12-15-2016

Data Dissemination And Information Diffusion In Social Networks

Guoliang Liu

Recommended Citation
Liu, Guoliang, "Data Dissemination And Information Diffusion In Social Networks." Dissertation, Georgia State University, 2016.
doi: https://doi.org/10.57709/9430394

DATA DISSEMINATION AND INFORMATION DIFFUSION IN SOCIAL NETWORKS

by

GUOLIANG LIU

Under the Direction of Yingshu Li, PhD

ABSTRACT

Data dissemination problem is a challenging issue in social networks, especially in mobile social networks, which grows rapidly in recent years worldwide with a significant increasing number of hand-on mobile devices such as smart phones and pads. Short-range radio communications equipped in mobile devices enable mobile users to access their interested contents not only from access points of Internet but also from other mobile users. Through proper data dissemination among mobile users, the bandwidth of the short-range communications can be better utilized and alleviate the stress on the bandwidth of the cellular networks. In this dissertation proposal, data dissemination problem in mobile social networks is studied. Before data dissemination emerges in the research of mobile social networks, routing protocol of finding efficient routing path in mobile social networks was the focus, which later became the pavement for the study of the efficient data dissemination. Data dissemination priorities on packet dissemination from multiple sources to multiple destinations while routing protocol simply focus on finding routing path between two ends in the networks. The first works in the literature of data dissemination problem were based on the modification and improvement of routing protocols in mobile social networks. Therefore, we first studied

and proposed a prediction-based routing protocol in delay tolerant networks. Delay tolerant network appears earlier than mobile social networks. With respect to delay tolerant networks, mobile social networks also consider social patterns as well as mobility patterns. In our work, we simply come up with the prediction-based routing protocol through analysis of user mobility patterns. We can also apply our proposed protocol in mobile social networks. Secondly, in literature, efficient data dissemination schemes are proposed to improve the data dissemination ratio and with reasonable overhead in the networks. However, the overhead may be not well controlled in the existing works. A social-aware data dissemination scheme is proposed in this dissertation proposal to study efficient data dissemination problem with controlled overhead in mobile social networks. The data dissemination scheme is based on the study on both mobility patterns and social patterns of mobile social networks. Thirdly, in real world cases, an efficient data dissemination in mobile social networks can never be realized if mobile users are selfish, which is true unfortunately in fact. Therefore, how to strengthen nodal cooperation for data dissemination is studied and a credit-based incentive data dissemination protocol is also proposed in this dissertation. Data dissemination problem was primarily researched on mobile social networks. When consider large social networks like online social networks, another similar problem was researched, namely, information diffusion problem. One specific problem is influence maximization problem in online social networks, which maximize the result of information diffusion process. In this dissertation proposal, we proposed a new information diffusion model, namely, sustaining cascading (SC) model to study the influence maximization problem and based on the SC model, we further plan our research work on the information diffusion problem aiming at minimizing the influence diffusion time with subject to an estimated influence coverage.

INDEX WORDS: Data Dissemination, Mobile Social Networks, Delay Tolerant Networks, Online Social Networks, Social Networks, Influence Maximization, Information Diffusion

DATA DISSEMINATION AND INFORMATION DIFFUSION IN SOCIAL NETWORKS

by

Guoliang Liu

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2016

DATA DISSEMINATION AND INFORMATION DIFFUSION IN SOCIAL NETWORKS

by

GUOLIANG LIU

Committee Chair: Yingshu Li

Committee: Raj Sunderraman

Xiaojun Cao

Hendricus Van der Holst

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

November 2016

DEDICATION

This dissertation is dedicated to my parents, my sisters and my lovely wife.

ACKNOWLEDGEMENTS

I would like to show my great gratitude to all of those who helped me complete dissertation and my study at Georgia State University.

I would like to express my deepest gratitude to my advisor, Dr. Yingshu Li for her excellent inspiration, direction, patience, and providing me with the great research environment.

I would like to give special thanks to Dr. Zhipeng Cai, who supported me in my first two years of study and research and kept giving me great suggestions both in research and life.

I would like to thank my committee members, Dr. Raj Sunderraman, Dr. Xiaojun Cao and Dr. Hendricus Van der Holst. Dr. Sunderraman and Dr. Cao guided me through my study and research. They always encourage us to be more creative in and out of classrooms. Besides, I feel it is an honor to have Dr. Hendricus Van der Holst as my committee after taking his mathematic class.

I also would like to thank the professors and staffs at our department, especially Ms. Tammie Dudley, who is the most professional and nicest secretary I have ever met.

I would like to thank my group members and fellow students. Special thanks go to Dr. Jing He, Dr. Shouling Ji, Dr. Mingyuan Yan and Meng Han, who provided me help and caring besides on research.

Last but not least, I would like to thank my family and my friends for their support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Background

Along with the development of Internet, cellular networks and the popularization of mobile devices, Social networks and its applications has gained more attention from all fields, especially computer science and engineering. Social networks first emerged from social psychology and it studied the relationship between entities. At first, entities are understood as people the social networks study the relationships of individual of group of people. Now entities could also be considered as groups, organizations, devices or even systems. A social network is a social structure of entities that connects to each other from some relationship. The very first study on social network structure focused on finding triads of social groups or affiliations. With the development of online social applications and mobile social applications, the concept of social networks has been used with the combination of information exchange and communication technologies to provide message delivery, data sharing and information spread services. In this proposal, we study information spread in social networks. Specifically, our study is based on two kinds of social networks. One is mobile social network and the other kind is online social network. The problem of how to spread information on these two kinds of social networks usually is referred as different names. They are data dissemination problem in mobile social networks and information diffusion problem in online social networks. Since these two kinds of mobile social networks have different characteristics, the information spread problem also have different design and applications. We introduce two kinds of networks separately, followed by the specific information spread problem in the networks.

