UNIVERSITY OF CALIFORNIA

Los Angeles

# Information Centric Mobile Ad Hoc Communications

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Computer Science

by

You Lu

2015

ABSTRACT OF THE DISSERTATION

# Information Centric Mobile Ad Hoc Communications

By

You Lu

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2015

Professor Mario Gerla, Chair

Information-centric networks have recently been drawing increasing attention in academia as well as in industry. Information and content retrieval is a critical service for mobile ad-hoc networks. It relies on many resources and tools, such as internal storage, content searching and sharing, delay-tolerant delivery, etc. Previous studies have shown that conventional ICN interest query schemes and content searching architectures, if not properly designed, can cause significant performance degradation and energy consumption, especially for large scale mobile ad hoc networks. In this dissertation, we propose a content retrieval architecture for ICN MANETs that is structured according to social hierarchy and is highly scalable.

The dissertation of You Lu is approved.

Danijela Cabric

Milos Ercegovac

Majid Sarrafzadeh

Mario Gerla, Committee Chair

University of California, Los Angeles

2015

# TABLE OF CONTENT

# LIST OF FIGURES

# LIST OF TABLES

# ACKNOWLEDGEMENTS

My immense gratitude goes to my advisor, Dr. Mario Gerla, who guided and inspired me in every aspect throughout my whole doctoral study. His encouragement and guidance is critical to my academic achievements. I would like to thank Dr. M. Y. Sanadidi, who offered lots of insightful. I also appreciate other members in my doctoral committee, Dr. Danijela Cabric, Dr. Milos Ercegovac, and Dr. Majid Sarrafzadeh, for their insight and advice. I thank all of my colleagues and peers in UCLA Network Research Lab: Biao Zhou, Yu-Ting Yu, Ian Ku, Lung-Chih Tung, Jui-Ting Weng, Tuan Le, Pengyuan Du, and many others for their help and advice. Finally, this study would not have been possible without the support of my family, especially my wife, Hongxia Zhou. Thank all the people around who have helped me and made my life incredibly rich and colorful.

# VITA

2004 B.S., Electrical and Information Engineering, Beijing University of Posts and Telecommunications, Beijing, China

2007 M.S., Communication and Information System, Beijing University of Posts and Telecommunications, Beijing, China

2012 M.S., Computer Science, UCLA, Los Angeles, California

# SELECTED PUBLICATIONS

1. T. Le, Y. Lu, and M. Gerla, “Social caching and content retrieval in Disruption Tolerant Networks (DTNs),” in Computing, Networking and Communications (ICNC), 2015 International Conference on, 2015, pp. 905–910.
2. P. Du, X. Li, Y. Lu, and M. Gerla, “Multipath TCP over LEO Satellite Networks,” in The International Wireless Communications & Mobile Computing Conference (IWCMC 2015), 2015.
3. Y. Lu, M. Gerla, T. Le, V. Rabsatt, and H. Kalantarian, “Community aware content retrieval in disruption-tolerant networks,” in Ad Hoc Networking Workshop (MED-HOC-NET), 2014 13th Annual Mediterranean, 2014, pp. 172–179.
4. Y. Lu, X. Li, Y.-T. Yu, and M. Gerla, “Information-centric delay-tolerant mobile ad-hoc networks,” in Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on, 2014, pp. 428–433.
5. I. Ku, Y. Lu, M. Gerla, F. Ongaro, R. L. Gomes, and E. Cerqueira, “Towards software-defined VANET: Architecture and services,” in Ad Hoc Networking Workshop (MED-HOC-NET), 2014 13th Annual Mediterranean, 2014, pp. 103–110.

6. I. Ku, Y. Lu, and M. Gerla, “Software-Defined Mobile Cloud: Architecture, services and use cases,” in Wireless Communications and Mobile Computing Conference (IWCMC), 2014 International, 2014, pp. 1–6.
7. Y. Lu, W. Cheng, L.-C. Tung, and M. Gerla, “Social-tie Based Content Retrieval for Delay-tolerant Mobile Ad-hoc Networks,” in Proceedings of the 8th ACM MobiCom Workshop on Challenged Networks, 2013, pp. 51–52.
8. Y. Lu, B. Zhou, L.-C. Tung, M. Gerla, A. Ramesh, and L. Nagaraja, “Energy-efficient Content Retrieval in Mobile Cloud,” in Proceedings of the Second ACM SIGCOMM Workshop on Mobile Cloud Computing, 2013, pp. 21–26.
9. Y. Lu, Z. Wang, Y.-T. Yu, R. Fan, and M. Gerla, “Social network based security scheme in mobile information-centric network,” in Ad Hoc Networking Workshop (MED-HOC-NET), 2013 12th Annual Mediterranean, 2013, pp. 1–7.
10. Y. Lu, K.-H. Su, J.-T. Weng, and M. Gerla, “Mobile social network based trust authentication,” in Ad Hoc Networking Workshop (Med-Hoc-Net), 2012 The 11th Annual Mediterranean, 2012, pp. 106–112.
11. Y. Lu, Y. Lin, B. Zhou, and M. Gerla, “MPR-aware performance improvement for inter-domain routing in MANETs,” in MILITARY COMMUNICATIONS CONFERENCE, 2011 - MILCOM 2011, 2011, pp. 673–678.
12. Y. Lu, B. Zhou, F. Jia, and M. Gerla, “Group-based secure source authentication protocol for VANETs,” in GLOBECOM Workshops (GC Wkshps), 2010 IEEE, 2010, pp. 202–206.
13. Y. Lu, B. Zhou, I. Ku, and M. Gerla, “Connectivity improvement for inter-domain routing in MANETs,” in MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010, 2010, pp. 617–622

# Chapter 1. Introduction

Mobile ad hoc networks (MANETs) are most effective in dynamic environments where network infrastructure is 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, different organizations or administrations equip and operate their ad hoc networks according to their own priority and cost constraints, resulting in fundamentally different designs, e.g., their multi-access schemes and routing protocols. In a disaster rescue scenario, police, firemen, and medical crews from different organizations with different MANETs may benefit from interconnecting to share terrain and traffic information and to coordinate rescue activities. Each group may use different technologies from fundamental communication protocols (e.g. 2.4GHz vs. 5GHz, and CSMA vs. TDMA [1]), to routing algorithms (e.g. OLSR [2] vs. AODV [3]). This will lead to challenges even if the upper layers applications run on standard protocols such as HTTP and TCP/IP. The challenges in inter-MANET routing, relative to the Internet, include dynamic network topology, intermittent connectivity, dynamic membership and its management, and routing protocols heterogeneity.

In general, tactical and vehicular MANETs must support various services such as communication, storage, and computing for a range of applications. The importance of MANET design is the ability to manage and serve queries to the large amount of contents and resources distributed among different nodes.

In current Internet, the information-centric network (ICN) is designed for content search and retrieval. An alternative approach is the IP-based computer network architecture. In ICN, users only focus on the content they are interested in, they need not know where this content is stored and carried. We assume content is identified by a unique name. Content retrieval follows the query-reply mode. Content consumer spreads his Interest packet 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 reversed route of the incoming Interest.

One major design challenge of the content retrieval in MANET is the energy-efficient Interest dissemination scheme. Especially for MANET, the dynamic network topology and intermittent connectivity are sensitive to the traffic load. Two common approaches to disseminate the Interest packet to the entire network are intelligent broadcast and probabilistic forwarding. They try to reduce the duplicate copies of Interest packet that each node received to decrease the traffic overhead. However, both solutions will spread the Interest packet to the entire network until the edge, and every single node in the network will be delivered at least one copy of the Interest packet. These solutions are only minor lightweight than the naive flooding, and will suffer from the scalability problem in a large scale MANET.

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. Thus, the sparse MANET architecture must support critical services, e.g., internal data storage, content search and sharing, etc.

Another major design challenge in MANET is to design an efficient content retrieval scheme in intermittent connectivity. 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, social network assisted disruption-tolerant routing cannot be deployed directly in mobile ICN, since the destination location, where the content is carried, is not exposed during the content search period, and naive-flooding query method will produce a high transmission cost. Moreover, content search and content delivery phase are sequential and , as we shall see, cannot use the same social network for routing.

To meet the challenges of content retrieval in MANET, in the first part of this dissertation we propose a new Heterogeneous Inter-MANET Routing (HIMR) protocol to achieve scalability and robustness to mobility by using clustering technique. The basic structure of HIMR includes clusters and gateways in each MANET. The distributed clustering algorithm selects gateways within each MANET. By its multiple activated interfaces, gateways connect MANETs with different routing protocol, different channels and different physical or link layer protocols. Thus, the new routing scheme of HIMR can operate across these incompatible MANETs using different PHY/MAC protocol and different radio channels, and make the final route selection based on the replies from different underlying MANET routing protocols.

To address the challenge of efficient query for content retrieval in MANET, in the second part of this dissertation we propose an energy-efficient content retrieval scheme to enhance the performance. We specifically address the scalability and energy efficiency of the Interest dissemination for large scale MANETs. We propose a direction-selective forwarding scheme for the Interest dissemination phase to decrease the traffic overhead of the duplicate copies of the Interest packet and save the transmission energy. Note that not every node in this scheme will receive the Interest packet. There are some missing gaps in the network where the Interest will not be delivered to. But these gaps don't contain high degree nodes or giant component of the MANET. We advocate performing the parallel search in which content requester selects several agents by random-walk to conduct the proposed direction-selective Interest dissemination scheme parallel. A content provider delivers several copies of his content data by random-walk to his neighbor's caches. The evaluation results show that the proposed content retrieval scheme is capable of retrieving any content in the large scale MANET with high hit rate and low traffic overhead.

To handle the challenge of successful content delivery in intermittent MANET, in the third part of this dissertation, we propose a Social-Tie based Content Retrieval (STCR) scheme in delay-tolerant MANETs, which address the scalable content retrieval in large-scale sparse MANETs. In STCR, nodes record the essential information when they encounter each other. To compute a social graph of encounter relationships, we build a hierarchical architecture using K-mean clustering algorithm based on the social-tie between nodes. This hierarchy is used to improve forwarding strategy's efficiency. We propose novel methods to compute the social-tie relationship considering both frequency and recency, and to compute centrality sequence considering both average social-tie and its balanced distribution. After the query is matched by one of the content digests, the search process will turn to social-tie routing to reach the content provider. We also propose a cooperative, socially inspired caching scheme in which popular content data are cached at cluster head nodes.Yet, due to the limited caching buffers in the mobile nodes, we also consider distributing cached data along content query paths. Neighbors of downstream nodes may also be involved for caching when there are heavy data accesses at downstream nodes. That is, downstream nodes move some of their existing cached data to neighboring nodes to make room for new data. Finally, we consider dynamic cache replacement policy based on both the frequency and fressness of data access.

The rest of this dissertation is organized as follows. The background and related work is discussed in chapter 2. The HIMR protocol is introduced in chapter 3. The Energy-Efficient Content

Retrieval protocol is described in Chapter 4. The Information-Centric Content Retrieval protocol is proposed in Chapter 5. Finally, we conclude the dissertation in chapter 6.

# Chapter 2. Background

## 2.1 Inter-MANET Routing Protocol

In this section, we introduce the existing inter-MANET routing protocols and techniques. A number of previous studies focused on the high level architectures and provided sketches of the required components, thus complementing our effort, e.g., the translation of naming spaces, protocol translation, BGP-style routing, and support for node mobility. Crowcroft et al. proposed Plutarch to translate address spaces and transport protocols among MANETs, thus enabling the interoperation of heterogeneous networks [4]. TurfNet is another proposal for Inter-MANET networking without global network addressing or a common network protocol [5]. Previous studies show that interconnecting heterogeneous MANETs requires several components and considerations: 1) the fundamental framework of inter-MANET routing which includes the basic definition and assumption of inter-MANET architecture, routing policy and gateway structure; 2) heterogeneous routing solutions that interconnect the different intra-MANET routing schemes; 3) dynamic gateway selection algorithm efficiently interconnecting multiple MANETs; 4) efficient MAC layer protocol, especially an optimized dynamic TDMA scheme in TDMA MANET to enhance the throughput and gateway selection.

### 2.1.1 Border Gateway Protocol (BGP)

In Internet, inter-domain routing enables interoperations among heterogeneous domains that usually employ different routing protocols and policies. The Border Gateway Protocol (BGP) [6] is the de facto inter-domain routing protocol for the Internet. BGP provides a standard mechanism for inter-domain routing among heterogeneous domains or autonomous systems (AS). The principle of BGP is to enable opaque interoperation, where each domain has the administrative control over its intra-domain routing protocol and inter-domain routing policy. In BGP, the routes to an internal destination within the same domain are determined by an intra-domain routing protocol, whereas the routes to an external destination are determined by the inter-domain routing policies among domains. BGP relies on a path vector protocol for exchanging inter-domain level reachability information. One of the advantages of the path vector protocol is that it makes it easy to detect a loop in a route. Also it makes it easy to specify domain administrator's preferences in the route selection thereby enabling a policy-based routing. Despite several reported inefficiencies, BGP has been operating nonstop in the Internet for the past two decades. There is a vast body of literature on BGP and its properties, including scalability, control overhead and security. However, these results are not directly applicable to MANETs because BGP design is based on a static Internet, and cannot survive in mobile, dynamic topology environments.

For example, BGP's capability to handle large numbers of routes makes it potentially valuable to large scale tactical networks. However, large scale causes slow convergence after routing changes. This is obviously not a significant issue in terrestrial networks, since links are generally very stable. But these BGP limitations are intensified by the MANET environment. Frequent network topology changes are possible due to the node movement in MANETs. Links can appear and disappear very quickly in this environment. Since BGP uses TCP for reliable control message exchange, it will be extremely vulnerable in such mobile environment. Likewise, BGP cannot support dynamic discovery of its members.

### 2.1.2 Mobile Ad-hoc Inter-MANET Networking (MAIN) Framework

The MAIN (Mobile Ad hoc Inter-MANET Networking) framework recently proposed in [7][8] assumes that each MANET functions as an autonomous system (AS) in the extended wireless Internet. MAIN requires special gateway nodes, and relies on a path vector protocol to support policy-based routing. In MAIN, the system of inter-connected MANETs is assumed to be traffic driven, i.e., reactive rather than proactive. Thus MAIN proposes a reactive path vector protocol. The proposed framework requires no surrender of the administrative control by each MANET. Thus each MANET can use its native Intra-MANET routing protocols without change, and specify Inter-MANET routing policies in the spirit of the policy-based routing as supported by BGP in the Internet.

But, there are several open issues that MAIN needs to handle: 1) partition and merge of MANETs; 2) membership announcement; 3) gateway function design; and 4) support for policy based routing. The first two points are due to node mobility and dynamic topology, and the latter two are general issues with Inter-MANET routing preserving the administrative autonomy of each MANET.

### 2.1.3 Interconnecting Heterogeneous Routing Solutions

Heterogeneous routing is a problem that must be faced when interconnecting MANETs. There have been proposals to take advantage of heterogeneous routing protocols to adapt to network dynamics and traffic characteristics.

Hybrid routing protocols combine different style routing protocols and adaptively use them to improve the performance in a single subnet. For example, SHARP [9] is an adaptive hybrid routing protocols that uses both proactive and reactive routing protocols to balance the tradeoff between the two and improve the performance. To achieve this goal, SHARP creates proactive routing zones around nodes with heavy traffic, and uses a reactive routing in other areas. Hybrid routing's main goal is to improve performance in a single zone via adaptation to the nodes traffic level. Our focus is on the internetworking of heterogeneous routing MANETs, seeking solutions that are independent of the specific internal routing protocols.

