clustering algorithm to build a social hierarchy, and nodes in social-tie table are assigned into different levels. The requester carries the Interest packet and forwards it to the first encountered node that has a higher social level than itself. Subsequently, the requester keeps a copy of the Interest packet and forwards to the next encountered node that has an even higher social level than the relay node it forwarded to last time.

After a node receives the Interest packet from other nodes it encountered, it will first check its local digest table to see if there is any matched name. If no matched name is found, it will continue forwarding the Interest packet. Each relay node performs the same strategy: forwarding the Interest packet to the next relay node that has a higher social level than the last relay node. This is because it is more efficient to forward an Interest packet to a relay node with a higher social level than the last relay node. Following this strategy, the Interest packet is forwarded upward level by level or jumps to a higher level toward the most popular node in the centrality hierarchy.

Since the content name digest keeps being updated and converges toward the higher social level nodes, the query of Interest passing toward the higher social level in the network will be solved eventually when the Interest name matches one content name in the digest table for a certain node at some level of the hierarchy. At this point, the Interest packet will turn into social-tie routing toward the destination since the content provider id has now been disclosed.

There is a potential problem caused by DTN. Similar with the convergence issue in a link-state routing protocol, due to the carry-and-forward scheme in DTN, the social-tie table convergence suffers from a significant delay, which causes the problem that the information used to compute the centrality sequence is not consistent between nodes. In order to make the design practical, we build the social relationship in a distributed method and the computing result comes from a node’s local database (i.e., social-tie table). When an encounter happens, the local social-tie table gets updated to refresh the social relationship result. This can be treated as a learning phase while the social relationship becomes more accurate during each update. Since the previous contact information can be used to predict the future encounter, and the social-tie table grows to be more accurate, the impact of inconsistent social-tie table diminishes and can be tolerated as time progresses. However, there may still be a routing loop due to the inconsistent centrality sequence.

A Time-To-Live (TTL) setting is configured in the Interest packet and counts down during a content query phase. And a waiting timer is setup by content requester after sending out the Interest packet. A routing loop will cause the TTL to go to zero and the requester’s waiting timer runs out. In this case,
we provide a fallback forwarding strategy that the relay node always delivers the Interest packet to a higher social level node than itself, not the higher social level than the last relay node in the previous method.

If the fallback forwarding strategy is adopted and the waiting timer runs out again, the content requester will begin epidemic routing [18] to flood the Interest packet throughout the network. The header of the Interest packet has a forwarding flag to indicate one of three forwarding strategies set by the content requester, and each relay node follows the specific forwarding strategy during the forwarding process.

6) Social-Tie Routing: In this step, the content provider’s node-id has been disclosed and attached in the Interest packet. Hence, the task is to forward the Interest packet to the destination. Similar with the centrality sequence, using the local social-tie table and K-means clustering, we can generate a content provider’s social-tie sequence and build a social-tie hierarchy. The relay node then forwards the Interest packet to a newly encountered node who has a higher social-tie level to the destination node compared to its own, and follows the three forwarding strategies indicated by the flag in the Interest packet header. In summary, in the content query phase, an Interest packet is forwarded toward higher social-tie level since a higher centrality node has more knowledge of content name digest in the network. After the Interest matches a content name digest, we turn into the Interest packet delivery phase which forwards the Interest to the content provider. In this phase, the Interest is forwarded toward higher social-tie level of the content provider since a higher level node is closer to the destination.

After the Interest packet reaches the content provider, the content provider will send the content back to the requester using the social-tie routing again and copy the forwarding flag from the Interest packet header. Each relay node will compare newly encounter node’s social-tie of the content requester and forward the content toward higher social-tie level of the content requester. The content provider only responds once to the same Interest packet and requester, and ignores the following received duplicate Interest packets. In the content data retrieval period, the destination node-id is the content requester. The content data packets will be forwarded using the same forwarding strategy that we use for the Interest packet.

C. Inter-Community Content Routing

In this section, we extend above implementation of single community routing to support efficient content retrieval in networks with multiple communities. We propose a greedy approach to quickly inject content requests into neighboring communities and a heuristic to select the best gateway node to relay packets between communities.