1.2 Characteristics of Mobile Social Networks and Data Dissemination Problem

Mobile Social Networks (MSNs) have become an emerging wireless communication techniques with the explosive increment of smart devices like smart phones and pads in people's daily life. In order to alleviate the daily growing needs of bandwidth of cellular networks, information sharing among mobile users through short-range radios communications like Bluetooth and Wi-Fi is highly encouraged. Through single or multiple short-range communications among the mobile users in MSN, users' interested data can spread over MSNs in a delay tolerant way. However, for a given message with some interest, it is not easy to disseminate it to its targeted users (who are interested in) directly or indirectly in an efficient way. The reason is that data dissemination in MSNs suffers from the intermittent connectivity and unpredictable node mobility among mobile users, which makes the user contact opportunities very cherishable. Also, a successfully delivered message may possibly require multiple relays from the users that are not interested in it and the challenges is whether the intermediate users are willingly to carry such messages or not. Even if they do, what messages should they store, carry and forward is the problem since it is not realistic to store and carry all messages due to storage capacity and bandwidth of the mobile devices. In this work, we try to maximize the delivery ratio by optimizing the message forwarding through learning the mobility pattern of mobile users and interest distribution with a new defined overhead that we could control. Along with over a decade of research on MSNs, different architectures have been developed and considered in existing works. Here we introduce the main architectures to model MSNs. In summary, MSNs have three kinds of architectures: centralized, distributed and hybrid architectures [42]. In the centralized architecture, mobile nodes are all connected with Access Points (APs) and communicate with each other in a client-server manner. In the distributed architecture, there is no AP and mobile nodes only communicate with each other using short-range ratio. There is no message disseminator and messages are generated by all mobile nodes. The hybrid architecture is a mixture of the previous two architectures. In the hybrid architecture, most messages are disseminated from APs and only a certain number of mobile nodes have uncertain access to APs and

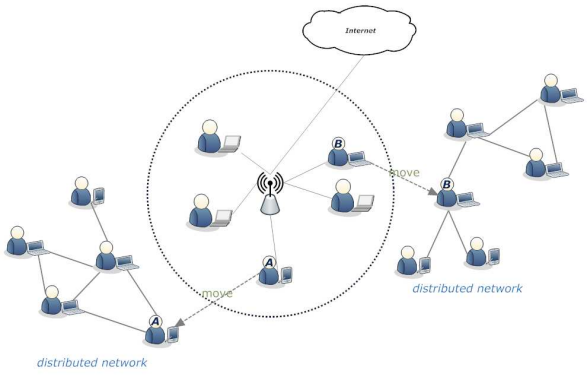


Figure 1.1. An MSN with the hybrid architecture.

these mobile nodes further disseminate messages to the rest of the nodes in a network using short-range radios. The hybrid architecture is the most realistic and commonly studied one in research and we also use this architecture to study mobile social networks in this proposal. In the hybrid architecture, two mobile users forward messages to each other only when there is a direct contact between them, *i.e.,* two mobile nodes need to be in each other's short radio range to carry out a message exchange. A message is disseminated in a delay-tolerant manner and has high risks of delivery failure. A message may be dropped before it finally reaches its prospective receivers. Fig.4.1 shows an instance of data dissemination in hybrid architecture, where node A and node B have access to APs and when they move out of the access range of the APs, they further disseminate messages to other nodes in a distributed manner.

Data dissemination problem in mobile social networks sometimes is also referred as content distribution problem. The problem is challenging in mobile social networks because of the intermediate connectivity, limited resources and social diversity. The data dissemination problem is to disseminate messages in different kinds of interests to users with corresponding interests while incurring minimum overhead. The underlying principal is to find the most appropriate forwarding node to carry, store and relay the messages. The overhead usually means the times of message forwarding by uninterested relay nodes who are willingly or stimulated to help others achieve their wanted messages. To design a good data dissemination scheme or protocol, usually social patterns and mobility patterns are studied to help

decide what messages should be forwarded in a contact and predict contacts in near future in order to achieve a better data dissemination ratio with less overhead. Social patterns can be learned through users' connectivity, interest, contact duration, closeness (social tie) and so on. The social patterns can influence users's interest from time to time. Also, social patterns may also influence their willingness to help forward their uninterested messages. Social patterns consider users' social profile the information may not be disclosed for research in the real world. Compared with social patterns, mobility patterns mainly involve people's (devices') encounter and r-encounter patterns. Through analysis of people's mobility patterns and using some prediction model, for each message, which contacts in future may be optimum can be predicted and therefore achieving a better dissemination ratio. In this proposal, we mainly study users' mobility patterns but also consider users' interest distribution's influence.

Most existing works through studying mobility patterns and social patterns which aim at improving dissemination ratio and reducing delay in MSNs assume that all the nodes are completely cooperative. However, the mobile users in reality can be either cooperative or selfish. More precisely, some mobile users may be cooperative if they have extra resources such as abundant AP access time, enough bandwidth and storage buffers. In the meanwhile, most of them are selfish naturally and resources are always limited at most time. Moreover, if resources are limited, no matter whether mobile users are cooperative or not, they also need to be smart to choose the messages considering both the prospective of the whole network performance and each individuals own benefits. Therefore, a practical incentive scheme is essential to encourage nodes to be wisely cooperative. In this proposal, we also study how to stimulate nodes to be cooperative in the data dissemination problem.

1.3 Characteristic of Online Social Networks and Information Diffusion Problem

In recent years, online social networks such as Facebook, Twitter and Instagram grow rapidly with hundreds of millions of users. Compared with mobile social networks, which

currently lack of applications in the real world, the development of these online social network applications enables the study of social networks at a large scale. Online social networks belong to complex networks and therefore it has the characteristics such as small-world effect, community structure, transitivity and power-law degree distribution. Below we list the characteristics of online social networks which needs to be considered in the information diffusion problem.

- Large Scale. The online social networks in research usually involve a part of a real social network, which may consist of hundreds of thousands to millions of nodes.
- Small-world effect. Even though the size of the online social network is very large, the diameter of the network may be shockingly small. E.g., a rumor may be able to spread over a network with millions of nodes in just several steps.
- Transitivity. It is observed that there are more triangles in online social network graph than a general random graph, i.e., if node A connects with node B and node B connects with node C, then it is highly possible that there is a link between node A and C. By studying the transitivity properties, researchers also find clusters in online social networks, which is more referred as community structure. Community structures also help people design information diffusion model to study information spread in online social networks.
- Degree follows power-law distribution. Large online social networks have very different graph structure from random graphs. One special property is the degree distribution, which follows a power-law distribution. Online social networks are scale-free networks since the degree of nodes scale with the size of the network. It is relatively easy to find some large degree nodes in the networks and which also explains to some extent why the diameter of the network is rather small. In information diffusion problem, detecting and using nodes with largest degrees is very critical.

Information diffusion in online social networks has been studied as a critical problem in all kinds of domains including computer science, mathematics, statistics, business and medi-

cal techniques. An information diffusion model can help people understand how information spread in online social networks. One particular problem based on information diffusion model is influence maximization problem. Influence maximization is to select k nodes under a particular information diffusion model to maximize the influence across social networks. When understand influence as a kind of message, influence maximization problem actually belongs to information diffusion problems. Information diffusion models decide the way how influence propagate through social network. In this proposal, we study information diffusion models and the influence maximization problem.

1.4 Organization

The rest of this dissertation proposal is organized as follows: Chapter 2 summarized the related literature. Chapter 3 first studies a prediction-based routing with packet scheduling in mobile social networks. Chapter 4 investigates a data dissemination scheme which based on a controlled limited overhead. Chapter 5 introduces our new proposed information diffusion model in online social networks. Chapter 6 investigates how to stimulate nodes to be more cooperative in data dissemination problem. In Chapter 7, based on the proposed information diffusion model in 5, we target at minimize influence diffusion time with estimated influence coverage. In Chapter 8, we conclude this dissertation.