Cluster-based networking (such as LANMA [10], ZRP [11]) is similar to Inter-MANET routing in that it is also concerned with the interaction among clusters of nodes at the network layer. The idea of cluster-based networking is to form self-organizing clusters and a routing backbone among cluster heads with the major advantage being scalability. Although cluster-based routing has a structural similarity with Inter-MANET routing, there are fundamental differences. Inter-MANET routing deals with multiple heterogeneous MANETs with autonomous control; the hierarchy of the network (i.e. MANETs) is a given. On the other hand, in classic cluster-based routing, the nodes are under an autonomous subnet, and may be aggregated into clusters, possibly with an optimized cluster size, taking into account the nodes geographic proximity, traffic level, etc. Thus clustering is actually more appropriate for a stationary rather than mobile network.

### 2.1.4 Dynamic Gateway Selection

In inter-MANET routing scenario, gateways must be selected carefully to ensure efficiency and accuracy of the communication. They provide a “bridge” between MANETs and are crucial for inter-MANET connections. IDRM [8] considered a simple case where some nodes are initially pre-designated as gateways whose movement is determined by their mission objectives not by infrastructure goals such as constructing an inter-MANET backbone. However, in some cases, such an approach will result in sub-optimal performance in terms of inter-MANET connectivity. For example, when a single MANET is partitioned into multiple sub-MANETs, it is possible that some of them may not have any gateways to connect to the rest of the MANETs. There are various ways to tackle this issue, e.g., by controlling the mobility of gateways, or by selecting gateway nodes dynamically in response to topology change. In this dissertation, we take the latter approach since we feel mobile nodes are now powerful enough to perform multi-protocol translation and extra inter-MANET operation. However, making all nodes gateways is not a good design because: 1) it will quickly deplete the scarce battery power even when the node is not participating in inter-MANET communications, and 2) it will generate excessive control messages between any nodes in the entire network. In this dissertation, we take a practical approach assuming that there are a set of nodes in each MANET that can become gateways when needed. We call them potential (or candidate) gateways and when they become actively involved in Inter-MANET routing operation, we call them active gateways. When gateways are active, they maintain inter-MANET routing information; perform protocol translation, and policy-based data forwarding.

## 2.2 Information-Centric Networks

In this section, we first introduce the general idea of information-centric network, and review the content retrieval methods in Internet and mobile ad hoc network. We also analyze the problems of content retrieval in MANET application.

Information-centric network is an alternative approach to the architecture of IP-based computer networks. The basic principle is that user only needs to focus on his interested content data, rather than having to reference a specific, physical location where that data is to be retrieved from. ICN differs from IP-based routing in three aspects. First, all content is identified or named by the hierarchical naming scheme. Name becomes the object of request. Second, carefully designed caching system through the entire network helps the content distribution and provides the native features to help many other applications, e.g., multicast. Third, the packet communication follows the form of query-reply mode. User (content requester) spreads his interested content name as the "Interest" packet through the network. Once the "Interest" packet hits the content name in any intermediate cache server or the media server (content provider), the content data packets will be forwarded back to the content requester along the reversed incoming route of the Interest.

A number of previous studies focused on the ICN with high level architectures and provided sketches of the required components. Content-centric network (CCN) [12] and named data network (NDN) [13] are two implemented proposals for the ICN concept in Internet. Their components including FIT, PIT, and Content Store form the caching and forwarding system for the content data in Internet application. However, neither the CCN nor NDN can be deployed directly in the mobile ad hoc network, since the dynamic network topology and intermittent connectivity causes the difficulty to maintain the caching and forwarding scheme. Several mobile ICN architecture designs have been proposed for the mobile ad hoc scenario, e.g., Vehicle-NDN [14] for the traffic information dissemination in vehicular networks, and MANET-CCN [15] for the tactical and emergency application in MANETs. One of the challenges for content retrieval in mobile ad hoc networks is the design of Interest dissemination scheme. It is obvious that the naive flooding method causes large traffic overhead and energy cost during the content request process. Researchers have proposed a number of improved forwarding protocols aiming at reducing the flooding overhead. Two typical categories of them are back-off timer based intelligent broadcast and probability based probabilistic forwarding.

Intelligent broadcast, also referred to as smart broadcast, is a position-based protocol aiming at the maximization of the one hop progress of the Interest packet dissemination and minimization of the forwarding delay, e.g., enhanced multi-hop vehicular broadcast (MHVB) [16], road-based directional-broadcast [17], and effective broadcast [18]. It is accompanied by a mathematical model providing a means to set the protocol's parameters optimally. Each node within the source's transmission range will compute a back-off timer before forwarding the message. The duration of this back-off timer is related to the relative geo-distance to the source. So a node far away from the source gets a short back-off value, and becomes the first one to relay to the source with message forwarding. The other nodes with a longer back-off timer will cancel their scheduled transmission upon hearing this forwarded message. The back-off timer based intelligent broadcast significantly decreases the overlap area of the packet forwarding and reduce the duplicate copies of the Interest packet. It also guarantees that every node in the network will receive at least one copy of the Interest packet.

The probabilistic forwarding is a further improved intelligent broadcast method, e.g., epidemic broadcast [19]. Similar to the intelligent broadcast, it setups a back-off timer based on the distance to source. During each broadcast interval, extended by the waiting time, the repeaters count the duplicate messages they received from their front and back. At the end of the waiting time, they enter a decision process instead of immediately sending their Interest packet. So the forwarding decision at the end of each interval is based on the duplicate messages during the waiting time. And then it will compute the probability of keeping the message alive based on a mathematical model. The decision process favors those nodes with an unbalanced message count, which means they are closer to the edge of the source's transmission range. The object of probabilistic forwarding is the same as the intelligent broadcast that they always try to spread the Interest packet to the entire network and keep the minimum traffic overhead.

Both the intelligent broadcast and probabilistic forwarding have the same drawbacks in the mobile content retrieval application. First, the Interest packet will be propagated to every node in the network. If the content data only has few copies in the network, this entire network spreading is redundant and inefficient obviously. Second, even if some intermediate nodes (e.g., cache server) match the Interest and reply with the content data, the content request will still be broadcasted to the edge of the network and cannot be terminated before the end.

## 2.3 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 [20]. 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 [21]. 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) [22]. PSNs are formed from opportunistic human contacts, typically by creating ad hoc links between mobile phones [23].

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 [24], 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 [25]. 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 [26].

Previous work studied various types of information that can be used to help packet delivery, including historical contacts, device mobility patterns, and social interaction between the participants. PeopleRank [27] is a typical social network based forwarding scheme. It is inspired by the PageRank algorithm [28] used in Google's search engine to measure the relative importance of a Web page within a set of pages. PeopleRank identifies the most popular nodes to forward the message to, given that a higher PeopleRank value means more "central" in the social graph. The explicit friendships are used to build the social relationships based on their personal communications. One potential problem in PeopleRank is that it is difficult to guarantee that the social graph from friendship and the one induced by the physical connection network are always consistent. The performance of PeopleRank highly depends on how the social graph is built.

SimBetTS [26] attempts to uncover a social network structure in DTN, using egocentric centrality and its social similarity to forward messages toward the node with highest centrality to increase the possibility of finding the potential carrier to the final destination. Although SimBetTS considers both the closeness and betweenness aspects, these two aspects reflect the social relationship and direct the messages toward higher centrality to improve the probability of meeting the destination. The social metrics proposed in SimBetTS can be improved further to implement more reasonable forwarding strategies.

BubbleRap [29] beats all above routing schemes as claimed by the author in terms of accuracy and efficiency. It combines the observed hierarchy of centrality and observed community structure with explicit labels, to decide on the best forwarding nodes. In BubbleRap, the centrality is calculated by the prior flooding experiment first. In order to make it practical, BubbleRap computes the collected data within either 6-hour S-window or cumulative C-window. It proved empirically that past contact information can be used in the future estimation. But all the DEGREE or RANK data collected from the past contact information only reflect the frequency. The S-window method can represent a kind of recency information but decrease the accuracy of the computing by lacking the data before the 6-hour window.

All the above studies attempt to build a hierarchical centrality structure and forward packets toward the higher centrality node. The higher centrality value only means that this node has a higher average probability to meet all other nodes in network other than the lower centrality nodes. If the destination of the message is known, this average probability of higher centrality may not provide a better successful rate of packet delivery for that specific destination since an average encounter probability may not be consistent with the encounter opportunity to one specific node. On the other hand, it is obvious that not all of the higher centrality nodes can provide equivalent benefit during each forwarding. It is necessary to identify which nodes improve utility and which do not even though their centrality is higher than the current node.

# Chapter 3. Inter-Domain Routing Protocols in MANETs

## 3.1 MPR-Aware Dynamic Gateway Selection

In our MANETs we assume OLSR is used. This is because OLSR is one of the most popular routing protocols in MANETs. OLSR uses MPR (Multipoint Relay) to reduce the control overhead [2].

### 3.1.1 Multipoint Relay in OLSR

Multipoint Relay contains a set of nodes in wireless Ad-Hoc networks that do the job of relaying messages between nodes; they also have the main role in routing and selecting the proper route from any source to any desired destination node.

MPR advertises link state information for their MPR selectors periodically in their control messages. MPR selectors are a set of nodes that have selected a particular node in question as their MPR. MPR is also used to form a route from a given node to any destination in route calculation. Each node periodically broadcasts Hello message for the link sensing, neighbor detection and MPR selection processes. Each node can get topology up to 2 hops from Hello messages. The information about the symmetric one-hop and two-hop neighbors is used to calculate the MPR set. Each node selects set of neighbor nodes as MPR from among 1-hop neighbors with symmetric link, which covers all the two-hop neighbors and records in MPR selector table. So, an MPR node will forward its own data as well as the data from the MPR Selectors, as shown in Figure 3.1.

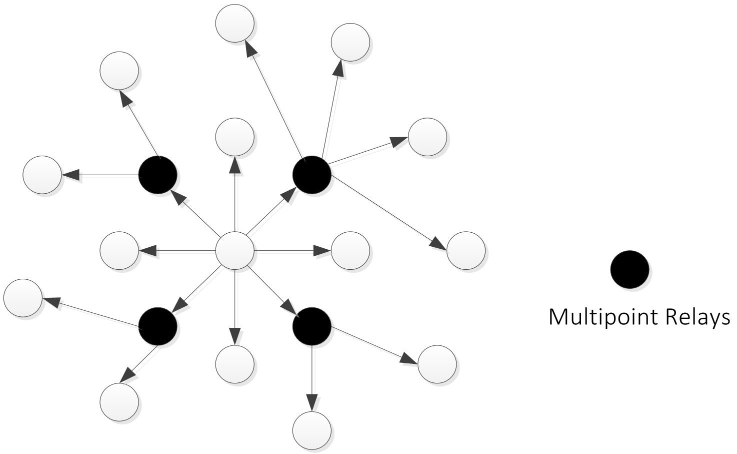


Figure 3.1 Multipoint Relay in OLSR

### 3.1.2 Dynamic Gateway Selection

Potential Gateways, as “bridges” among MANETs, continuously advertise their connectivity, members, inter-MANET routing table, and other information. Gateways must be carefully selected among the candidate set to insure the efficiency and accuracy of communications between MANETs. Next we enhance gate connectivity using the dynamic selection scheme and the MPR concept.

In order to represent the “bridging” efficiency of a candidate gateway, a “neighbor ratio” R is used. R is defined as the ratio of the number of neighbors of the candidate gateway to the total number of nodes in the MANET, as shown in equation (3.1), where n is the gateway’s total number of neighbors, and N is total number of nodes in the network.

$$R = \frac{n}{N} \tag{3.1}$$

However, connectivity depends not only on the gateway’s number of neighbors, but also on its neighbor distribution across multiple MANETs. We propose a new gateway selection method that considers both the number of neighbors and their distribution to achieve higher degree of connectivity. In MANETs, the connectivity of the networks is very dynamic. The primary problem is how to make the connectivity more reliable. The gateway nodes should have more neighbors relative to other nodes, and should also be able to connect more Intra-MANET nodes. Meanwhile, a gateway's neighbor distribution in different MANETs should be balanced. Table 3.1 shows an example. MANET 1 has 10 members; MANET 2 has 20 members; and MANET 3 has 30 members. The numbers of Node 1's neighbors in each MANET are 1, 2, and 3. The numbers of Node 2's neighbors in each MANET are 0, 5, and 6. It is obvious that node 1 is a better gateway because of its balanced neighbor distribution.

Table 3.1 Neighbor's distribution

| | MANET 1 | MANET 2 | MANET 3 |
|---|---|---|---|
| Node 1's neighbors | 1 | 2 | 3 |
| Node 2's neighbors | 0 | 5 | 6 |
| Total members | 10 | 20 | 30 |

We adopt Jain's Fairness Index mechanism [30] to measure the neighbor distribution balance as in equation (3.2). Jain's Fairness Index is used to determine whether users or applications are receiving a fair share of network resources.

$$\text{Jain'sFairness Index: balance} = \frac{(\sum \mathrm{x_i})^2}{n\times \sum x_i{}^2} \tag{3.2}$$

Jain's equation rates the balance of a set of values. The result ranges from $1/_n$ (worst case) to 1 (best case). This metric identifies underutilized channels and is not unduly sensitive to atypical network flow patterns.

In our approach, $B$ is defined as the value of MANET balance, as in (3.3).

$$B = \frac{(\sum x_i)^2}{(n\times \sum x_i{}^2)} \tag{3.3}$$

$n$ is the total number of MANETs. $x_i$ is the ratio of the number of neighbors to the number of members in $MANET_i$. That is (3.4):

$$x_i = \frac{\#\ of\ neighbors\ in\ MANET_i}{\#\ of\ members\ in\ MANET_i} \tag{3.4}$$

If a MANET uses the OLSR routing protocol, the nodes that act as MPR should be more likely to become gateways among the gateway candidates. The more MPR Selectors (MPRS) an MPR node has, the better the opportunity to become a gateway node. We define the weight of an MPR node as the ratio of the number of its MPR Selectors vs. the total number of MPR Selects in its MANET (assuming it runs OLSR), as shown in (3.5).

$$W = \frac{\#\ \ of\ MPRS}{total\ \#\ \ of\ MPRS} \tag{3.5}$$

This weight factor is zero for the nodes that do not run OLSR. The gateway node is selected considering MPR weight $W$, node distribution balance $B$ and transmission efficiency $R$:

$$Gateway\ value = Max[\alpha \times W\ +\ \beta \times B +\ \gamma \times R] \tag{3.6}$$

Here $\alpha, \beta, \gamma$ (set in our experiment as 0.2, 0.5, and 0.3) are parameters that are decided by the user based on the specific scenario and network conditions, and their sum is 1. For example, if there is less MANET amount but more intra-MANET nodes, we prefer larger $\beta$ value since here the balance connection between MANETs is more important. On the contrary, if there are more MANETs but less intra-MANET nodes, we prefer higher $\gamma$ since the higher neighbor amount may lead to better connectivity among MANETs. If one of the relative large MANET runs on OLSR routing protocol, $\alpha$ can give a more weight on MPR nodes to be potentially gateways. The gateway node is selected such that it maximizes the function in (3.6). In the Inter-MANET routing scenario, the gateway selection should consider both topology connections and efficiency/balance. Namely, not only the number of the gateway's neighbors is a determinant fact, but also its connection with neighbors and even coverage of different MANETs. We value not only the packet delivery ratio but also the traffic throughput. In summary, an MPR node is a more desirable gateway candidate than a regular node.

# 3.2 Heterogeneous Inter-MANET Routing Protocol

## 3.2.1 Design Challenges

In this section, the design challenges of inter-MANET routing among MANETs are analyzed. The domains in MANET may split and merge dynamically, which is a salient difference from static networks. There are three key design challenges in the inter-MANET routing among MANETs: network dynamics, membership management, and heterogeneous intra-MANET routing.

### 3.2.1.1 Network Topology Dynamics

MANETs are fundamentally different from wired networks: the networked devices are subject to their mobility patterns and can split from the current routing MANET and merge with a different MANET.