1) Interest Packet Forwarding: In the worst case, intermediate relay nodes may not have knowledge of the content provider for the requested content, and the Interest packet will eventually be forwarded to the cluster head, which is the node that has highest social level in the community. If the cluster head has no information about the content provider, it is likely that the requested content does not exist in the community that the content requester belongs to. In the case of a single community, the cluster head performs no action. However, with multiple communities, the local cluster head often does not know the identity of the content owner which is not within the cluster head’s own community. Though we allow the content name digest to be exchanged between communities, we found that the content name digest relayed by gateway nodes often contains limited information and does not capture a large amount of content that a community owns. The reason is because gateway nodes typically reside at the border (not at the center) of a community. Therefore, gateway nodes often do not have high centrality in their own community, which is an important metric to determine whether a node has a broad knowledge of its community.

Since the local cluster head cannot reliably rely on alien content name digests to obtain global knowledge of all communities, we follow a quick and greedy approach to locate content in neighboring communities. After an Interest packet reaches the local cluster head, the local cluster head will check its local digest table to see if there is any matched name. If no matched name is found, it will select the best local gateway node for each foreign community. We define the best gateway node for a foreign community X as a local node that meets the most nodes belonging to community X. Then, the cluster head social-tie routes the Interest packet to each of the gateway nodes by forwarding the Interest packet to the next relay node that has a higher social-tie value to the destination/gateway node compared to its own.

After the Interest packet reaches the gateway node, the gateway node will forward the Interest packet to a foreign node upon their next encounter event. To reduce the transmission cost overhead, the gateway node records the identity of the last foreign encounter node for each known community. Gateway nodes only forward the Interest packet to a foreign node that has a higher social level than the last encounter node which also belongs to the same community with the newly encountered node. Note that it is possible for a gateway node to infer and compare social level of foreign nodes since we allow gateway nodes from different communities to exchange and merge their social-tie tables, resulting in global network state to be spread across all communities. Once the Interest packet reaches the foreign community, we can proceed with intra-community routing to locate the content provider.

The Interest packet forwarding strategy is similar in the case where the cluster head knows the identity of an alien content provider. In this scenario, the cluster head selects the best gateway node for that community only, and then social-tie routes the Interest packet to that gateway. This is different from the case above where cluster head does not know which community owns the content and therefore it has to social-tie route the Interest packet to multiple gateway nodes corresponding to multiple communities. Once the gateway node injects the Interest packet to the target community, social-tie routing will be used to route the Interest packet directly to the content provider without the need to go through foreign
cluster head. Pseudocode 1 summarizes our Interest packet forwarding strategy.

**Pseudocode 1 Interest Forwarding Strategy**

```plaintext
1: when an Interest packet is received
2: check my local content table
3: if there is a match then
4: if content provider belongs to a different community then
5: select the best gateway node for target community
6: social-tie route Interest packet to selected gateway node
7: else
8: social-tie route Interest packet to local content provider
9: end if
10: else
11: if I am a cluster head then
12: for each known neighboring community X do
13: select the best gateway node for community X
14: social-tie route Interest packet to selected gateway
15: end for
16: end if
17: end if
```

2) Content Packet Forwarding: After the Interest packet reaches the foreign content provider, the content provider will send the content to the best local gateway node for the community the requester belongs to. Subsequently, the gateway node will inject the content into the requester’s community. Then, social-tie routing can again be employed to route the content to the original requester.

Fig. 3 illustrates all the steps from when a node requests a content until the content is delivered to the original requester.

1) An Interest packet is generated and routed to A’s cluster head using social-level routing.
2) A’s cluster head social-tie routes the Interest packet to A’s best gateway node.
3) A’s gateway node injects the Interest packet to community B through B’s border node.
4) B’s border node propagates the Interest packet to B’s cluster head through social-level routing.
5) B’s cluster head social-tie routes the Interest packet to the content provider.
6) The content provider social-tie routes content to B’s best gateway node.
7) B’s gateway node forwards content to community A through A’s border node.
8) A’s border node social-tie routes content to the original requester.

**Fig. 3.** Steps in locating the content provider across communities and routing the content back to the original requester

IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed STCRC scheme in a packet-level simulation using a synthetic trace. We first describe the properties of our synthetic trace, followed by the simulation setup, the metrics used and the results.