Chapter 2

RELATED WORK

2.1 Data Dissemination Problem in MSN

Before MSNs draw people's attention, Delay Tolerant Networks (DTNs) have been studied for a decade. Compared with DTNs, MSNs are considered having more mobility and social patterns of people's activities like people's daily regular contacts and different kinds of content interest distributions. Early works on DTNs or MSNs can be classified according to the propagation methods: epidemic routing [75], multicast [76][77][34] and data dissemination [80][81][82][83] [36][87][88][35] [89][90][91][92]. Unicast and multicast protocols are extended from routing protocols and limited in specific destinations. Data dissemination in MSNs is to disseminate contents of one or multiple interest types to interested mobile nodes and meanwhile try to minimize the number of uninterested mobile nodes carrying out relay jobs. Data dissemination in MSNs can be further classified into three categories.

The first category is data dissemination through learning social and mobility patterns. The works in [80][81][82][83] all argue that user contacts in MSNs are repeatable and future contacts are predicable through learning the encounter and re-encounter pattern between users and users social properties like their interests and closeness of friendships. The work in [83] also studies the *tie strength* between two users, *i.e.*, the closeness, contact quality like contact frenquency and contact durations between two users. The works in [84][85] further study the *tie strength* to better route and disseminate messages in MSNs. Our work can be classified into this category since we also need to learn users' encounters and their interest matches in future contacts. Different from the previous works, instead of focusing on contacts between two users, we combine users' interests with user contacts to see how users' interests may be overlap with or differentiate from each other according to the contacts during a period of time. Inspired by the work in [93], we apply a time-homegeneous semi-Markov

model to help decide proper nodes to serve as servers and when the role transitions from client to server or from server to client may happen for a kind of messages.

The second category involves the utility-based optimization methods [36][87][88][35]. The work in [36] defines a utility function to minimize dissemination delay. The work in [87] employs a Markov decision process to maximize the number of users who have *fresh* messages. The work in [35] defines global and local utilities for data dissemination. It transfers the data dissemination problem to the maximum weight bipartite matching problem to find the optimal solution. In our work, we also define a utility function to decide, for each message, between a pair of nodes, which one is better to be a server to further disseminate the message.

The third category is context-aware data dissemination [89][90][91][92]. These works assume to know the situation where the node is addressed. The information of a user like user identity, locations, interests and *etc* can be achieved during data dissemination. The learning process can be used to increase the QoS of data dissemination by adopting users behaviors to different situations.

Most existing works aim to optimize the data dissemination, but without a control or a good estimation of overhead, especially when network size changes. Our work in this proposal proposes a new overhead, namely, *service overhead* to control the message load in the network effectively. We aim to maximize the data dissemination through optimizing the nodes to find the best message carrier to disseminate it. As to our best knowledge, it is the first work to define user's neighbors' interest transition functions through a semi-Markov model to enable nodes' social-awareness.

Most works which aim to optimize the data dissemination ratio assume that nodes in MSNs are willing to help each other for data dissemination. However, in the real world, users are believed to be selfish naturally. Therefore, we discuss the current incentive schemes in wireless networks and their applications in delay tolerant networks and MSNs (both kinds of networks have the characteristic of intermittent connectivity). Furthermore, we discuss the employed incentive scheme for data dissemination in MSNs.

The incentive scheme has been well studied in mobile ad hoc networks to strengthen

nodal cooperations. In general, there are three main kinds of incentive schemes in the literature: reputation, barter (or Tit-for-Tat) and credit (virtual currency).

The reputation-based schemes are widely used in mobile ad hoc networks [45][46][55][56][57][58]. In this scheme, a node's reputation increases as reward when it becomes more cooperative. Nodes with higher reputation have higher priority to transfer their own messages. Reputation-based schemes are usually employed in solving the routing problem in mobile ad hoc networks. However, reputation-based schemes suffer from the safety issues such as sybil attacks in which malicious users can collude with each other to get high reputation.

The barter-based schemes are based on the pair-wise exchange mode. Each pair of nodes treat each other equally in forwarding packets. The works in [59][60][61] are barter-based applications in delay tolerant networks. The optimization based barter scheme can formulate a problem as the classic problem in game theory and try to reach Nash Equilibria [61]. However, the barter-based scheme is not suitable for stimulating nodal cooperation for data dissemination in MSNs. The pair-wise mode focuses on the fairness between a pair of nodes, which can encourage each pair to be cooperative but also degrades the degree of cooperation in the whole network. For example, consider three nodes A, B and C. Every two of them form a pair. Node A is able to provide more services than it can get from peer B. Node B has the same conditions with peer C. Node C has the same conditions with peer A. The imbalance in getting and providing services among the three nodes forms a directed circle. In this scenario, a barter-based scheme limits the cooperation among the three nodes. Moreover, the works in [59][61] focus on unicast which is not a common communication mode for data dissemination.

The credit-based scheme was first applied in mobile ad hoc networks to solve the packet forwarding and routing problems [62][63] and multi-hop cellular networks [64][65]. In a credit-based scheme, a node can earn credits by helping others forwarding packets and then the node uses credits to rent others to help forwarding their own packets. The work in [66] proposes a credit-based scheme in delay tolerant networks for routing. The work in [66] also identifies and addresses the edge insertion attack and edge hiding attack in DTNs. Since

the credit-based scheme in [66] is mainly for routing, a trusted third party is required to provide payment services and both long-range and short-range radios are needed, each of which incurs more security complexity and hardware cost. The work in [78] is the first work that incorporates incentive stimulation into data dissemination in DTNs [47] with selfish nodes and multiple interest types. The incentive scheme employed in [78] is a credit-based scheme and the stimulation of nodal cooperation between a pair of nodes is transferred to a game theory problem and solved using the Nash Theorem. From the simulation results of [78], we found the dissemination ratio and delay is improved and the overhead is limited to a relatively low level. However, the work in [78] left some questions unanswered, which makes their incentive scheme unpractical. In the credit-based scheme of [78], the credits are not fluent and users try to earn credits with no reason, which makes it more like a reputation-based system without benefiting high-credit users. Moreover, it is not clear where the credits come from. If one node has no credits, there is no way for this node to reward others. Without considering these issues, the incentive scheme in [78] is not practical for real delay tolerant network applications. The work in [73] proposed a credit scheme to disseminate advertisements in MSNs. The advertisement packet is delicately designed to carry virtual check. The goal is to disseminate advertisements to a subset of destinations. The nodes who help the destination to get the advertisements will be cashed in the signed virtual check. The results indicate their scheme can effectively increase advertisement delivery ratio. However, it is only suitable for packets like advertisement containing virtual check. Moreover, the pay process in [73] is unreliable and may have a large delay. The virtual check can be cashed in if the node meets the advertisement disseminator or the node seeks others to help with passing the check to disseminator and returning the cash. Note that, both strategies result in a large delay or even fail if the round path disappears due to node mobility. In this proposal, in our nobel credit-based scheme, the aforementioned drawbacks will be overcome through a careful design.