The first problem caused by network dynamics is that it will incur routing loops in path vector based routing like BGP. As shown in Figure 3.2, originally AS 1001 has four nodes in its group, but after a while it splits into two separated parts with two nodes each. The two separated groups will retain the original AS ID as 1001. Any routing update passing through these two groups will have a path vector such as "*→1001→789→1001", and the gateway receiving this routing update will discard it because it has a routing loop in it. If the routing protocol has no way to detect the group split and reassign new AS IDs to them, the path vector routing will not be applicable. By reassigning new AS IDs to split MANETs, this problem can be circumvented.

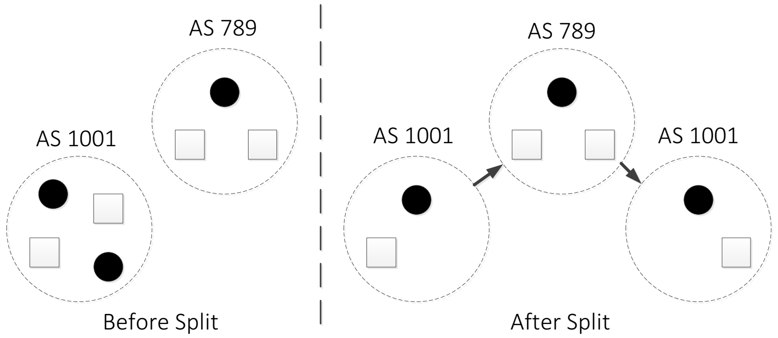


Figure 3.2 Routing loops caused by MANET split.

Secondly, it is not a trivial task to detect the MANET split in an efficient way. The "MANET split" is defined as the situation that one part of the network is unable to connect to the other part(s) of the network. Some solution [7][8] makes use of intra-domain routing to detect the split MANET, for example some error notification in routing protocols may be applied to detect the split. But this kind of local error information is not enough to detect the MANET split, because localized link breaks do not necessarily result in MANET split according to our definition.

### 3.2.1.2 Membership Management

In BGP, each domain has its own class of hosts with IP address with a same prefix. To announce the destinations in a domain, gateways will aggregate the IP addresses in the domain by suitable IP prefix. For instance, by announcing "192.168.0.0/16", the gateway means the hosts within this domain have the same prefix of the first "16-bit" of "192.168.0.0". However, mobility and ad hoc deployment in MANET can create arbitrary network partition, which often splits the network into parts that do not have distinct prefix across the network. As Figure 3.3 shows, originally the MANET has a prefix as "192.168.1.0/30". But when the MANET splits into two parts with {192.168.1.0, 192.168.1.2} and {192.168.1.1, 192.168.1.3}, the two split parts do not have a prefix more specific than “192.168.1.0/30”, and they cannot use the original prefix since it will cause conflict in the routing table establishment. To overcome this problem, the proposed HIMR protocol lets gateways in the partitioned domain to advertise membership information in the form of membership digest, which contains both the IP address prefix and the member nodes’ IDs.

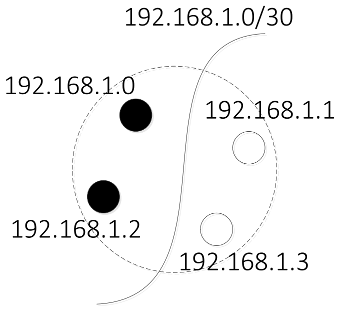


Figure 3.3 Prefix based membership management

### 3.2.1.3 Heterogeneous Intra-MANET Routing

Usually different groups in MANETs have different intra-MANET routing protocols, which makes some assumptions required by BGP inapplicable. BGP implicitly assumes the availability of the following:

1) Internal Gateway Detection: The internal gateways within the same MANET can detect the presence of each other so that they know whom to communicate with about the information of external routes.

2) Internal Network Knowledge: The gateways know the reachable destinations and the internal routes to the destinations within the MANET.

These functions are normally supported by the proactive intra-MANET routing protocols through continual maintenance of network state information. However, because of the heterogeneous intra-MANET routings, these assumptions do not always hold true. An ideal inter-

MANET routing should be independent of the underlying intra-MANET routing scheme. How to set up the required routing information in a way independent of intra-MANET routing in HIMR is described in details in the following sector.

## 3.2.2 Clustering Structure and Membership Management

### 3.2.2.1 Basic Structure: Clusters

Cluster technique is the key technique that is studied extensively in the inter-MANET routing protocol. Clusters help HIMR to obtain efficient communication among MANETs and to achieve scalability in large networks.

The proposed approach exploits the clustering by group affinity. In each MANET, the distributed clustering algorithm discovers the set of “traveling companions” – these are the nodes that stick together as a group for some time or for some common tasks. One MANET could have multiple sets which form multiple clusters. It elects within each set a gateway for each affinity group (Note that a cluster can have several gateways to obtain effective communications among MANETs). Affinity is defined in terms of some common characteristics, such as group motion or same tasks. The clusters (i.e., MANETs) are defined a priori or evolve dynamically by the affinity of geography, motion, or task. The gateway in the MANET acts as local DNS for own cluster and also neighboring clusters. The gateway advertises to neighbors and the rest of the network its connectivity, members, and MANET information (such as Autonomous System (AS) ID, etc.). The advertising protocol plays the role of BG Protocol. Moreover, gateways act as protocol translator for different PHY/MAC and routing protocol.

Note that the clustering algorithm requires periodic communications between nodes in the underlying pool nodes that are candidates to become members in the cluster. If the cluster uses a proactive routing algorithm, e.g. OLSR, the routing algorithm itself can be used for cluster creation and gateway election. In the case of on-demand routing like AODV and DSR, a separate periodic algorithm such as Distance Vector must be implemented to support the cluster functions and to propagate the gateway advertisements across the cluster.

An example scenario of inter-MANET routing is illustrated in Figure 3.4. MANETs are working above inter-MANET routing protocols (i.e. HIMR), and incompatible intra-MANET routing protocols (i.e. AODV, BELLMANFORD, etc.).

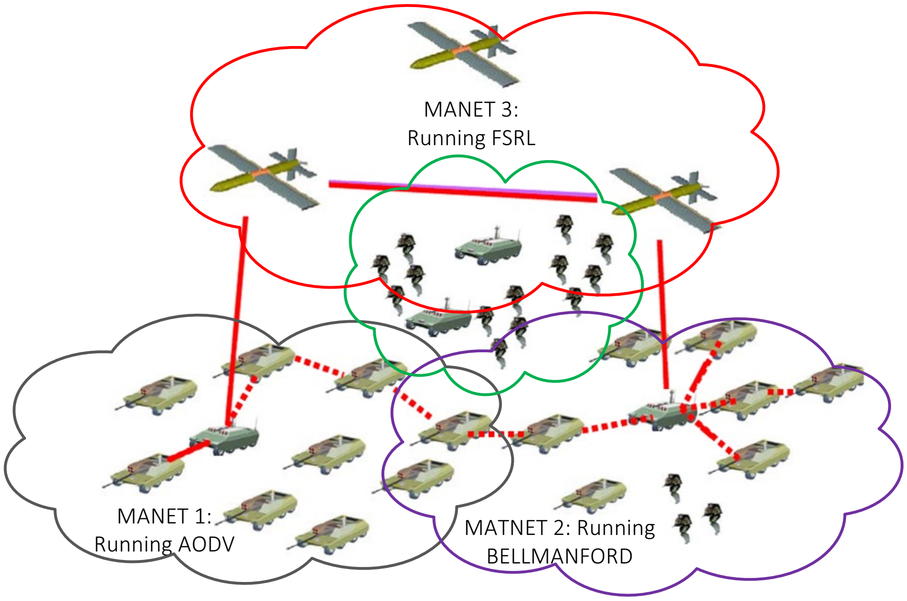


Figure 3.4 A typical scenario of inter-domain routing

In order to communicate among incompatible MANETs, gateways must have multiple network interfaces. The number of gateways in each MANET could be more than one according to the requirement of the MANETs. These gateways provide “bridges” among MANETs and are crucial for inter-MANET connections. An example of the situation of multiple interfaces is illustrated as Figure 3.5. The central node is a gateway and has three interfaces to communicate with three different MANETs so that node A in domain1 can transmit data packets to node B in domain2 and node C in domain3.

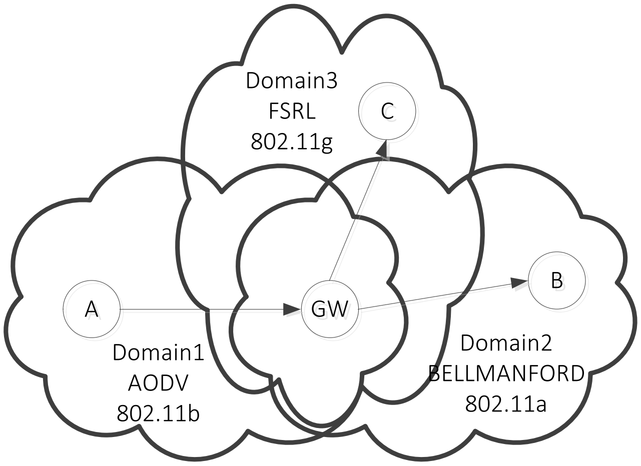


Figure 3.5 Situation of multiple interfaces and channels.

In the proposed Heterogeneous Inter-MANET Routing (HIMR) protocol, gateways can understand the messages and control packets from other MANETs. The control packet (i.e., the routing update packet) contains the topology table (including geo-locations of neighbors, etc.), the member list, and the AS ID of a MANET. The information exchanged among gateways makes it possible to efficiently communicate with other gateways in different MANETs. On the other hand, non-gateway nodes can't understand the control packets from other MANETs, but they will forward these control packets to their gateways.

Once the clusters are created and the gateways are elected, the routing is a two level operation. In the proposed protocol, packets to remote nodes are routed via gateway advertised routes, and packets to local destinations are routed using the local routing algorithm. When a source node wants to transmit a packet, the destination node ID is first searched in the source node's local routing table. The local routing protocol will be applied if the destination node is located in the same MANET as the source node. The destination node ID won't be found in the local routing table of the source node if the destination is located in a different MANET. In this case, the source node transmits the packet to its gateway using local routing protocol and local PHY/MAC. The packet travels from the source node's gateway to the destination node's gateway, using the advertised inter-MANET routing information. From the latter it is delivered to the destination node via local routing protocol and local PHY/MAC.

#### 3.2.2.2 MANET Split and Isolated Nodes

In the proposed HIMR protocol, gateways send periodic beacons to detect MANET split. If one gateway cannot hear any beacon from other gateways within the same MANET, HIMR considers the MANET as partitioned into disconnected components. A regular (non-gateway) node within the MANET will respond with an acknowledge message upon receiving the first beacon message from one of the gateways in the same MANET. If regular nodes do not receive any beacon message within a timeout threshold, they will consider themselves as isolated nodes and trigger the new gateway election algorithm to elect a new gateway within these isolated nodes.

Either MANET split or isolated nodes will trigger the birth of a new MANET. A new AS ID needs to be generated. The member digest information and the timestamp of the new MANET are fed into a pseudo random function, which will generate a new AS-ID for the new-born MANET. The new AS-ID is guaranteed to be different from existing AS-IDs.

In the new-born MANET, new gateways need to be elected. Each node in the MANET triggers gateway-election algorithm to elect a new gateway. Since a gateway with better connectivity is preferred, the gateway-election algorithm will elevate the node with the most neighbors to gateway rank. Using a neighbor discovery scheme, each node will know its number of neighbors, and broadcast this number to its neighbor. If it does not receive a neighbor count greater than its own neighbor count, it will elect itself as the gateway. Note that the above gateway-election algorithm is also applied to gateway-election processes in existing domains.

With the birth of each new MANET, the routing path needs to be updated in the routing tables of gateways in existing MANETs. Upon receiving the advertised update control packet from new-born MANET, the gateways in existing MANETs will update their routing tables and membership information for the new-born MANET. The HIMR protocol carefully adds MANET information in order to prevent MANET path loop. As long as the received MANET ID is the same as the MANET ID on the existing MANET path, the advertised MANET ID will be abandoned.

#### 3.2.2.3 Member Digest with Bloom Filter for Membership Management

The membership management in an inter-MANET routing among MANETs is a challenge. The prefix based routing of BGP does not work since gateways are not able to aggregate MANET members by suitable IP prefix. Global gateways coordinate and reassign node IDs so that each node has a unique prefix, which is not feasible either. In the proposed HIMR protocol, the MANET membership information is advertised in the form of membership digest. The advertised control packet broadcast by the gateway node contains the member digest of that MANET.

Using a plain member list in the control packet by a gateway is costly when the network becomes large. Bloom Filter is the technique to map a member list to a bit vector, in which the membership verification operation can be carried out within O(1) operations instead of O(m)O(m) operations (m is the member count) required by the plain member list. When a Bloom Filter is used to represent the member list of a cluster, the size of the control packet advertised by gateway is much smaller than the size of the conventional control packet with full (plain) member list. Thus the proposed HIMR protocol becomes more scalable by taking advantage of the Bloom Filter.

Figure 3.6 shows the construction of the Bloom Filter according to a plain member list. A bit vector of m bits is used to represent a set of n members $\{id_{1,} id_2, \ldots, id_n\}$. Originally all the bits in the Bloom Filter are set to “0”. By hashing each item using a hash function of $\log_2(m)\log_2(m)$ bits, the Bloom Filter will set the corresponding bit to 1. To check the membership of the element x, it is sufficient to verify whether the bit corresponding to $h(x)h(x)$ is set to “1”. The verification will cause “false positive”, i.e., an element not belonging to the set may be checked as a member. But Bloom Filter is free from false negatives, i.e., any element verified as a non-member shall not belong to the set. Many hash functions such as MD5 and SHA-1 are evenly distributed in the “bit vector” domain, so the false positive probability can be decreased to a large extent.

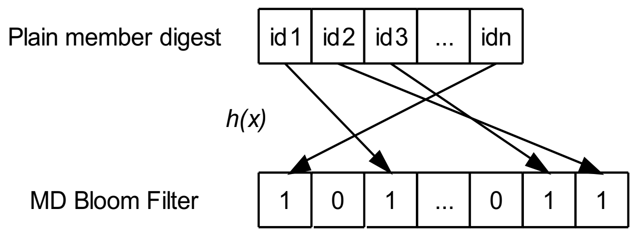


Figure 3.6 Using Boom Filter to compress the Member List

## 3.2.3 Core Routing Components

### 3.2.3.1 Geo-DFR

Geo-DFR [31] is of particular interest in multi-MANET scenarios, where a cluster head is elected in each MANET and propagates advertisements to the other MANETs. HIMR uses Geo-DFR (Greedy Forwarding + Direction Forwarding) as its core components to route among MANETs. The packet travels from the source node’s cluster head to the destination node’s cluster head by using Geo-DFR. From the latter it is delivered to the destination node via local routing protocol.

Geo-DFR is a geographical based routing scheme. The key idea of Geo-routing [32] is known that each node knows its geo-coordinates either from GPS or Galileo, and the source knows the destination geo-coordinates and stamps it in the packet. At each hop, the packet is forwarded to the neighbor closest to destination. Some forwarding schemes are used in Geo-routing, such as Greedy forwarding, Perimeter forwarding and Direction forwarding. In Geo-DFR, direction forwarding is designed to complement and even replace Perimeter forwarding in dead end recovery. A packet in Geo-DFR is first forwarded to the neighbor which yields the most progress towards the destination, i.e., greedy forwarding. If greedy forwarding fails, the packet is "directionally" forwarded to the "most promising" node along the advertised direction.

Figure 3.7 is the comparison between geo-routing (i.e., Greedy Forwarding here) and Geo-DFR. It also shows how direction forwarding in Geo-DFR helps packets to detour from a "hole", i.e., an obstacle. The upper part of Figure 3.7 illustrates the Geo-routing using greedy forwarding. When the node's routing run into a "hole", the greedy forwarding terminated, and the routing path fails because of the "hole". The lower part of Figure 3.7 shows the functionality of Geo-DFR. Each of nodes in Geo-DFR calculates the direction to the destination. If the greedy forwarding fails because of the "hole", the backup direction forwarding in Geo-DFR will be used to select the next hop for the further forwarding packets.