A. Synthetic Trace

As shown in Fig. 4, we generate a synthetic trace that features multiple communities, and a small subset of nodes which move frequently from one community to the other. This trace was designed to emulate the interaction of nodes in two separate communities. Nodes within each community are clustered into smaller groups of sub-communities, to ensure a heterogeneous social structure with certain nodes featuring higher centrality values. Nodes which move between the two communities are instrumental for forwarding the Interest packet to the destination node.

![Fig. 4.](image-url) This figure illustrates the network topology used to evaluate the proposed mechanism. Nodes are categorized into two separate groups featuring sub-communities, and a small subset of nodes typically traverse the divide in-between the two communities to relay information back and forth.

Since each community is comprised of multiple communities with varying radii, there is significant variation in the centrality level of different nodes within a community. Furthermore, the movement speed of different nodes, as well as the pause time, varies considerably. Therefore, nodes which move more quickly will be exposed to more nodes and are therefore more likely to be in a higher social level.

In order to develop a complex trace featuring the interaction of multiple communities, many separate trace files were generated and merged. Nodes were grouped to a particular location through the assignment of an attraction point to a location of the simulation area, with a particular standard deviation of attraction, to ensure that nodes do not converge onto the same point. Nodes responsible for relaying content and Interest packets between communities are assigned multiple attraction points in separate communities.

B. Simulation Setup

We implemented the proposed STCRC scheme using the NS-3.19 network simulator. DTN nodes advertise their Hello message every 300 ms. In order to test the bottom line of the performance, we assume that each node has unique content
We use the following metrics in the experiments: copying the messages to all nodes that do not yet have a copy. In Epidemic creating an unlimited number of messages by exchanging messages that they have not yet seen. This results in Epidemic routing, when two nodes encounter each other, they connect to the community (Epidemic has the highest delivery probability). In routing in an attempt to understand the upper bounds of connectivity (Epidemic has the highest delivery probability). The performance of Epidemic is marginally higher than STCRC. The gap between two schemes is around 15%. Furthermore, in the presence of intra-community requests, the discovery of content provider occurs more quickly, resulting in improved hit rates for both Epidemic and STCRC.

2) Average Delay: Fig. 6 and Fig. 9 illustrate the average delay of the two methods when the number of requests is varied. The figures show that the delay of Epidemic is lower than our scheme. In Epidemic, Interest and content are rapidly flooded in the network. As a result, the Interest and content can reach its destination after a short delay. Our STCRC scheme, however, suffers a delay to propagate the Interest to the local cluster head node and also to the foreign cluster head nodes in the case of an inter-community request. Similar to the hit rate experiments, in the presence of intra-community requests, the delay decreases for both schemes.

3) Total Cost: Fig. 7 and Fig. 10 show the average cost of the two methods. The cost of Epidemic is much higher than STCRC as the number of content requests increases. In Epidemic, each node delivers the Interest or content packet to every encountered node, thus resulting in very high cost. In STCRC, the choice of relay node is very selective. Within a community, to discover the content provider, STCRC forwards Interest only to nodes with a higher social level. Across communities, STCRC lets the border node inject Interest packet only to the foreign node with a higher social level than the last foreign encountered node. Both strategies lead to significantly low cost for STCRC while ensuring the successful discovery of the content provider. As in the case of average delay, when the content requesters and content providers belong to the same community, the cost is reduced for both schemes.

4) Impact of K-means: The parameter K is set to 10 for the above experiments. A higher value of K results in more clusters in the network, thus generating more packet forwarding which benefits hit rate and average delay at the expense of total cost. Therefore, the value of K should be selected carefully according to specific scenarios. We have evaluated the performance of our proposed STCRC scheme for different values of K. Fig. 11-13 show the impact of K across all three evaluation metrics.

V. CONCLUSION

In this paper, we propose STCRC, a social-tie and community aware content retrieval architecture that is highly scalable in disruption-tolerant mobile information-centric networks. The STCRC generates the social-network based routing structure in order to support the efficient Interest and content data forwarding among communities. The evaluation results show that our system requires less control overhead while maintaining comparable performance for content retrieval applications.
REFERENCES