2.2 Information Diffusion in Large Social Networks

Information diffusion is widely researched as to maximize the influence spread in social networks. Since influence maximization was firstly studied as an algorithmic problem by Domingos and Richardson using probabilistic method [4] [13], this problem has been extensively studied for over a dozen of years.

In [10], Kempe, Kleinberg, Tardos first formulated the influence maximization as a discrete optimization problem, which means selecting the most influential subset of nodes under certain cascade model, furthermore, they introduced an approximation algorithm with a provable approximation guarantee. By restricting computations and tuning the size of local influence region, [3] proposed a scalable heuristic algorithm that can be scaled up to deal with millions of nodes according to their results. Improved greedy algorithm and degree discount heuristic algorithm were introduced by [2]. Community greedy algorithm(CGA) was proposed by [15], the main idea behind(CGA) is to divide a network into communities, then find the most influential top-K nodes of these communities. However, all of these papers are simply based on independent cascade (IC) and linear threshold(LT) cascade model. As we know, even though IC and LT model are the most widely accepted models, IC and LT model are criticized for its imperfect reflection for the realistic influence propagation scenario, it will be better if there is new spread model that not only can accurately depict the realistic scenario but also be easily used to design efficient algorithm in social network.

In Recent years, more diffusion models have been proposed to describe the way in which influences and ideas spread through the social networks. Jung [8] using ICN (independent cascade negative) model to capture the feature of negative opinion, it is basically an extension of the IC model. Zhuang [16] studied the influence maximization problem under dynamic network, it is a more accurate diffusion environment, but they fail to define a new propagate model. An improved greedy algorithm was proposed in [2] to speed up the seed selection process, which is based on WC model. All of these model make sense to some extent and there are some very efficient algorithms designed above them. However, the human influence behavior is far more complex than the existing models. Therefore, in this paper, we try to

propose a more accurate influence model to bridge this gap, we believe our model is more human-influence descriptive than the existing diffusion models.

Besides, based on the proposed new influence model, for the first time in literature as we know, we could be able to minimize the diffusion time while achieving an estimated influence coverage.

Chapter 3

PREDICTION-BASED ROUTING WITH PACKET SCHEDULING UNDER TEMPORAL CONSTRAINT IN DELAY TOLERANT NETWORKS

3.1 Introduction

Delay- or Disruption-Tolerant Networks (DTNs) have attracted much attention recently. DTNs attempt to route messages via temporarily or intermittently connected nodes. Compared with Wireless Sensor Networks (WSNs) and Mobile Ad-hoc Networks (MANETs), both of which have been modeled as connected graphs with stable end-to-end paths even though paths may vary according to time [24][25][48][27][28][49][50][52][54][33], DTNs lack continuous connectivity and therefore the protocols for WSNs and MANETs may fail in DTNs.

The nodes carrying and relaying messages in a DTN are referred to as data mules. A typical example DTN is a university environment. With the advances of mobile data storage and delivery devices such as smart phones, PDAs, and laptops, people on campus carrying such devices can be modeled as data mules. People move from one building to another, sometimes following pre-assigned routines, thereby making the entire campus as an intermittently connected network - a DTN.

Most existing algorithms employing the store-carry-and-forward scheme simply fall into two categories [18]. One category of the algorithms use the flooding strategies. Flooding usually requires very limited pre-knowledge, or sometimes even no knowledge about the historical information of networks. The flooding strategies include Direct Contact, Two-hop Relay, Tree-based Flooding and Epidemic Routing. For Direct Contact, the source node does not forward the message until it meets the destination node in its moving trace. Two-hop Relay allows one relay before the message arrives at the destination node. The algorithm *Spray&Wait* employs this strategy. Tree-based Flooding extends Two-hop Relay by con-

structing a tree structure. Each level of the tree can be seen as a two-hop relay. Epidemic Routing is the most simple strategy which is extremely robust to the delivery success of each message with maximum redundancy. The disadvantage is that too many copies induce much energy cost and may occupy a lot of limited resources like buffer size, bandwidth and available contact opportunities, which makes this strategy the most unpopular one. Epidemic routing is usually an optimal algorithm compared with the routing algorithms aiming at maximizing delivery ratio. Another category of algorithms use the forwarding strategies. These algorithms usually make use of topology information to predict the best path for a specific message to arrive at the destination. According to [18], they are classified as the Location-based Routing [19], [21], [22], Gradient Routing [20] and Link Metrics Routing [21]. Location-based Routing requires very little network topology information [19], [22]. They usually assume that nodes may visit some places or coordinates in their moving traces and then predict the locations [21] where the contacts may happen or assign different locations with different priorities [22] to delivery the messages. Gradient Routing assigns different nodes with different priorities according to the suitability of delivering one specific message. The spirit of Link Metric Routing is more like traditional wireless networks routing protocols. They generate a contact graph and assign different weights to different links and then run a shortest path algorithm to predict the probability with which the message may arrive at the destination.

Since the most important performance metric in DTNs is delivery ratio and then delivery latency, we try to seek a new algorithm that can increase delivery ratio. The work in [23] indicates that adding a dose of altruism to a network can help with improving the overall delivery ratio. Meanwhile, it also brings us the questions like how to include the dose of altruism to maximize delivery ratio under the factors such as packet size, source, destination, TTL and *etc.*.

We propose a novel Prediction-based Routing algorithm with Packet Scheduling (*PRP-S*). We assume messages may have different TTLs, sources, and destinations. Our strategy consists of two main parts. The first part is to model the abilities of each node in deliv-

ering packets. The second part is to schedule the packets at every single node in order to maximize the overall delivery ratio. The modeling in the first part decides the precision of the scheduling in the second part. The *FirstComeFisrtServe* does not apply in the second part since we add the spirit of altruism to scheduling which may degrade the overall delivery latency but can improve delivery ratio than other pure Link Metrics algorithms. The main contributions are summarized as follows:

1. We model the abilities of each node to delivery packets through finding k-disjoint shortest paths with time constraint.

2. For each node, we transform the scheduling problem to the bipartite matching problem in order to optimally schedule packets at each node locally and increase the overall delivery ratio globally.

3. Extensive simulations are conducted on both synthetic DTN traces and real DTN traces such as INFOCOM06 [40] and SIGCOMM09 [41]. The results show that PRPS can increase the overall delivery ratio and the advantage becomes more obvious when in a time slot, the average number of packets at each node increases.