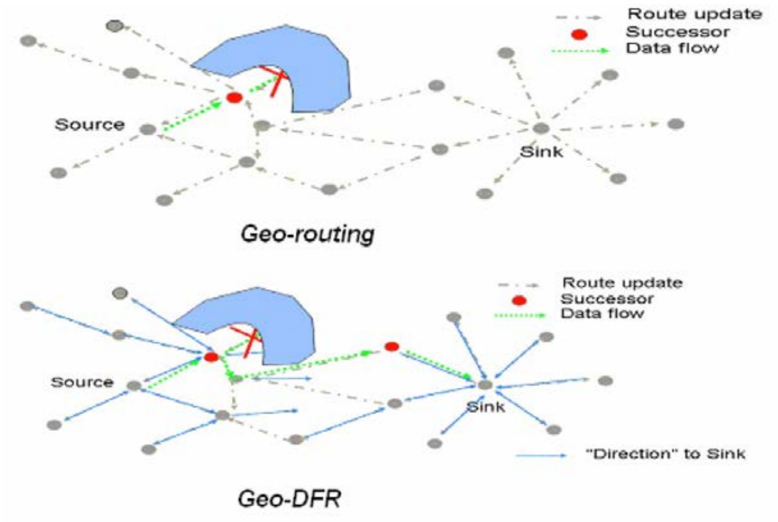


Figure 3.7 Comparison between Geo-routing and Geo-DFR.

The direction forwarding in Geo-DFR chooses the next hop based on the direction of each node to the destination which is calculated when the routing update received. Also, if multiple updates are received at the same time from different neighbors with same hop distance and sequence number, the direction will be calculated by the vector sum of directions. Figure 3.8 gives an example to illustrate the computing of the direction. Suppose Node A receives direction update packets from Node B and Node C, the direction to the destination is the vector sum of direction from Node A to Node B and the direction from Node A to Node C. It is marked with red color in Figure 3.8.

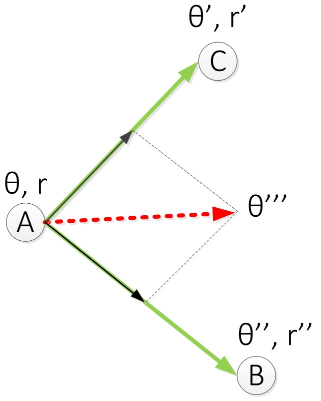


Figure 3.8 Computing the direction to the destination.

By using Geo-DFR, each node remembers the "direction" on the way to each cluster head in the same MANET. The node knows which zone can be reachable from the cluster head. Among MANETs, cluster heads perform the Geo-DFR protocol to find an adoptive way to the destination as shown in Figure 3.9. Within the MANET, an intra-MANET routing protocol such as DSDV, AODV is used. The HIMR inter-MANET routing protocol chooses the routing path which is marked by red lines from S to D in Figure 3.9.

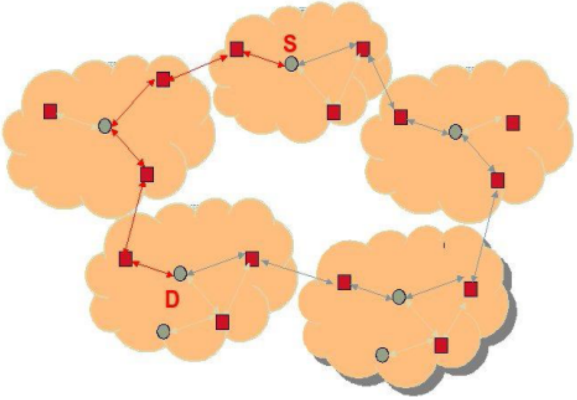


Figure 3.9 Scenario of HIMR.

### 3.2.3.2 Fisheye Scheme for Broadcast

A gateway node will broadcast two kinds of control packets: intra-MANET control packet and inter-MANET control packet, whereas a non-gateway node will only broadcast intra-MANET control packet. The inter-MANET control packet contains information update about gateway's topology table, member digest in a cluster, MANET ID of the cluster, and Distance Vector (DV) routing update among gateways, etc. The intra-MANET control packet contains information update about a node's connectivity, neighborhood change, and local DV routing update within a cluster.

A Fisheye like algorithm is applied in the broadcast of intra-MANET control packets within a MANET and inter-MANET control packets cross MANETs. In the HIMR protocol, the value of Time To Live (TTL) is used in order to limit the spatial propagation of the update control packet, and the transmission is differentiated in time. At the beginning, the TTL value is set to a specific value that is a function of the current time. After one update control packet transmission, a node wakes up every $t_e$ seconds (observation time) and sends a control packet with TTL set to $s_1$ (scope within one hop) if there has been a change during the last $t_e$ seconds. The change can be topology change, neighborhood change, DV routing update, membership change, or MANET information change, etc. The node wakes up every $2t_e$ seconds and transmits an update control packet with TTL set to $s_2$ (scope within two hop) if there has been a change during the last e $2t_e$ seconds. In general, an update control packet is transmitted with TTL set to $s_i$ (scope within i hops) if there has been a change during the last $2^{i-1}t_e$ seconds.

By differentiating the update rate of control packet in space and over time, the above FSR-like scheme efficiently reduces the control overhead of control packet updates and offers good scalability properties for the proposed HIMR protocol.

## 3.3 Evaluation

HIMR is implemented under Qualnet network simulator 4.5.1 [33]. Network data traffic is generated by CBR sources. Packet size is 512 bytes and packet interval is 0.25s. The source-destination pairs are randomly selected. The dimension of the network scenario is 1000m×1000m. Different seeds are used in the simulations.

The mobility model is RPGM [34]. Each node in a MANET has a common group motion component. In addition, each node has an individual intra-group motion component. In our simulation the group speed varies under different scenarios, while the intra-group speed is fixed in the range of [0-5 m/s] and the pause time is 10 seconds. Total simulation time is 1000 seconds.

PHY/MAC protocol is IEEE 802.11b, which use CSMA/CA with RTS/CTS, and radio range of 375m.

### 3.3.1 Under Different Gateway Percentages

The performance of HIMR is tested under scenarios that have different percentage of gateways and different number of MANETs. The percentage of gateway is the ratio of number of gateway over the total node number. Total nodes in network are 60.

The delivery ratio in the scenarios of different percentages of gateway is illustrated in Figure 3.10. It has been shown that the delivery ratio becomes higher when the percentage of gateway increases. More gateways result in higher delivery ratio. Every node needs to send its packets to gateways when transferring packets to nodes in other MANET. Gateway functions as the communicator between MANETs. Insufficient gateways reduce the connectivity of the network and thus easier for packets to be dropped. So by increasing of the number of gateway, the delivery ratio becomes better. The packet delivery ratio drops when the number of MANET increases. When the packet transfers across more MANETs, the possibility of packet loss increase since the packet needs to be routed by the "communicator" of gateway and the number of gateways in each MANET is limited.

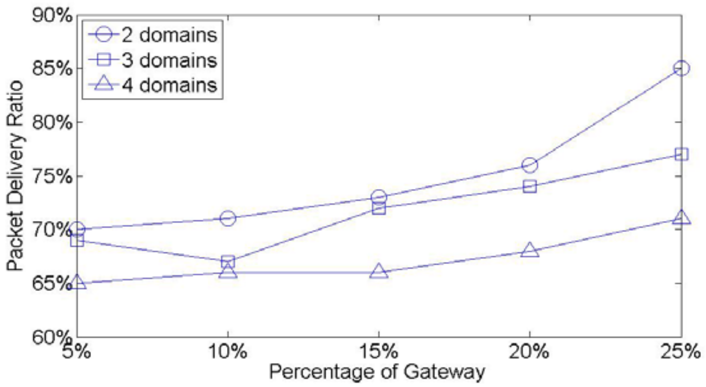


Figure 3.10 Packet Delivery Ratio vs. Percentage of Gateway

### 3.3.2 Benefit of Bloom Filter

The effectiveness of bloom filter is based on member digest scheme. In our simulation, 800-bit, 1200-bit and 1600-bit bloom filters are tested respectively as a hash table for the member digest. The bloom filter always compresses the member digest for each domain into its hash table. For example, to 800-bit bloom filter, since each node address structure in Qualnet is 32-bit, this bloom filter will introduce more overhead than a plain member digest when node number in each domain is less than 25 (e.g., 800/32). When the node number is greater than 25, bloom filter will help to decrease the control overhead in the proposed HIMR protocol. Figure 3.11 clearly indicates this behavior. The X-Axis is the number of nodes in each MANET, and the Y-Axis is the control overhead reduction achieved by using a bloom filter compared to a plain member digest (i.e., the overhead of using plain member digest – the overhead of using bloom filter). When the node number in one MANET is 20, this reduction is negative, meaning that the bloom filter introduces more overhead than plain member digest. When the node number in one MANET is equal to or greater than 40, the bloom filter helps to alleviate the routing control overhead. The 1200-bit bloom filter helps to alleviate the routing control overhead when the node number in a MANET is equal to or greater than 40, and the 1600-bit bloom filter helps to alleviate the routing control overhead when the node number in a MANET is equal to or greater than 60. The same trends are found for different bit bloom filter: more nodes in MANET, the benefit of bloom filter reducing control overhead is more obvious.

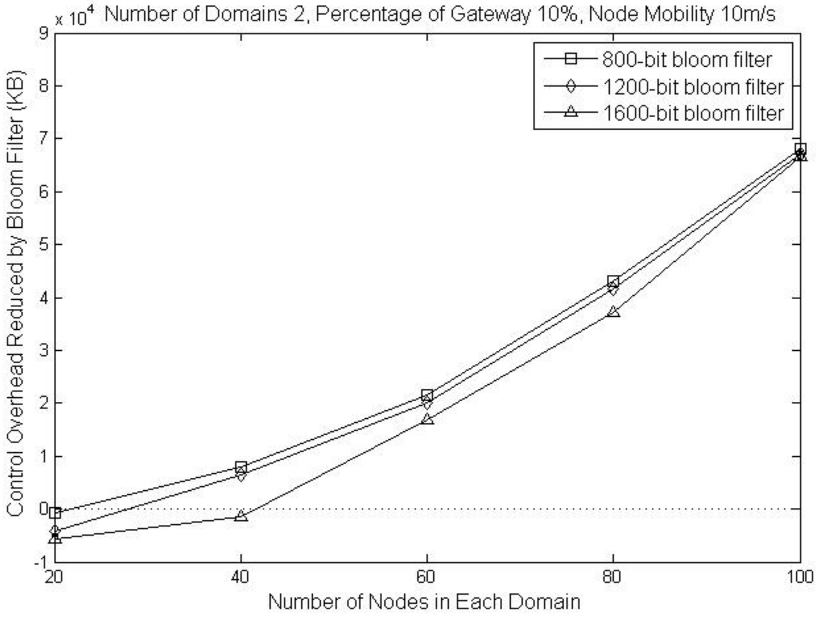


Figure 3.11 Benefit of Bloom Filter on Control Overhead

# Chapter 4. Energy-Efficient Content Retrieval in MANETs

In this section, we first give the common assumptions which drive some of the design decisions of the protocols. After that, we describe the design of our content retrieval scheme in detail.

## 4.1 Assumptions

In this charpter, the following assumptions are made.

1) One of the primary factors considered in content retrieval scheme is the geo-location of the various nodes such as the sender, receiver and the relay nodes. We assume Global Positioning System (GPS) infrastructure would be available, which would help us to get the coordinates associated with a particular node at a given point of time.

2) Also we assume the topology under consideration is bi-directional thus facilitating two way communications.

3) In the given topology, node-id is used as the unique identifier for a given node in the mobile network.

4) The hierarchical naming scheme in ICN is used in mobile network to identify the content data carried by nodes.

5) We follow the current design of the content caching scheme in mobile ICN [14][15].

6) Each node starts with a unique content, and requests are made randomly for any of these contents from any of the nodes.

## 4.2 System Initialization

To initialize the system, each node in a network of size $N$ duplicates its content list through a random walk of size $M$ starting from itself, and setups a freshness timer. These copies stored in the caches of the nodes during random walk can decrease the forwarding times of the Interest packet before hitting the content to save the energy. The factor $M$ depends on the topology of the network. Generally, it is related to the network size, i.e., $M = f(N)$ where $N$ is the number of nodes in the network. After the freshness timer runs out, the content provider will refresh the $M$ duplications for any data update. The freshness timer duration depends on the application requirement. After the content list replication is complete, to start a request, an Interest packet is generated by the user and implanted through a random walk of size $M$ starting from the requester. After that, each node that has the same Interest packet starts parallel process of the direction-selective dissemination described in section 3.4. We analyze the value of $M$ in section 4 and give the simulation result in section 5.

## 4.3 Packet Format

In this charpter, four types of packet formats are used, as shown in Table 4.1.

Table 4.1 Packet types

| Packet type | Description |
|---|---|
| INT | Interest packet |
| ACK | Acknowledgement packet |
| CMD | Command packet |
| CNT | Content data packet |

1) INT packet format

The INT packet is used by the sender node to express a request for content by the Interest name. The Interest packet is forwarded by the relay node after appending the path history with its own node-id and geo-location, as shown in (4.1). The path history is a list of node-id, which records each hop along the transmission. Before sending out the Interest packet, the relay node adds its node-id in the path history.

$$\text{INT: \{interest\_name, geo-location, path\_history<id1, id2 …>\}} \quad (4.1)$$

2) ACK packet format

The ACK packet is used by the Interest receivers to indicate acknowledgement and send other information back to the Interest packet sender (either the original requester or a relay node). The information incorporated into the ACK packet helps the Interest packet sender to choose an appropriate relay node for the next round forwarding. The information added by the Interest receiver includes its node-id, geo-location, and the duplicate count of the Interest packet that the node has received expressing the same Interest, as shown in (4.2).

$$\text{ACK: \{node-id, geo-location, dup\_INT\_\#\}} \quad (4.2)$$

3) CMD packet format

The CMD packet is used by the Interest packet sender to notify a chosen node to forward the Interest packet in the next round. The next relay node would be decided based on the information received from the ACK packets. The CMD packet is broadcasted to one hop neighbors and carries a list of relay node-id, as shown in (4.3).

$$\text{CMD: \{relay\_node\_list\}} \quad (4.3)$$

4) CNT packet format

The content packet is used to reply the Interest packet and carry the content data to the requester. A reversed path history is attached into the content packet, which indicates the return path to the requester, as shown in (4.4).

CNT: {content_data, reversed_path_history<… id2, id1>} (4.4)

## 4.4 Direction-Selective Dissemination

The direction-selective dissemination for content query in MANET can be described as an iterative approach in which the Interest packet is forwarded one hop at a time until it hits the content in cache server or content provider. The dissemination process consists of four steps as described in pseudo code 4.1.

Pseudo code 4.1 Selective Directional Dissemination

```
Step 1:
INT packet is generated.
Add sender's id into INT.
Divide the area into four quadrants.
Send out the INT.
Step 2:
Receive and cache the INT.
Count the duplicate INT #.
Check local content repository.
if (found) then
    Reply with CNT.
else
    Setup back-off time.
    Send ACK back to sender.
end if
Step 3:
Receive the ACK.
Select one relay node in each quadrant.
Send out CMD with relay_node_list.
Step 4:
Receive CMD.
Decide if forward or not.
Repeat from step 1.
Note: don't select relay node in the quadrant where the last forwarding node exists.
```

In the first step, the Interest sender initially divides its surrounding space into 4 quadrants, as shown in Figure 4.1(a). Four quadrants are divided by north/south/east/west directions. The sender A generates the Interest packet and adds its own node-id and geo- location into the path_history field, then broadcasts the Interest packet to all the one hop neighbors (as shown as yellow nodes in Figure 4.1(b)) in its transmission range.