3.2 Problem Formulation & Network Model

We consider a typical DTN with a number of mobile nodes which has intermittent connectivity. All the nodes can periodically connect to a base station or a server which maintains the topology information and contact graph. The server can model the abilities for the nodes containing all kinds of packets with different packet information including TTL, source and destination. This assumption is reasonable for DTNs with mobile devices that can connect to a server on the internet or a base station that all nodes can connect with [21]. Since there are multiple factors such as buffer size, contact opportunities, bandwidth between each contact that can influence the overall delivery ratio, to simplify the problem, we assume there is no limited buffer size at each node and during each contact, only one

packet can be delivered. The goal of this paper is to schedule all the packets during each contact such that the best sequence to maximize the overall delivery ratio can be achieved.

More specifically, a network consists of n mobile nodes $V = \{v_1, v_2, \cdots, v_n\}$ and m edges $E = \{e_1, e_2, \cdots, e_m\}$. The value of e_i represents the contact probability of two nodes where $0 \leq e_i \leq 1$. The contact probability between a pair of nodes a and b at time i is c_{ab}^i, then the contact probability matrix at time i is denoted as C^i. There are t messages generated by all the nodes, denoted by $M = \{m_1, m_2, \cdots m_t\}$. We define the ith message as a 3-tuple $\langle m_{source}, m_{dest}, m_{TTL} \rangle$. Each message has only one copy in its lifetime. Our goal is to maximize the overall delivery ratio R,

$$R = \frac{\sum_{i=0}^{t} Arrive_{m_i}}{t},$$

where $Arrive_m = 1$ if message m_i arrives at the destination within its TTL, and 0 otherwise.

3.3 The PRPS Protocol

In this section, we present a novel Prediction-based Routing algorithm with Packet Scheduling (PRPS). We aim at increasing the overall delivery ratio which may sacrifice delivery time for some packages. PRPS consists of two phases. The first phase is to model the probability of each message arriving at its destination within its TTL. We call it ability graph in the following. The second phase is to schedule the packets in the pairs of nodes which may contact to achieve an optimal schedule at each node so as to increase the delivery ratio globally.

3.3.1 Ability Graph

Note that c_{ab}^i is the probability between a pair of nodes a and b in a time slot i. Usually it is called a *contact graph* or *contact file* in the previous works, *e.g.*, [21]. The work in [21] adopts a time homogeneous semi-Markov model to predict future contacts between each pair of nodes. In this paper, we do not intend to develop a new model for depicting the contact

graph. Similar with the approach in [35] [34] and [36], we model the contact process of each pair of nodes as a homogeneous Poisson process. The contact probability of two nodes does not vary with time. The random Poisson variable p_{ij} can be described as the number of events that happen between entities i and j within time interval τ. In our case, it indicates the number of meetings between nodes i and j within one time slot τ. That is

$$P\left[\left(N\left(t+\tau\right)-N\left(t\right)\right)=\mu\right]=\frac{e^{-\lambda t}(\lambda\tau)^{\mu}}{\mu!},\mu=0,1,\cdots$$

where $N\left(t+\tau\right)-N\left(t\right)=\mu$ is the number of contacts in the time interval $(t,t+\tau)$. When $\mu=0$, it means there is no event in the time interval τ. That is

$$P\left[\left(N\left(t+\tau\right)-N\left(t\right)\right)=0\right]=e^{-\lambda t}$$

where λ is the rate parameter of the Poisson process. For modeling the contact profile of nodes in a network, the contact rate between nodes a and b can be represented as λ_{ab}, which is considered as the given parameter since it can be achieved through historical information. Then the probability that nodes a and b do not meet within time τ is

$$q_{ab}=e^{-\lambda_{ab}\tau}.$$

Therefore, the probability that nodes a and b meet within time interval τ is

$$c_{ab}=1-q_{ab}.$$

This can depict the contact profile. Now we can model the ability graph. For a specific message m at node a with destination b and a TTL, the probability of m arriving at b is denoted as P_{ab}^{TTL}. The probability P_{ab}^{TTL} depends on three parameters, source, destination and TTL. So we can use a three-dimensional matrix with TTL as the first dimension, source as the second dimension and destination as the third dimension to represent all the

probabilities, which is called the ability graph.

Calculating P_{ab}^{TTL} requires the information of all the possible paths between nodes a and b. However, there might exist exponential number of paths, thus we cannot derive it in polynomial time. In order to speed up the process, we approximate P_{ab}^{TTL} by using limited number of disjoint paths between nodes a and b with length no more than TTL.

$$P_{ab}^{TTL} = 1 - \prod_{s=0}^{S} (1 - RP_s)$$

where S is the path set including at most k paths and RP_s is the probability of the *s-th* reachable path from node a to node b which is defined as the multiplication of all the edges' probabilities in the path [38]. *E.g.*, in Fig.3.2, two paths $\{e2, e3, e10, e11\}$ and $\{e1, e6, e8, e12, e14\}$ can be used as two paths to calculate P_{ab}^{TTL}.

Note that the problem of finding k shortest paths with limited length is an NP-hard problem if k increases to infinity. This problem can be reduced from the problem of finding the maximum number of shortest paths with bounded length which has been proven to be NP-Complete when the bounded length is larger than 4. To reduce the computation cost, we propose a heuristic algorithm to calculate P_{ab}^{TTL}. When the bounded length is greater than 4, we limit the given parameter k to find k shortest paths. Otherwise, we try to find the maximum number of shortest paths. The benefit of dynamically adjusting the strategies is that it can better approximate P_{ab}^{TTL}. With the decrement of TTL, the maximum number of shortest paths actually gives high priority to the messages which are going to expire. Below we show our heuristic algorithm (Algorithm 1) to find k shortest paths with bounded length.

3.3.2 Packet Scheduling Process

During one time slot, only one message can be forwarded. Under this assumption, our problem is to determine the best message from a to forward once there is a contact between node a and node b. Let $P_{a,i}^{\beta}$ be the probability that message m_i from node a reaches its destination within β time. For each m_i and a, we can obtain a vector

Algorithm 1: CONSTRUCTING AN ABILITY GRAPH

Input: A graph $G = (V, E)$, two distinct nodes a and b in G, and time constraint TTL

Output: The probability of successfully forwarding a message from a to b within TTL: P_{ab}^{TTL}

1 $S = \emptyset$.

2 $N_a = \{v \mid (a, v) \in E\}$. $S = \{(a, v) | v \in N_a\}$. Regard every edge in S as a path and $S = \{P_1, P_2, \cdots, P_{|S|}\}$.

3 $RP_\tau = max(RP_i)$ where RP_i is the reachable probability of path P_i and $P_i \in S$.

4 Set h to be the endpoint of P_τ. Let $N_h = \{v | (h, v) \in E\}$. Remove P_τ from S and add $P_\tau + (h, v)$ in S if $|P_\tau| < TTL$ where $v \in N_h$ and $|P_\tau|$ is the number of edges in P_τ.

5 Output $P_i \in S$ if the endpoint of P_i is b. Remove P_j from S if $\exists e(e \in P_i \wedge e \in P_j)$ where e is an edge in any path.

6 Repeat Step 3 and Step 4 until $S = \emptyset$, or k paths are found, or there are no more edges which can be added;

7 Calculate P_{ab}^{TTL}.

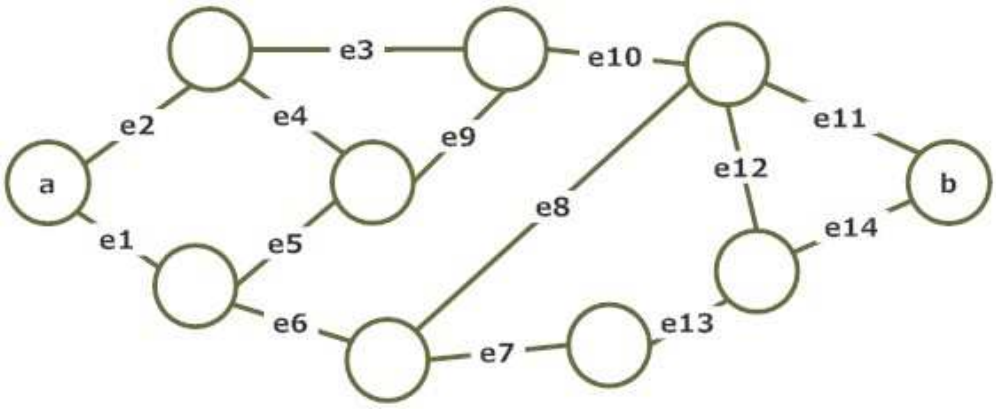


Figure 3.1. An example of finding k shortest paths.

$\left(P_{a,i}^{TTL}, P_{a,i}^{TTL-1}, P_{a,i}^{TTL-2}, \cdots, P_{a,i}^{1}\right)$. If there is a contact between node a and b, considering the probability as a utility gain, the utility function for node a can be defined as follow

$$U_{ab}(\omega) = \sum_{i=0}^{t} P_{b,i}^{TTL}\omega_{i,0} + \sum_{i=0}^{t}\sum_{j=1}^{TTL} P_{a,i}^{TTL-j}\omega_{i,j}$$

where t is number of messages in a. $U_{ab}(\omega)$ is the total expectation of successfully delivering messages given a schedule ω. $\omega_{i,j} = 1$ if message m_i is forwarded at the jth time and $\omega_{i,j} = 0$ otherwise. Without adding a dose of altruism, for fairness, most protocols adopt the strategy of first-come-first-serve (FIFS), which deals with all the messages according to their arrival or generation sequence without considering the temporal constraint of all the messages. A dose of altruism may lower the priorities of the messages who come first, but may increase the overall delivery ratio. We propose a greedy method and an optimal method to reflect the dose of altruism. Intuitively, for each contact between nodes a and b, the current forwarder a can greedily choose the message with the maximum increment of the differential probability between forwarding to node b and staying at node a. We call this the locally greedy solution since every forwarding helps the message arrive at a more proper forwarder. However, the greedy strategy is sometimes suboptimal because the forwarder only considers the gain in the current time slot during which a contact between nodes a and b happens. Consider a scenario where there are three messages m_1, m_2 and m_3 at node a with $TTL = 4$, $TTL = 4$ and $TTL = 3$, respectively. The priorities of m_1, m_2 and m_3 decrease according to their arrival time. A contact between nodes a and b happens. As shown in Table 3.1, if m_1 stays at node a, m_1's probability set is $(0.6, 0.5, 0.5, 0.4)$. If m_1 is forwarded to node b, m_1's probability set is $(0.65, 0.6, 0.5, 0.4)$. Similarly, if m_2 stays at node a, m_2's probability set is $(0.8, 0.8, 0.7, 0.6)$. If m_2 is forwarded to node b, m_2's probability set is $(1.0, 1.0, 1.0, 1.0)$. As for m_3, if m_3 stays at node a, m_3's probability set is $(0.5, 0.3, 0.1)$. If m_3 is forwarded to node b, m_3's probability set is $(0.5, 0.35, 0.3)$. The probability 1.0 means node b is the destination of the message. In this case, without adding the dose of altruism, the total expectation of successfully delivering messages m_1, m_2 and m_3 is $0.65 + 0.8 + 0.1 = 1.55$. Using the locally

greedy algorithm, the total expectation of successfully delivering messages m_1, m_2 and m_3 is $0.5+1.0+0.1=1.6$. However, the best assignment of global optimization can achieve the expectation of $0.5+0.8+0.5=1.8$.

	m_1		m_2		m_3	
TTL	a	b	a	b	a	b
4	0.6	0.65	0.8	1.0	0.5	0.5
3	0.5	0.6	0.8	1.0	0.3	0.35
2	0.5	0.5	0.7	1.0	0.1	0.3
1	0.4	0.4	0.6	1.0	N/A	N/A

Table 3.1. Example of Utility Expectation

From the above example, we can conclude the current forwarder a can achieve the maximum global utility U^*_{ab} by considering multiple time slots instead of only focusing on the current contact time slot. Define $\kappa = \max(TTL_{m1}, TTL_{m2}, \cdots, TTL_{mi}, \cdots)$ as the largest TTL of all the messages at node a. For each time slot in $(1, \kappa)$, $\omega_{i,j} = 1$ if message m_i is forwarded at time j, otherwise, $\omega_{i,j} = 0$. At each time slot, at most one message can be forwarded, therefore for any time slot j, $\sum_{i=0} \omega_{i,j} \leq 1$. Meanwhile, all messages cannot be forwarded more than once. Therefore, given any message m_i, $\sum_{j=0}^{\kappa} \omega_{i,j} \leq 1$. Now the maximum-utility scheduling problem for forwarder a when a contact happens between nodes a and b can then be formalized as follows:

$$\begin{aligned} U^*_{ab} &= \max_{\omega} U^*_{ab}(\omega) \\ &= \max_{\omega} \sum_{i=0}^{t} P^{TTL}_{b,i} \omega_{i,0} + \sum_{i=0}^{t} \sum_{j=1}^{TTL} P^{TTL-j}_{a,i} \omega_{i,j} \end{aligned}$$