In the second step, each Interest recipient stores the received Interest packet into its local cache and also maintains a mapping between Interest name and the corresponding duplicate count with the same Interest it ever received before (starts with 0).  Once the node gets an Interest packet, the count corresponding to the Interest name is incremented. The node then checks its local content repository to see whether the content corresponding to the Interest is available. If no such content is available, the node will send back an ACK packet back to the sender, as shown in Figure 4.1(c).

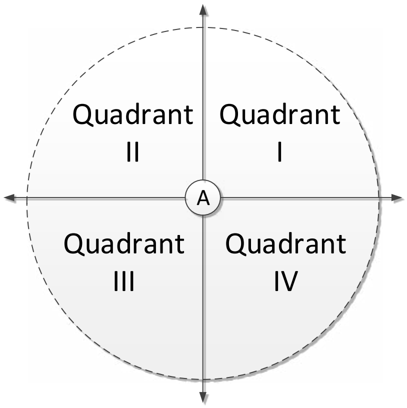


Figure 4.1 (a) Node A’s four quadrants

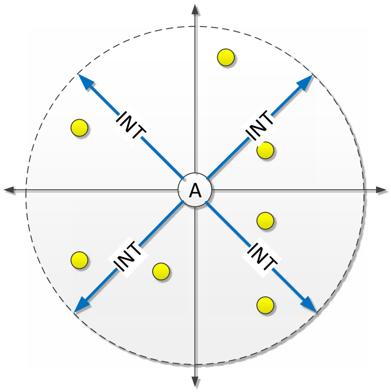


Figure 4.1 (b) Node A broadcasts Interest to neighbors

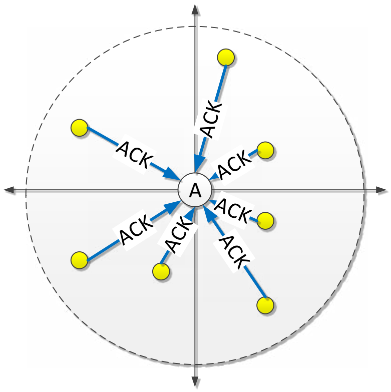


Figure 4.1 (c) Recipients reply ACK

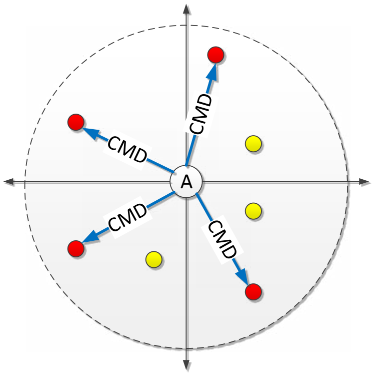


Figure 4.1 (d)  Node A assigns relay node

In order to avoid collisions at the sender, each recipient setup a back-off timer before sending ACK, where the waiting time $t_{wait}$ is proportional to the distance between the sender and itself, as shown in (4.5).

$$t_{wait} = t_{max}(1 - \frac{d}{r_{trans}}) \tag{4.5}$$

In (4.5), $r_{trans}$ is the maximum transmission range. $t_{max}$ is selected by a formula exponentially biased towards nodes farther away from the sender.

In the third step, after receiving the ACKs, the Interest sender chooses one node in each of its 4 quadrants as relay node. This decision is based upon the geo-location and the duplicate Interest count in the ACK packet. The node chosen in a quadrant would have the farthest distance to the sender and a value of 1 for the duplicate Interest count. This is because, farther the distance, we would be able to cover more area with respect to the sender node. Also a higher value of duplicate count indicates that the position of this recipient is covered by multiple relay nodes thus being a bad candidate in our effort to avoid increased number of Interest packet copies in the network. Once the sender has chosen relay nodes in each of its 4 quadrants, the sender would then add these selected relay node-ids into the relay node list attached in the CMD packet and broadcasts CMD packet to its one hop neighbors, as shown in Figure 4.1(d), where the red nodes are the selected relay node by the sender. The recipient nodes perform a lookup on the relay node list presented in the received CMD packet. If a node determines its id is presented in this list, it would then take up the responsibility of forwarding the Interest packet cached previously.

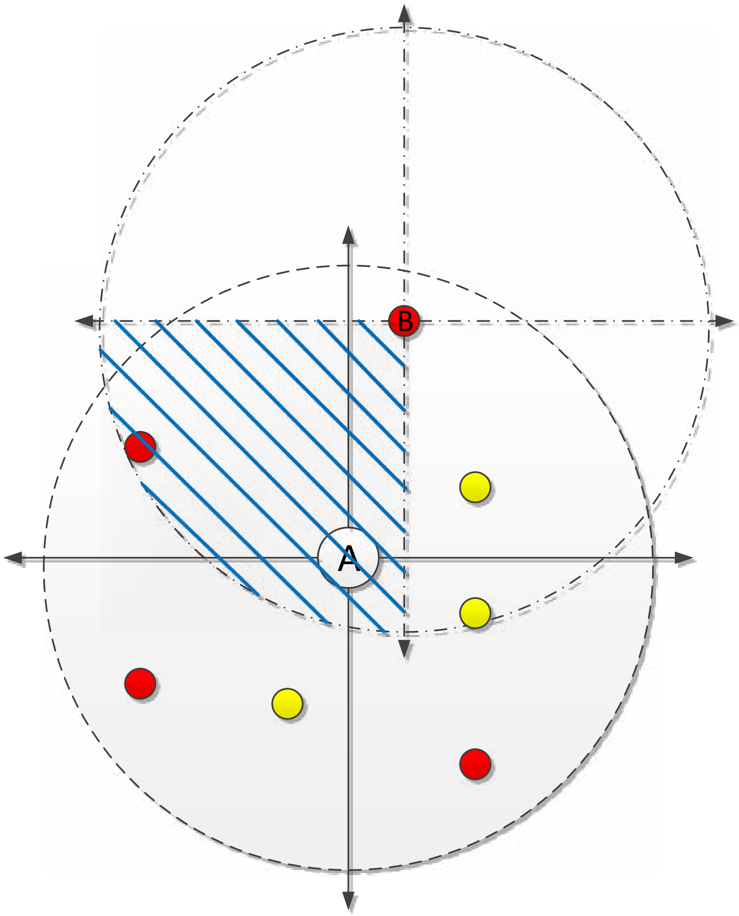


Figure 4.2 Node B relays the Interest dissemination

In the fourth step, the relay node adds its node-id into the path_history field of the Interest packet, and then again forwards the Interest packet in the same method: the sender in this round performs a direction-selective dissemination again. However, this time the relay node only chooses the future relay nodes in the next round from the other three quadrants expect the quadrant where the last forwarding node exists. For example, as shown in Figure 4.2, after the third step, node B is chosen by node A as its relay node. Then node B will repeats from the first step to the third step, but only choose the next round relay node from its quadrants I, II, and IV, expect III, since the quadrant III is the most overlap area by node A in the last round compared to other three quadrants. Also note that, since we only choose the relay node whose duplicate Interest packet count equals to one, we can avoid the formation of loops in the path history. From Figure 4.2 we can see that, this Interest dissemination scheme doesn’t guarantee that every node in the network will receive at least one copy of Interest. But the Interest packet propagation can cover most of the giant components and high degree nodes in the network, as discussed in section 4 and shown in experiment results.

## 4.5 Content Data Reply and Dissemination Termination

Once the Interest packet hits the content provider or the content cache server, the content data will be transmitted back to the requester. The path_history in the received Interest packet is reversed and added into the content data packet in order to enable the nodes along the path to route the content data packets back to the requester.

Obviously, if a node has the content and reply the content data back to the requester, it will stop the Interest dissemination process. In another case, if a node is processing the Interest dissemination process, once it hears the content data replied to the Interest by other nodes, the step 3 will be terminated immediately so that no more relay nodes will be selected in this direction and the Interest dissemination is terminated at this node. This method will further decrease the traffic overhead in the content request phase.

## 4.6 System Analysis

By taking a short random walk through the network, we will reach a high degree node with higher probability [35][36]. In our content retrieval scheme, the initialization phase of the proposed direction-selective Interest dissemination method is that each content consumer starts random walk with $M$ steps to make $M$ copies of the Interest packet at the beginning, which is the same as the content provider who makes $M$ copies by random walk. We then run the search process in parallel. The random walk only works in the initialization phase to make $M$ copies for both Interest and content data. When the layout is complete, the direction-selective dissemination will take the responsibility to forward the Interest packet. In this way, the Interest packet in the proposed method will hit the content provider with higher probability and reduce the querying overhead significantly compared to the current existing methods in [16][17][18][19]. We will show the experiment results of different values of $M$ in section 4.7.4.

A major limitation of this method is that it cannot guarantee the discovery of content as 100% confidence. We must provide a fallback solution just in case that the single content is not covered by the Interest dissemination. We may choose to setup a TTL in Interest packet. The value of TTL depends on the network scale. TTL will decrease by one after each relay. If a relay nodes reaches TTL=0, it simply reply one alert message to the requester. Then the requester will turn into the traditional mode to use the intelligent broadcast for content query which leads to more traffic overhead but will guarantee 100% successful rate.

## 4.7 Evaluation

In this section, we evaluate the performance of the proposed content retrieval scheme in a packet-level simulation with the real mobility data set.

### 4.7.1 Simulation Setup

We implemented the proposed scheme in NS-2 v2.35 network simulator. The proposed energy-efficient content retrieval protocol is installed directly above the 802.11b devices. We performed the evaluation with the mobility pattern in the Dartmouth outdoor MANET experiment data set [37]. We converted the mobility trace to a 2000*2000m2 dimensions with 120 nodes. The results are averaged from 20 runs with different seeds with presented 95% confidential interval.

The performance is evaluated under: 1) total cost: the total number of packets that have been generated during a test. This metric reflects the overall cost of a method in the discovery of requested files; 2) prune ratio: the percentage of packets which are dropped and not forwarded over the total packets. This metric indicates the extent of messages which are not used by the nodes to propagate further; experiments 1) and 2) are conducted under $M$ =10; 3) hit rate: the percentage of Interests that are successfully delivered to the matching content provider or cache server. This metric reflects the capability of a method to discover the requested contents.

We compared our design against the methods of naive flooding and probabilistic forwarding.

### 4.7.2 Total Cost

The total cost calculates the number of all messages involved in the Interest propagation and content retrieval. Messages like ACK and CMD are new additions in our design as compared to the traditional methods. As shown in Figure 4.3, the naive flooding method causes the largest total cost. While the probabilistic forwarding can significantly reduce the total cast. Our design can further decrease the total cast than all other methods.

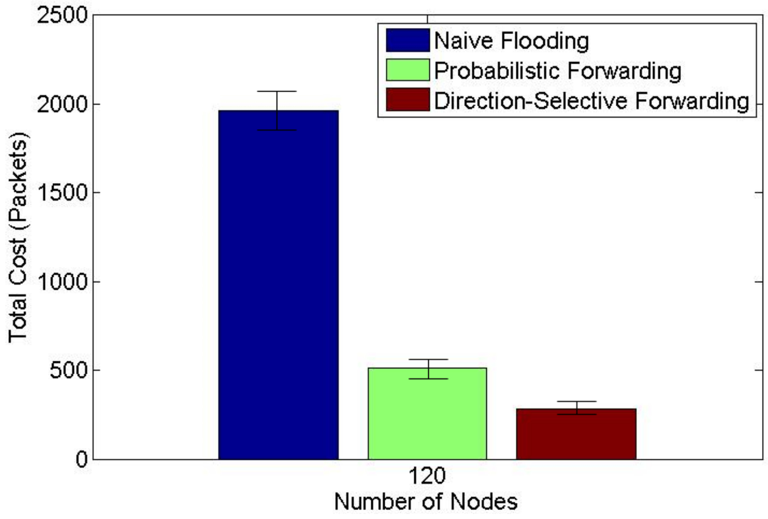


Figure 4.3 Total cost

### 4.7.3 Prune Ratio

In the propagation of the Interest in the network, the messages are progressively spread through the entire network. During this process, the nodes drop the received Interest packets if they are not selected as relay nodes. As shown in Figure 4.4, it is obvious that both the probabilistic forwarding and our direction-selective forwarding have a much higher prune ratio than the naive flooding method. And our scheme can prune even more duplicate Interest packets than probabilistic forwarding since the relay nodes are selected hop by hop based on the overlap condition, and Interest packet is propagated directionally.

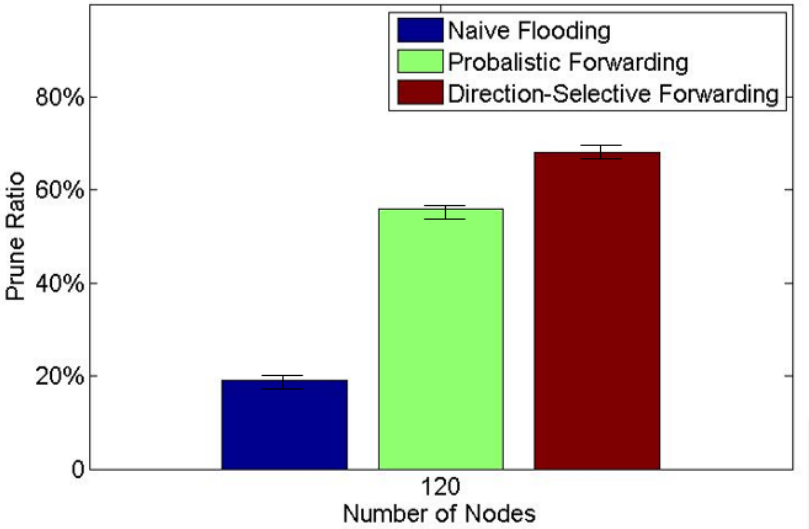


Figure 4.4 Prune ratio

### 4.7.4 Hit Rate

Hit rate is an indication of the successful delivery. As shown in Figure 4.5, the naive flooding method reaches the 100% hit rate since it floods Interest packet in the entire network. We test the parameter $M$: copies of the Interest and content data in the initialization phase of our direction-selective forwarding scheme. We can see from the results that, with $M = 10$, it can reach almost

96% hit. And with the lower $M$ value, the hit rate of our scheme decreases rapidly. $M = 0$ means that in the initialization phase the node does not replicate Interest and content data by random walk, but keeps only one unique copy in the content provider. In this case, the hit rate decreases even as low as about 64. It is because that, with the lower number of copies, it is more possible to miss the content data existing in the gap area which is not covered by the direction-selective dissemination scheme. And with the large node density in the network, more nodes exist in the gap area so that the contents on those nodes cannot be covered.

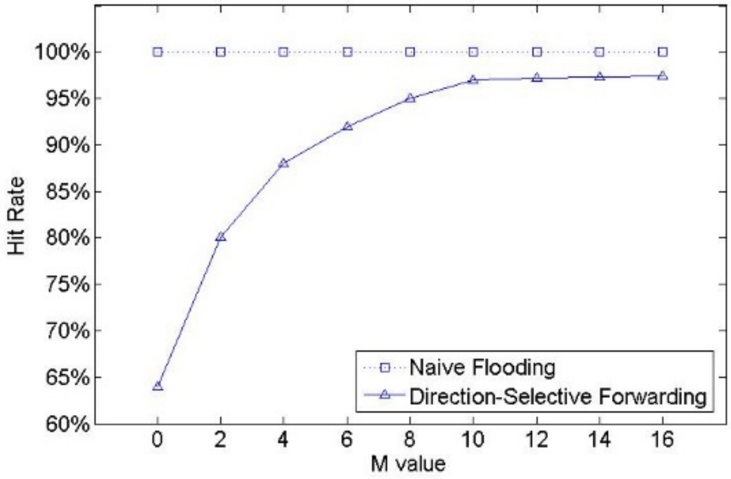


Figure 4.5 Hit rate with different M value

# Chapter 5. Information-Centric Content Retrieval in Disruption-Tolerant Networks

In this section, we first give the common assumptions which drive some of design decisions of the protocols. After that, we describe the design of our Social-Tie based Content Retrieval (STCR) scheme in detail.