subject to

$$\sum_{i=0} \omega_{i,j} \leq 1, \forall j \in TTL$$

$$\sum_{j=0}^{\kappa} \omega_{i,j} \leq 1, \forall i \in t$$

$$\omega_{i,j} \in \{0, 1\}$$

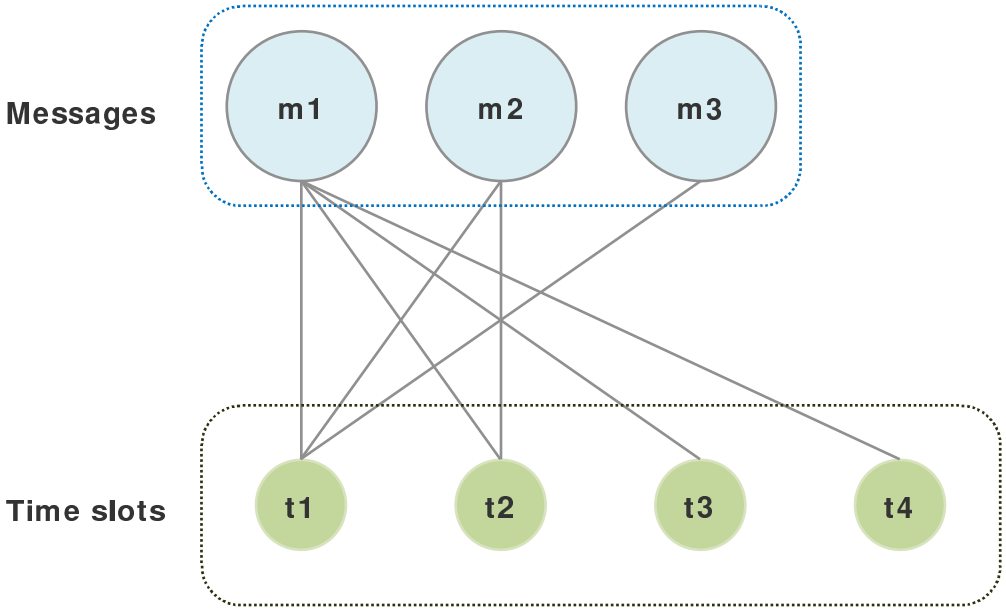


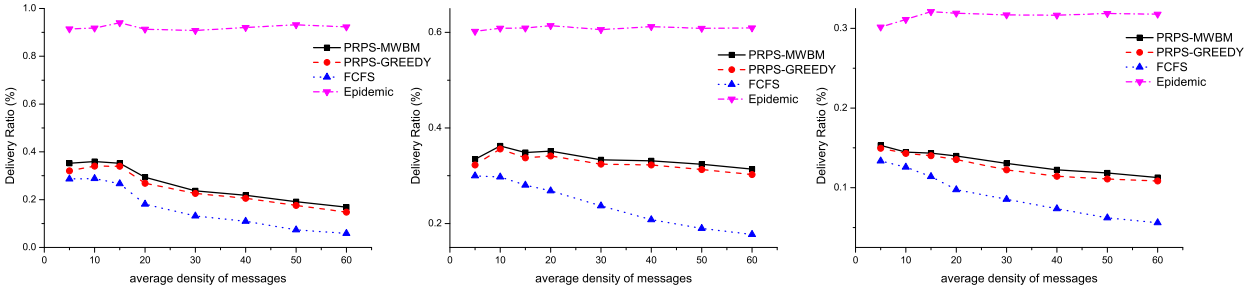
Figure 3.2. The optimal forwarding schedule problem can be transformed to the MWBM problem.

Hence, our objective is to find the optimal forwarding schedule ω so that forwarder a can achieve the maximal utility $U_{ab}^{*} = U_{ab}^{*}(\omega)$.

The optimal forwarding schedule problem can be transformed to the Maximum Weight Bipartite Matching (MWBM) problem [37]. Suppose that $G = (V, E)$ is a bipartite graph with vertex classes T and M, representing time slots and messages, respectively. M is a matching in the bipartite graph G, where $M \subseteq E$. As shown in Fig.3.2, there are three messages m_1, m_2, and m_3 with $TTL = 4, 2, 1$, respectively at node a when a contact happens between node a and b. Each edge e_t^i represents the utility that a specific message m_i can get if it is forwarded in time slot t. For the first time slot $t = 1$, since each contact related to node a is assured, then each message can only be forwarded to the nodes that meet a. For the time slot $t > 1$, the contacts are probabilistic and uncertain. e_t^i represents the probability that a message is forwarded from node a. Therefore, the forwarding schedule ω is a matching M in the bipartite graph G. Hence, finding the maximal utility problem is equivalent to solving the MWBM problem in a bipartite graph G. The MWBM problem can be solved in polynomial time and we apply the classic Hungarian algorithm [39] to solve it. In our case,

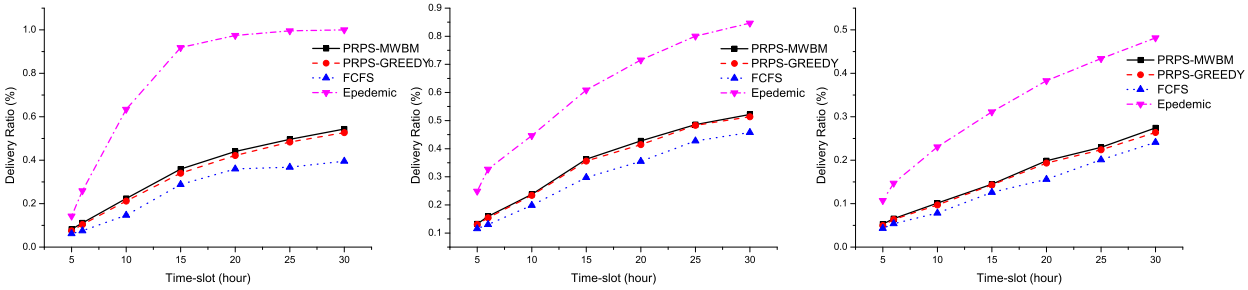
we use the Hungarian algorithm to find the optimal forwarding schedule ω^*.

3.4 Performance Evaluation



(a) Synthetic trace (b) INFOCOM06 (c) SIGCOMM09

Figure 3.3. Comparison of delivery ratio on different data sets with Maximum $TTL = 15$.



(a) Synthetic trace (b) INFOCOM06 (c) SIGCOMM09

Figure 3.4. Comparison of delivery ratio on different data sets with average density of messages $\sigma = 10$.

In this section, we evaluate PRPS using our own simulated synthetic trace and two real traces INFOCOM06 [40] and SIGCOMM09 [41]. Discrete time is used in our simulations. After modeling the ability graph in Section III.A, we present the locally greedy algorithm and the optimal algorithm which are named as PRPS-GREEDY and PRPS-MWBM in the following, respectively. We compare PRPS-GREEDY and PRPS-MWBM against an algorithm that processes messages without considering the time constraint of the messages and

processes messages according to messages' arrival time, which is named as FCFS. For fairness, FCFS uses the same prediction-based scheme and ability graph that PRPS-GREEDY and PRPS-MWBM use. Epidemic routing is also considered in our simulations that can generate optimal solutions in evaluating delivery ratio and delivery latency.