## 5.1 Assumptions

In this chapter, the following assumptions are made.

1) We assume the connection associated with each encounter is bi-directional thus facilitating two way communications during the period of encounter

2) In the given topology, node-id is used as the unique identifier for a given node.

3) We follow the naming scheme in NDN [2].

4) Requests can be made randomly for any of contents from any of nodes.

## 5.2 Preliminaries

In this section, we provide the basic definitions for our protocol design. A sparse MANET is a network where the node density is not high enough to instantly connect all the nodes. Thus, packets must be forwarded in carry-and-forward mode if necessary. When there is no path from source to destination, a node will hold the packet until it encounters another node that has a higher possibility than itself to deliver the packet to the destination. This delivery mechanism causes a significant delay from sender to receiver, and is only suitable for applications with no real-time requirements.

To support the content retrieval in sparse MANETs, STCR performs the following main operations.

1) Advertise Hello message

Each node periodically advertises Hello messages for discovery of encounters in its transmission range. Hello message contains sender's node-id. Hello transmission interval is 100ms.

2) Record encounter event

A data structure called encounter-vector which includes encountered node-id and timestamp of this encounter event, as shown in (5.1), will be created after each encounter event:

$$\langle \text{node-id, timestamp} \rangle \tag{5.1}$$

Every node maintains an encounter-table which stores encounter-vectors created by the node at the encounter time.

3) Compute social relationship

The social relationship is the results of the aggregation of several indicators. Previous proposals have included metrics such as online social network graph [27], mutual friends [26], community label, friends connection duration [29], etc. All of them can reflect the social relationship in some aspects. In human networks, intuitively, if I have a message for a certain receiver, I will try to find a relay person who knows more people than I, since a more sociable person has a higher probability to see the receiver in the near future. This is the reason why all previous studies have built a social hierarchy based on the centrality relationship stemming from either historical contacts or human social graph. The higher centrality represents a higher average encounter probability to all other nodes. However, it is natural to assume that if two people have met frequently in the recent past, they must be in a close relationship. This recent encounter history can be used to predict the near future. BubbleRap [29] has shown that past contact information can be used to accurately predict future contacts. Frequency and recency become then the fundamental factors which must be monitored to build other metrics, e.g., social graph and mutual friends. Inspired by SimBetTS [26], we use frequency and freshness to describe social relationship.

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. 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.

Inspired from LRFU [38], each node computes a social-tie value to evaluate its relationship with other node and prioritize those relationships. As discussed earlier, the social-tie value is derived from the encounter history. 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 based on $n$ encounter events of node $i$, the social-tie value of node $i$ relationship with node $j$ at time $t_{base}$, denote by $R_i(j)$, is defined as (5.2).

$$R_i(j) = \sum_{k=1}^{n} F(t_{base} - t_{j_k}) \tag{5.2}$$

where $F(x)$ is a weighing function and $\{t_{j_1}, t_{j_2}, \cdots, t_{j_n}\}$ are the encounter time when node i met node $j$ and $t_{j_1} < t_{j_2} < \cdots < t_{j_n} \leq t_{base}$.

For example, assume that node i met node j at time 1, 3, and 5 and the current time ($t_{base}$) is 10. Then node i 's social-tie to node j at $t_{base}$, denoted by $R_i(j)$, is computed as

$$R_i(j) = F(10-1) + F(10-3) + F(10-5) = F(9) + F(7) + F(5)$$

We take $F(x) = (\frac{1}{2})^{\lambda x}$ where $\lambda$ is a control parameter and $0 \leq \lambda \leq 1$ as the weighing function, which have been proved in [38]. First, this control parameter allows a trade-off between freshness and frequency in contributing to the social-tie value. As $\lambda$ approaches 0, frequency contributes more than freshness. When $\lambda$ equals to 0, the social-tie value is simply derived from frequency. On the other hand, as $\lambda$ approaches 1, freshness has much more effects on the social-tie value than frequency. When $\lambda$ equals to 1, the social-tie value is simply determined by freshness. Following the example in [38], we set $\lambda = 1e^{-4}$. Second, suppose that node $i$ has $n$ encounter events with node $j$, the social-tie value at the time of $nth$ encounter event can be computed from the time of the $(n-1)th$ encounter event and the social-tie value at that time. The computational and storage overhead can be reduced drastically due to this feature and each node is not required to maintain the record of all the past encounter events. Third, as discussed before, for each node, the relationships with other nodes should be prioritized or ordered according to the social-tie values. Due to the property of social-tie value definition and the feature of the above weighing function, the order of the social-tie values does not change until a new encounter event occurs, i.e., the order of the social-tie values changes only if there is a new encounter event. Hence, reordering of the social-tie values is needed only upon a new encounter event, though the social-tie values change over time.

4) Exchange social-tie table

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. This process is similar to a routing update in link-state routing protocol. When a node receives a social-tie table from other nodes, it will refresh the local social-tie table according to the timestamps. Eventually, a social-tie table in a node will contain all the nodes' social-tie in network, but social-tie table convergence progress is very slow due to the long latency feature of DTN.

5) Compute centrality

Based on the social-tie table, a node can compute each node's centrality. Centrality measures the average social distance from the given node to all other encountered nodes. The centrality can be regarded as a measure of how long it will take information to spread from a given node to all other nodes in the delay-tolerant network. It is obvious that the average social-tie from the given node to all other nodes can be computed as (5.3), where N is the number of nodes observed from social-tie table, and R is the social-tie from the given node to each of other nodes.

$$\frac{\sum_{k=1}^{N} R_i(k)}{N} \tag{5.3}$$

However, in some case a node may have an unevenly distributed social-ties to other encountered nodes. This may also cause a relatively high average social-tie result. Obviously, the average social distance we preferred in network depends not only on the average social-tie values, but also on its social-tie's distribution. In delay-tolerant MANETs, the connectivity of the networks is very dynamic. The high centrality node should not only have a high average social-tie value, but also have a high chance to encounter more other nodes. Accordingly, the social-tie distribution should be balanced. We propose a centrality estimation method that considers both the average social-tie values and their distribution to achieve higher degree of centrality. We adopt Jain's Fairness Index mechanism [30] to evaluate the balance distribution of social-tie values. As in equation (5.4), Jain's Fairness Index is used to determine whether users or applications are receiving a fair share of network resources.

$$Jain's\ Fairness\ Index{:}\ balance = \frac{(\sum x_i)^2}{n \times \sum {x_i}^2} \tag{5.4}$$

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 (5.5), where N is the encountered node count in the encounter table.

$$C_i = \alpha \frac{\sum_{k=1}^{N} R_i(k)}{N} + (1-\alpha) \frac{(\sum_{k=1}^{N} R_i(k))^2}{N \times \sum_{k=1}^{N} \left(R_i(k)\right)^2} \tag{5.5}$$

Here α (set in our experiment as 0.5) is a parameter decided by the user according to the specific scenario and network conditions. For example, if there are few nodes in a large area with high mobility, we prefer a smaller α since here the balanced connection opportunity between nodes is more important. Otherwise, if more nodes exist in a relatively small ground, we may consider a bigger α value. A higher centrality value means that the node has been meeting other nodes more often and more recent in network.

## 5.3 Content Name Digest

Content management in content retrieval application is a challenge. Using a plain content name list is costly and not scalable when the network and number of content become large. In our proposed scheme, the content names are advertised in the form of content name digest. When a Bloom Filter is used to represent the content name list of a node, the size of the digest advertised by the node is much smaller than plain content names. Thus, our proposed scheme becomes more scalable.

## 5.4 Digest Convergence

A node can compute each observed node’s centrality from social-tie table and form a sequence of social relationship among the observed nodes in network. The higher centrality node has a higher probability to meet other nodes than the lower level node. In the content query phase, in order to avoid pure flooding, we design this 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, as shown in table 5.1, to store the received digests from lower centrality nodes.

Table 5.1. Digest table

| Provider ID | Digest | Timestamp |
|---|---|---|
| Node-2 | Digest-2 | Time-2 |
| Node-3 | Digest-3 | Time-3 |
| … | … | … |

And then, the digest table will be sent to another higher centrality node encountered later, as shown in Pseudo code 5.1. If received multiple copies of content name digest of 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 towards the higher centrality nodes in network. The higher centrality node has the larger knowledge of the content name digests, and knows which content provider contains which content.

Pseudo code 5.1: Digest Convergence Process

```
when receiving a content digest from other node
    insert this digest into my local content table
when encountering a node
    if (his social hierarchy is higher than me) then
        send my local content table including my content digest to him
    else
        do nothing
```

```
8:    end if
```

## 5.5 Content Request

When a content requester wants to request a certain content, an Interest packet which contains requester's node-id and content name will be generated and forwarded to a higher centrality node to avoid naïve flooding, because higher centrality node has more knowledge on content name and content providers. Each node can compute the centrality of the newly encountered node from local social-tie table. If we compute the centrality of each node in social-tie table and sort then in order, we will find that the interval of centrality is not even, as shown in Figure 5.1.

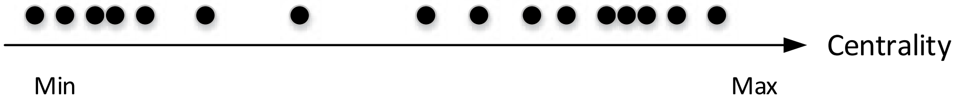


Figure 5.1 Centrality sequence

If the relay node has a similar centrality with 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 current node, to further reduce transmission cost. Inspired by clustering algorithms, periodically, we divide nodes into clusters according to their centrality distribution, and forward the Interest packet to a newly encountered node which belongs to a higher centrality cluster, as shown in Figure 5.2. The Interest packet is only forwarded from cluster A to cluster B. There is no Interest forwarding within a cluster.

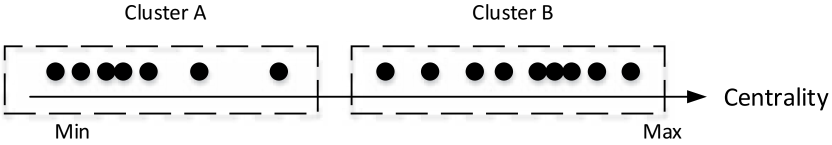


Figure 5.2 Centrality clusters

### 5.5.1 Centrality clustering

We use K-mean clustering algorithm to define clusters. K-means clustering is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. Given a set of observations ($x_1$, $x_2$, …, $x_n$), k-means clustering aims to partition the n observations into k sets (k ≤ n) **S** = {$S_1$, $S_2$, …, $S_k$} so as to minimize the within-cluster sum of squares (WCSS), as shown in (5.6), where $\mu_i$ is the mean of points in $S_i$.

$$arg\, min_s \sum_{i=1}^{k} \sum_{x_j \in S_i} \left\| x_j - u_i \right\|^2 \tag{5.6}$$

Here K (set in our experiments as 10) is a parameter that is decided by user according to the specific scenario and network scalability. The larger K value will benefit the packet delivery ratio but cause higher transmission cost. The nodes in the same cluster form a social level in centrality hierarchy. The packet will be forwarded to upper social level in our forwarding strategy.

### 5.5.2 Interest packet forwarding

As described above, we use K-mean 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 forward it to the first encountered node that has a higher social level than itself. After that, the requester keeps a copy of 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, after the last forwarding, if the node meets a better relay node which has a higher social level than last one, forwarding the

Interest packet to this new relay node will get more benefit. Following this strategy, the Interest packet is forwarded upward level by level or jumps to a higher level towards the most popular node in the centrality hierarchy, as shown in Pseudo code 5.2.

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 node’s local database (i.e., social-tie table). When the 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 routing loop due to the inconsistent centrality sequence.

Pseudo code 5.2: Forwarding Strategy

```
when having an Interest packet
check my local content table
if there is a match, then
    turn into social-tie routing
else
    keep this Interest packet
    loop: when encountering a node
    if (his social level is higher than me) then
```

```
      if (last_relay_node == null) then
          forward the Interest packet to him
          update last_relay_node
      else
          if (his social level is higher than last_relay_node) then
              forward the Interest packet to him
              update last_relay_node
          else
              do nothing
          end if
      end if
    else
        do nothing
    loop back
end if
```

A Time-To-Live (TTL) setting is configured in Interest packet and counts down during content query phase. And a waiting timer is setup by content requester after sending out the Interest packet. Routing loop will cause the TTL gets zero and 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, as shown in Pseudo code 5.3.

Pseudo code 5.3: Fallback Forwarding Strategy

```
when having an Interest packet
check my local content table
if there is a match, then
    turn into social-tie routing
else
    keep this Interest packet
    loop: when encountering a node
    if (his social level is higher than me) then
        forward the Interest packet to him
    end if
    loop back
end if
```

If the fallback forwarding strategy is adopted and the waiting timer runs out again, the content requester will start the epidemic routing [40] to flood the Interest packet throughout the network. The header of Interest packet has a forwarding flag to indicate one of three forwarding strategies set by content requester, and each relay node follows the specific forwarding strategy during the forwarding process.

## 5.6 Social-Tie Routing

In this step, the content provider's node-id has been disclosed and attached in the Interest packet. So the task is to forward the Interest packet to the destination. Similar with the centrality sequence, using the local social-tie table and K-mean 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 the 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 described in section E, an Interest packet is forwarded toward higher centrality level since a higher centrality node has more knowledge of content name digest in network. After Interest matches a content name digest, we turn to the Interest packet delivery phase which forwarding Interest packet to the content provider. In this phase, Interest packet is forwarded toward higher social-tie level of content provider since a higher level node is closer to the destination.

After the Interest packet eventually 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 responses 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 by the same forwarding strategy as the Interest packet has.

## 5.7 Caching Scheme

In this section, we first describe three prominent issues for any caching system: which data to cache, where to cache, and how to manage the cache (cache replacement policy). The ultimate goal is to maximize the cache hit rate. Subsequently, we present the caching protocol in detail.

### 5.7.1 Cached Data Selection

When the cache buffer space is free, a natural choice is to cache any data. When the cache space is full, it is more selective toward which data to cache. Intuitively, popular data is a good candidate for caching. We compute the content popularity (relative to the current node) by considering both the frequency and freshness of content requests arriving at a node over a history of request arrivals. Equation (5.7) defines the popularity of content i based on the past n requests to this content.

$$P_i = \sum_{k=1}^{n} F(t_{base} - t_k) \tag{5.7}$$

In this equation, $t_{base}$ is the current time, and $t_k$ is the past arrival time of the request for content i. We assume that the system time is represented by an integer and that t1 < t2 < ... < tn ≤ tbase. We use a weighing function $F(x) = (\frac{1}{2})^{\lambda x}$, where λ is a control parameter and 0 ≤ λ ≤ 1. The control parameter λ allows a trade-off between recency and frequency. As λ approaches 0, frequency contributes to the content popularity more than recency. On the other hand, when λ approaches 1, recency has a greater influence on the content popularity than frequency. Following [38], to achieve a good rade-off between recency and frequency, we set λ = $e^{-4}$.

### 5.7.2 Caching Location

If each node has unlimited cache space, then it is trivial to identify suitable caching locations, as data can be cached everywhere. Given that each node has limited space for caching, we follow a conservative approach and only cache data at nodes satisfying the following conditions:

1) Selected nodes are on the query forwarding paths.

2) They are traversed through by many common requests.

In social-based forwarding, requests are forwarded upward, level by level, toward socially active nodes that have high centrality value, and thus have broad knowledge of content ownership in the network. Hence, the two conditions can be easily satisfied by cluster head nodes, which have the highest social level in the network. Furthermore, to ensure that cluster head nodes are not overloaded by too many requests and cached data, we further replicate and cache popular data at downstream nodes along the request forwarding paths. This will benefit requester nodes that are in close proximity to each other as the second will get the data requested by the first. Note that each node maintains its own local view of which data is popular based on how frequent and recent requests arrive at the node. Once the node determines that a certain data is popular, it will actively request the data for caching from the content provider (if it is a cluster head node) or from an encounter node (if an encounter node carries that data).

Caching in neighbors of central nodes, whose caches are heavily utilized, is another optimization implemented in this scheme. When a central node (typically, a cluster head node or any node along the popular request forwarding paths) cannot cache new data due to limited space, it will move some of its existing cached data to neighboring nodes. Within the list of data that can be moved, the central node first moves more popular data to nodes with the strongest ties to the central node. We avoid moving data to nodes on the same forwarding paths, as the cache buffers of these nodes tend to be already heavily utilized. Query processing (i.e. cache lookup) is handled in the same order. That is, we first lookup the current node's cache. If the data is not found in the cache, the current node propagates the query to higher social-level nodes and to nearby nodes to which it has the strongest ties.

### 5.7.3 Cache Replacement

When the cache buffer is full, existing data must be evicted from the cache, to accommodate new data. There are two related issues:

1) Determining the amount of data to evict.

2) Identifying particular data to evict.

For the first issue, we need to evict as much data as the size of the new data. Regarding the second issue, we propose to remove data from the cache that is identified as least popular. That is, we consider both the frequency and freshness of data access. This replacement policy is superior to traditional cache replacement strategies such as Least Recently Used (LRU) or Least Frequently Used (LFU). In LRU-based caching, contents that were popular (but not often requested in recent times) tend to get evicted from the cache. This can lead to the eviction of popular contents when the temporal distribution of the requests to a content is not uniform. Similarly, LFU-based caching schemes do not perform well when the content pool is dynamic and the popularity of the contents in a cache decreases with time. By considering both the frequency and recency of accesses, we account for the temporal changes in content popularity.

### 5.7.4 Caching Protocol

Pseudocode 5.4 and 5.5 outline our caching protocol. In pseudocode 5.4, we assume that the Interest packet arrives at a node that does not already have the data (either cached data or owned data) that matches the request. In lines 15-16, the node actively asks the content provider for a data copy for caching at the current node only when the content popularity exceeds a threshold δ. The threshold is used to avoid frequent data replication and forwarding overhead from the content provider to the current node.

In addition, to enable cooperative sharing of popular data, nodes exchange a list of cached and owned data in the form of content name digest upon encountering each other. If the cache candidates belong to the list, nodes request them from the corresponding encounter nodes. Nodes also periodically advertise their spare cache capacity to each other. This allows central nodes to opportunistically make decisions regarding which cached data to move to neighboring nodes so that central nodes have more space to cache new popular data.

Figure 5.3 illustrates all the steps from when nodes request a content until the content is delivered and cached at intermediate nodes. We assume nodes request the same content.

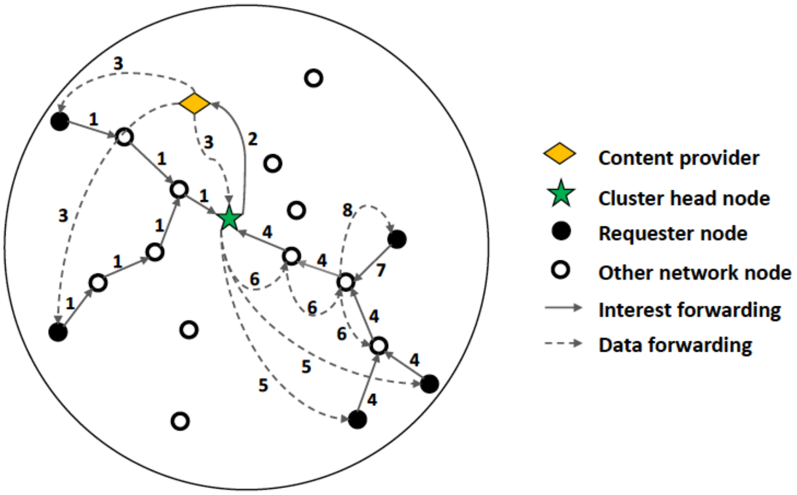


Figure 5.3 An example of caching a popular content

Pseudocode 5.4 Handle Interest Packet Arrival

```
when an Interest packet is received
if there is enough free space then
    mark the content as a cache candidate
else
    re-evaluate the popularity of requested content & cached data
    find cached data that are less popular than requested content
    if evicting them creates enough space for the content then
        mark the content as a cache candidate
    end if
end if
check my local content table for the content provider ID
if there is a match then
    social-tie route Interest packet to content provider
    if the requested content is a cache candidate then
        if popularity of requested content is higher than $\delta$ then
            request content provider to replicate data to this node
        end if
    end if
end if
```

Pseudocode 5.5 Handle Data Packet Arrival

```
when a Data packet is received
if there is a cache candidate matching the data then
    if there is not enough space in the cache buffer then
        evict data that are less popular than cache candidate
    end if
    cache the received data
end if
```

1) Interest packets are generated and routed to the cluster head on two different paths using social-level routing.

2) Cluster head social-tie routes the Interest packets to the content provider. Assume the popularity of the requested content exceeds threshold δ. Cluster head also requests the content provider to send it a copy of the content.

3) The content provider social-tie routes the Data packets to the two requesters and the cluster head. The content is then cached at the cluster head.

4) Additional Interest packets are generated and routed toward the cluster head using social-level routing.

5) Since the cluster head has a cache copy of the content, the cluster head social-tie routes the content to the requesters.

6) Nodes along the common request forwarding paths re- quest a cache copy of the content from upstream nodes, and cache it locally.

7) An Interest packet is generated and routed toward higher social-level node.

8) Since the node has the cache copy, it can serve the request without propagating the request upward to the cluster head.

# 5.8 Evaluation

In this section, we evaluate the performance of the proposed STCR scheme in a packet-level simulation with the real mobility trace dataset.

## 5.8.1 Simulation Setup

We implemented the proposed STCR scheme in NS-3.17 network simulator. DTN nodes advertise their *Hello* message to each other every 100 ms. In order to test the bottom line of the performance, we assume that each node has a unique content which is different with all other nodes. Periodically (every 30 seconds), a random node is selected to generate an Interest packet for any content. We also assume that the content data can be retrieved as 1MB size so that the measurement will not be affected by the content size variance. We fix the content popularity threshold $\delta$ to 2.5. That is, we consider content as popular when it arrives at a node 3 times within 300ms interval. Finally, we assume that the caching buffer of nodes is uniformly distributed in range [10MB, 30MB].

We use the IEEE 802.11g wireless channel model and the PHY/MAC parameters as listed in Table 5.2. To gain meaningful results, we use INFOCOM’06 contact traces taken from 98 nodes during the INFOCOM’06 conference in 95 hours [37].

## 5.8.2 Evaluation Metrics

To abstract away from any particular routing algorithm, we use the Epidemic routing [40] in an attempt to understand the upper bounds of connectivity (epidemic has the highest delivery probability). In Epidemic, when two nodes meet each other, they exchange messages that they haven’t seen. This scheme makes the Epidemic routing creates unlimited number of messages by copying the messages to all nodes that do not yet have to copy. We also compare our design to PeopleRank and BubbleRap which is advanced to SimBetTS as claimed in [29]. We use the following metrics in the experiments.

1) Hit rate: the percentage of Interests that are successfully delivered to the content providers and the content data are successfully delivered to the requesters. This metric reflects the capability of a method to discover the requested content.

2) Average delay: the average delay time of the successfully delivered content from the Interest has been sent out. This metric reflects the efficiency of a method to discover the requested content and retrieve it back.

3) Total cost: the total number of messages replicas in the network. To normalize this, we divide it by the total number of unique messages created.

Table 5.2 Simulation parameters

| Parameter | Value |
|---|---|
| RxNoiseFigure | 7 |
| TxPowerLevels | 1 |
| TxPowerStart/TxPowerEnd | 12.5 dBm |
| m_channelStartingFrequency | 2407 MHz |
| TxGain/RxGain | 1.0 |

| | |
|---|---|
| EnergyDetectionThreshold | -74.5 dBm |
| CcaMode1Threshold | -77.5 dBm |
| RTSThreshold | 0 B |
| CWMin | 15 |
| CWMax | 1023 |
| ShortEntryLimit | 7 |
| LongEntryLimit | 7 |
| SlotTime | 20 µs |
| SIFS | 10 µs |

### 5.8.3 Experiment Results

1) Hit rate

Figure 5.4 shows the hit rates over time in different methods. The parameter *K* is set to 10. We find that Epidemic can retrieve most requested contents while our proposed STCR scheme finally reaches about 68% which is better than BubbleRap and PeopleRank. Epidemic has the highest hit rate because a node copies its Interest to all other nodes, which will eventually hit the content provider in highest probability. The performance of STCR is worse than that of Epidemic method, but still better than that of BubbleRap and PeopleRank, which shows the social-tie routing scheme is more efficient than BubbleRap's centrality + community scheme and even better than PeopleRank's centrality-only scheme.

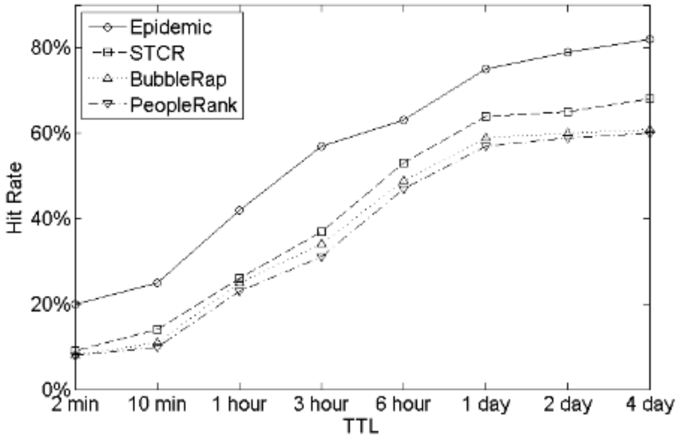


Figure 5.4 Hit rate

2) Average delay

Figure 5.5 illustrates the average delay over time of the three methods. From the figure we can see that the delay of Epidemic is much lower than other three methods. We only measure the delay of successful content retrieval. In Epidemic, Interest is rapidly flooded in network. As a result, the Interest can reach its destination after a short delay. Both BubbleRap and PeopleRank show larger delays than STCR because both Interest and content packets are only forwarded to the higher social centrality nodes which causes a longer waiting time both in content query phase and content retrieval phase. And PeopleRank gets the highest delay since it only uses centrality without any community information as BubbleRap.

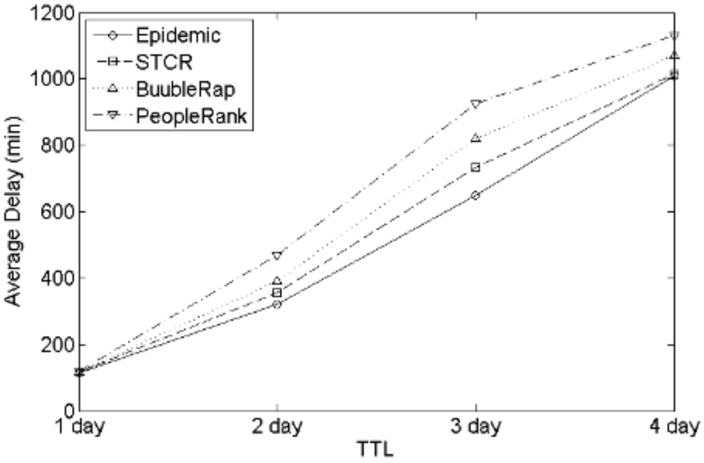


Figure 5.5 Average delay

3) Total cost

Figure 5.6 shows the total cost of different methods. The cost of Epidemic exceeds all other protocols. In Epidemic, each node delivers its Interest or content packet with every encountered node, which contributes to the highest cost. In STCR, each node exchanges its social-tie table with other newly encountered nodes, but only deliver the Interest and content packet to selected relay nodes using the clustering method. So, in terms of transmission cost, STCR is better than BubbleRap and PeopleRank.

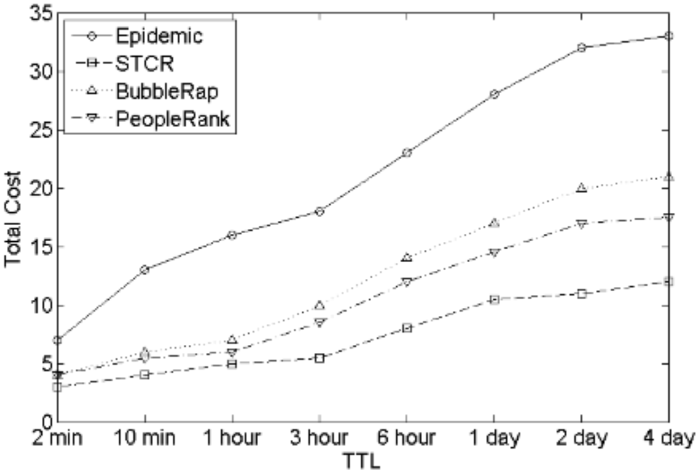


Figure 5.6 Total cost

4) Impact of K-mean

The parameter K is set to 10 in above experiments. A higher K will induce more clusters in network, thus generate more packet forwarding which benefits Hit Rate and Average Delay but hurts Total Cost. So the value of K should be selected carefully according to the specific scenario. We exam K with different value to show its impact to these evaluation metrics, as shown in Figure 5.7-5.9.

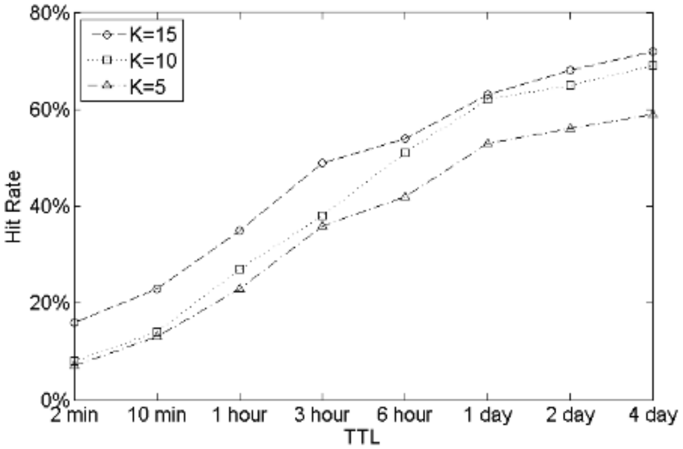


Figure 5.7 Hit rate with different K

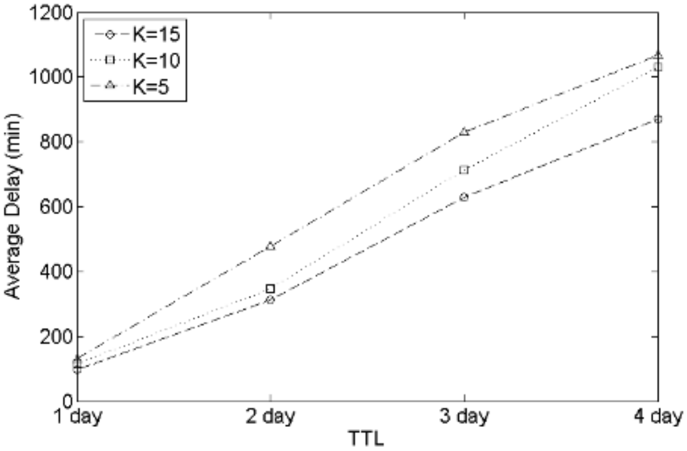


Figure 5.8 Average delay with different K

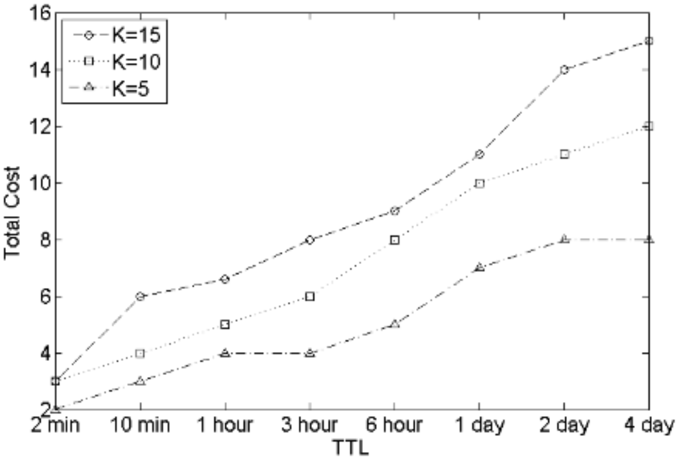
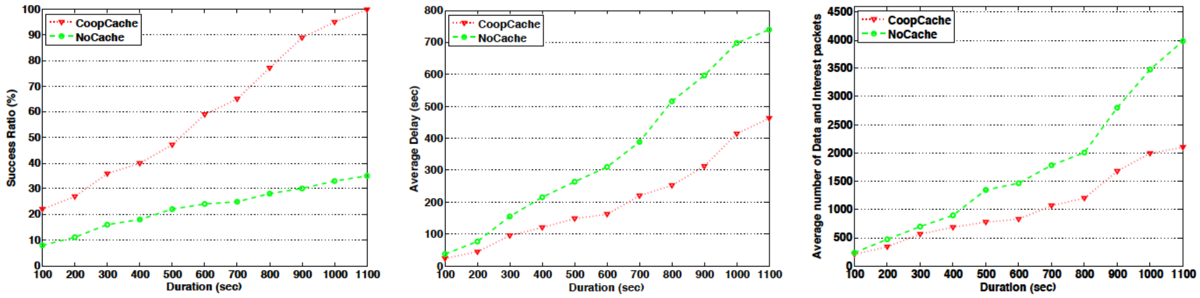


Figure 5.9 Total cost with different K

5) Caching Performance

Figure 5.10 shows the performance of NoCache scheme and our proposed CoopCache scheme. As we increase the simulation time from 100s to 1,100s, the success ratio of both schemes is improved, because data has more time to be discovered and delivered to requesters. However, the improved ratio is increased at a significantly faster rate in CoopCache scheme than in NoCache scheme. This is because CoopCache replicates and caches popular data at nodes close to the requesters, resulting in a higher hit rate and lower latency. For the same reason, CoopCache has much lower average delay than NoCache. In NoCache, the majority of content queries need to be propagated to the cluster head node, and then social-tie routed to the content providers. The Interest forwarding step alone adds a significant delay to the overall content query and delivery delay in NoCache. CoopCache, by leveraging intermediate caching nodes along the common forwarding paths, can eliminate many of the delays from the NoCache scheme. Finally, in Figure 5.10(c), NoCache suffers a very high cost. Content query and delivery in NoCache often traverses many hops, thus resulting in a large amount of Interest and Data packet replication. CoopCache, on the other hand, uses intermediate nodes for caching, thus shortening the Interest forwarding paths and lowering the overall cost of the system.



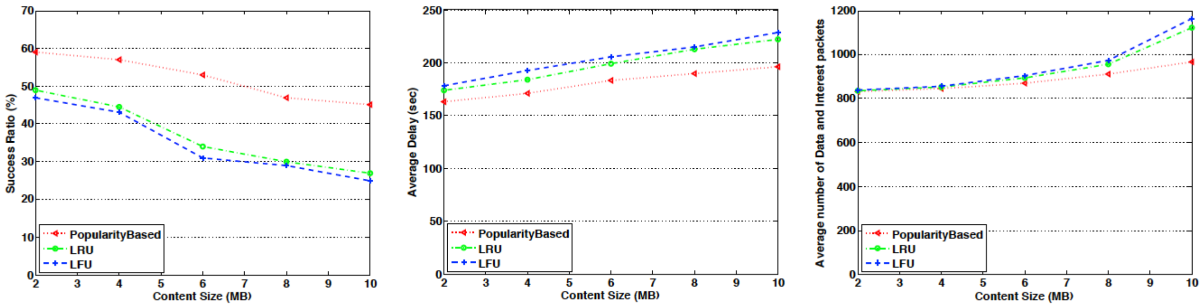
**(a)** Success ratio  **(b)** Average delay  **(c)** Total cost

Figure 5.10 Performance of content retrieval with different simulation duration

6) Performance of Cache Replacement

In this subsection, we evaluate the performance of our proposed cache replacement policy based on content popularity. We compare our policy against the traditional replacement policies including LFU and LRU. We fix the simulation duration to 600s. We vary the content size from 1 to 10MB and still assume that all contents are of the same size. This enables us to increasingly put more pressure on cache replacement to observe the effectiveness of different schemes.

The simulation results are shown in Figure 5.11. The popularity-based replacement scheme outperforms LFU and LRU policy on all three metrics. The performance gap grows bigger as we increase the content size. This is because when the content size is small, the cache buffer constraint is not tight, and therefore cache replacement is not frequently con- ducted. Subsequently, the performance difference is not too significant. However, when the content size becomes larger, cache replacement is conducted more frequently, and LFU and LRU do not always select the most appropriate data to cache, due to improper consideration of content popularity. Thus, the advantage of our popularity-based scheme rises significantly when the content size is set to 10MB.



**(a)** Success ratio

**(b)** Average delay

**(c)** Total cost

Figure 5.11 Performance of content retrieval with different cache replacement policies

# Chapter 6. Conclusion

Content retrieval in mobile ad hoc networks has been an active research area. However, the scalability of the content retrieval in MANET has not been assessed well. Current solutions spread the Interest packet to the entire network which will cause a large traffic overhead and will suffer from the scalability problem in the large scale MANETs. In this dissertation, we proposed an energy-efficient content retrieval scheme for mobile ad hoc network to decrease the traffic overhead of the Interest dissemination and save the transmission energy. The selected relay node forwards the Interest packet along the specific direction under the control of last hop. It will significantly decrease the traffic overhead of the duplicate copies of Interest, and will terminate the further relay in a way once the content data has been found. We advocate the parallel search method, in which content requester selects several agents by random-walk to conduct the proposed direction-selective Interest dissemination scheme parallel. It decreases the impact of the missing gap from the direction-selective dissemination and increases the hit rate for the content retrieval. In addition to the extensive study in MANET scenatios, we further explore efficient content retrival scheme for disruption-tolerant networks. We proposed STCR, a social-tie based content retrieval scheme that is highly scalable in disruption-tolerant mobile information-centric networks. The STCR generates the social-tie based routing structure in order to support an efficient Interest and content forwarding. We proposed some novel methods to compute social metrics considering both the frequency and freshness of encounters, and balanced connectivity with all other nodes to improve the delivery rate. We proposed a new cooperative caching scheme based on the social relationship among nodes in DTNs. In this scheme, data is dynamically cached at selective locations in the network such as cluster head nodes, which have the highest social levels, and nodes along the common request forwarding paths. We described a new cache replacement policy based on the content popularity, which is a function of both the frequency and recency of data access.

The experiment results show that the proposed content retrieval schemes reduces the total cost of the traffic overhead compared to the current ICN forwarding scheme, and prunes a number of duplicate copies of the Interest with a high hit rate to retrieve the content successfully.

# Reference

[1] D. E. Comer, *Computer Networks and Internets*, 5th ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2008.

[2] T. Clausen and P. Jacquet, "RFC 3626 - Optimized Link State Routing Protocol (OLSR)," *IETF RFC3626*. p. 75, 2003.

[3] C. Perkins, E. Belding-Royer, and S. Das, "RFC 3561 - Ad hoc On-Demand Distance Vector (AODV) Routing," *Internet RFCs*. Mobile Ad Hoc Networking Working Group of the Internet Engineering Task Force, 2003.

[4] J. Crowcroft, S. Hand, R. Mortier, T. Roscoe, and A. Warfield, "Plutarch: An Argument for Network Pluralism," in *Proceedings of the ACM SIGCOMM Workshop on Future Directions in Network Architecture*, 2003, pp. 258–266.

[5] S. Schmid, L. Eggert, M. Brunner, and J. Quittek, "TurfNet: An Architecture for Dynamically Composable Networks," in *Autonomic Communication SE - 8*, vol. 3457, M. Smirnov, Ed. Springer Berlin Heidelberg, 2005, pp. 94–114.

[6] S. Hares and R. White, "BGP Dynamic AS reconfiguration," in *Proceedings - IEEE Military Communications Conference MILCOM*, 2007.

[7] C.-K. Chau, J. Crowcroft, K.-W. Lee, and S. H. Y. Wong, "Inter-domain Routing over Mobile Ad-hoc Networks," in *Hotmobile*, 2007.

[8] C.-K. Chau, J. Crowcroft, K.-W. Lee, and S. H. Y. Wong, "IDRM: Inter-Domain Routing Protocol for Mobile Ad Hoc Networks," *Univ. Cambridge Tech. Rep. UCAM-CL-TR-708*, 2008.

[9] V. Ramasubramanian, Z. J. Haas, and E. G. Sirer, "SHARP: A Hybrid Adaptive Routing Protocol for Mobile Ad Hoc Networks," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing*, 2003, pp. 303–314.

[10] G. Pei, M. Gerla, and X. Hong, "LANMAR: Landmark Routing for Large Scale Wireless Ad Hoc Networks with Group Mobility," in *Proceedings of the 1st ACM International Symposium on Mobile Ad Hoc Networking & Computing*, 2000, pp. 11–18.

[11] Z. J. Haas, M. R. Pearlman, and P. Samar, “The zone routing protocol (ZRP) for ad hoc networks,” *Draft. txt*, 2002.

[12] V. Jacobson, M. Mosko, D. Smetters, and J. Garcia-Luna-Aceves, “Content-centric networking,” *Whitepaper, Palo Alto Res. Cent.*, pp. 2–4, 2007.

[13] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos, and others, “Named data networking (ndn) project,” *Relat. Tec. NDN-0001, Xerox Palo Alto Res. Center-PARC*, 2010.

[14] L. Wang, A. Afanasyev, R. Kuntz, R. Vuyyuru, R. Wakikawa, and L. Zhang, “Rapid Traffic Information Dissemination Using Named Data,” in *Proceedings of the 1st ACM Workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications*, 2012, pp. 7–12.

[15] S. Y. Oh, D. Lau, and M. Gerla, “Content Centric Networking in tactical and emergency MANETs,” in *Wireless Days (WD), 2010 IFIP*, 2010, pp. 1–5.

[16] M. N. Mariyasagayam, T. Osafune, and M. Lenardi, “Enhanced Multi-Hop Vehicular Broadcast (MHVB) for Active Safety Applications,” in *Telecommunications, 2007. ITST ’07. 7th International Conference on ITS*, 2007, pp. 1–6.

[17] L.-C. Tung and M. Gerla, “An efficient road-based directional broadcast protocol for urban VANETs,” in *Vehicular Networking Conference (VNC), 2010 IEEE*, 2010, pp. 9–16.

[18] E. Fasolo, A. Zanella, and M. Zorzi, “An Effective Broadcast Scheme for Alert Message Propagation in Vehicular Ad hoc Networks,” in *Communications, 2006. ICC ’06. IEEE International Conference on*, 2006, vol. 9, pp. 3960–3965.

[19] M. Nekovee, “Epidemic algorithms for reliable and efficient information dissemination in vehicular,” *Intell. Transp. Syst. IET*, vol. 3, no. 2, pp. 104–110, Jun. 2009.

[20] K. Fall, “A Delay-tolerant Network Architecture for Challenged Internets,” in *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 2003, pp. 27–34.

[21] K. L. Scott and S. Burleigh, “Bundle protocol specification,” in *RFC 5050*, 2007.

[22] N. Sastry, D. Manjunath, K. Sollins, and J. Crowcroft, “Data Delivery Properties of Human Contact Networks,” *Mob. Comput. IEEE Trans.*, vol. 10, no. 6, pp. 868–880, Jun. 2011.

[23] J. Scott, J. Crowcroft, P. Hui, and C. Diot, “Haggle: A networking architecture designed around mobile users,” in *WONS 2006: Third Annual Conference on Wireless On-demand Network Systems and Services*, 2006, pp. 78–86.

[24] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, “Pocket switched networks: Real-world mobility and its consequences for opportunistic forwarding,” 2005.

[25] W. Hsu and A. Helmy, “Impact: Investigation of mobile-user patterns across university campuses using wlan trace analysis,” *arXiv Prepr. cs/0508009*, 2005.

[26] E. M. Daly and M. Haahr, “Social Network Analysis for Information Flow in Disconnected Delay-Tolerant MANETs,” *Mob. Comput. IEEE Trans.*, vol. 8, no. 5, pp. 606–621, May 2009.

[27] A. Mtibaa, M. May, C. Diot, and M. Ammar, “PeopleRank: Social Opportunistic Forwarding,” in *INFOCOM, 2010 Proceedings IEEE*, 2010, pp. 1–5.

[28] S. Brin and L. Page, “The anatomy of a large-scale hypertextual web search engine, 1998,” in *Proceedings of the Seventh World Wide Web Conference*, 2007.

[29] P. Hui, J. Crowcroft, and E. Yoneki, “BUBBLE Rap: Social-Based Forwarding in Delay-Tolerant Networks,” *Mob. Comput. IEEE Trans.*, vol. 10, no. 11, pp. 1576–1589, Nov. 2011.

[30] R. Jain, D.-M. Chiu, and W. Hawe, “A quantitative measure of fairness and discrimination for resource allocation in shared computer systems,” 1998.

[31] B. Zhou, Y.-Z. Lee, and M. Gerla, “‘Direction’ assisted Geographic Routing for mobile ad hoc networks,” in *Military Communications Conference, 2008. MILCOM 2008. IEEE*, 2008, pp. 1–7.

[32] X. Hong, K. Xu, and M. Gerla, “Scalable routing protocols for mobile ad hoc networks,” *Network, IEEE*, vol. 16, no. 4, pp. 11–21, Jul. 2002.

[33] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla, “Glomosim: A scalable

network simulation environment," *UCLA Comput. Sci. Dep. Tech. Rep.*, vol. 990027, p. 213, 1999.

[34] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, "A Group Mobility Model for Ad Hoc Wireless Networks," in *Proceedings of the 2Nd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 1999, pp. 53–60.

[35] N. Sarshar, P. O. Boykin, and V. P. Roychowdhury, "Percolation search in power law networks: making unstructured peer-to-peer networks scalable," in *Peer-to-Peer Computing, 2004. Proceedings. Proceedings. Fourth International Conference on*, 2004, pp. 2–9.

[36] N. Sarshar, P. O. Boykin, and V. P. Roychowdhury, "Finite Percolation at a Multiple of the Threshold," *arXiv Prepr. cond-mat/0601211*, 2006.

[37] J. Yeo, D. Kotz, and T. Henderson, "CRAWDAD: A Community Resource for Archiving Wireless Data at Dartmouth," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 2, pp. 21–22, Apr. 2006.

[38] D. Lee, J. Choi, J. H. Kim, S. H. Noh, S. L. Min, Y. Cho, and C. S. Kim, "LRFU: A Spectrum of Policies that Subsumes the Least Recently Used and Least Frequently Used Policies," *IEEE Trans. Comput.*, vol. 50, no. 12, pp. 1352–1361, 2001.

[39] M. V Ramakrishna, "Practical Performance of Bloom Filters and Parallel Free-text Searching," *Commun. ACM*, vol. 32, no. 10, pp. 1237–1239, Oct. 1989.

[40] A. Vahdat, D. Becker, and others, "Epidemic routing for partially connected ad hoc networks," 2000.