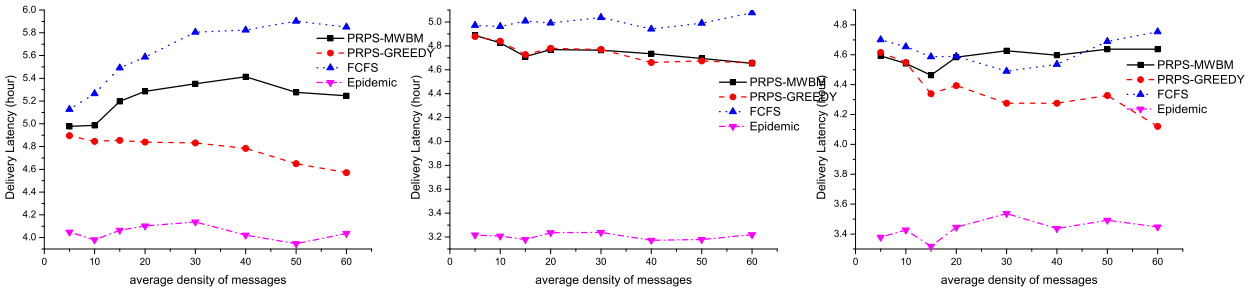
3.4.1 Simulation Settings

We used java to implement a custom packet-based simulator that can simulate the topology of a DTN. As mentioned in Section III.A, the contact processes of most DTNs follow the Poisson process [34]. Therefore, we could adjust the contact profile by adjusting the parameter λ. For the density of a graph, we used an average degree d to control the number of nodes that a specific node could meet with a probability. Based on the contact probability of each pair, we randomly generated the future contacts in each time slot. All the messages were generated at all the nodes with a TTL whose range was from 0 to the maximum TTL. We defined a parameter σ as the average density of the messages at one node. Assume there were t messages in the simulation lifetime ζ, then

$$\sigma = \frac{t \cdot TTL}{\zeta}$$

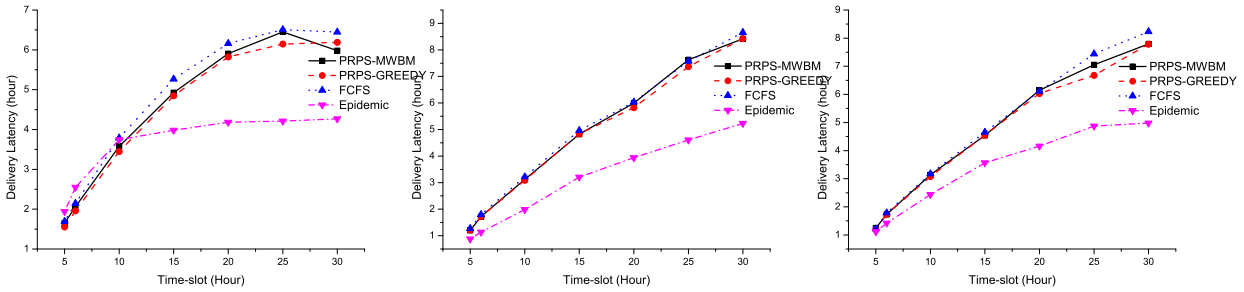
In the experiments, we adjusted TTL and σ to compared the results of the above mentioned algorithms.

The real trace INFOCOM06 involves 78 users who were student volunteers in the conference INFOCOM 2006 and each of them carried a device that had a short radio range. The contacts during 4 days were recorded in the INFOCOM06 trace. Similarly, the real trace SIGCOM09 involved 76 users in the conference SIGCOMM 2009. The social profiles of the participants were also included. All the messages were generated in the same way as in the synthetic trace.



(a) Synthetic trace (b) INFOCOM06 (c) SIGCOMM09

Figure 3.5. Comparison of delivery latency on different data sets with Maximum $TTL = 15$.



(a) Synthetic trace (b) INFOCOM06 (c) SIGCOMM09

Figure 3.6. Comparison of delivery latency on different data sets with average density of messages $\sigma = 10$.

3.4.2 Results

In this subsection, we analyze the results of the simulations on three data sets: our synthetic trace, INFOCOM06 and SIGCOMM09. Fig.3.3 plots the delivery ratio under different average message density σ for the three different DTN data sets. It shows that PRPS-GREEDY and PRPS-MWBM are 10% better than FCFS on average on our synthetic trace and the INFOCOM06 trace. Also, for our synthetic trace and the INFOCOM06 trace, we find that when $\sigma \leq 10$, as σ increases, the delivery ratio of all the algorithms increase. However, if $\sigma > 10$, except for the epidemic algorithm, the delivery ratio of the other three algorithms decrease. The SIGCOMM09 trace may have a lower value of σ. The reason is that a proper increment of the messages in a network can increase the probability that nodes can deliver part of the messages to a more proper relay node, but high message density brings much pressure to each node and surpasses the ability of the nodes to deliver all the messages with their $TTLs$, *i.e.*, too many messages result in congestions. Fig.3.3 also shows that with the increasing of σ, PRPS-GREEDY and PRPS-MWBM perform much better than FCFS and the difference between PRPS-GREEDY and PRPS-MWBM also becomes slightly bigger since the packet forwarding schedule behaves better facing more messages at a specific node.

Fig.4.4 shows the delivery ratio under different maximum TTLs for the three traces. We set $\sigma = 15$. With the increasing of TTL, the delivery ratio of all the algorithms increase as expected. When the maximum TTL is low, *e.g.*, $TTL = 5$, even the delivery ratio of the epidemic algorithm is also very low. This is because even during each contact, all the messages can be copied to other nodes. The contacts are limited and messages may have no chance to reach the destination nodes.

We also conducted two sets of simulations to show the overall delivery latency comparisons among the three data sets. Fig.6.8 and Fig.3.6 show the results. In Fig.6.8 where the maximum $TTL = 15$, with the increasing of the average message density, except for the epidemic algorithm, the overall latency of the other three algorithms first increases and then decreases. The phenomenon is similar with the one in the delivery ratio simulations. It is the result of message congestions. In Fig.3.6, $\sigma = 10$. It shows that even though PRPS-

GREEDY and PRPS-MWBM have increased the delivery ratio, especially PRPS-MWBM has a better performance, the overall delivery latency of PRPS-GREEDY and PRPS-MWBM has not been degraded.

Chapter 4

SOCIAL-AWARE DATA DISSEMINATION SERVICE IN MOBILE SOCIAL NETWORK WITH CONTROLLED OVERHEAD

4.1 Introduction

Mobile Social Networks (MSNs) have become an emerging wireless communication techniques with the explosive increment of smart devices like smart phones and pads in people's daily life. In order to alleviate the daily growing needs of bandwidth of cellular networks, information sharing among mobile users through short-range radios communications like Bluetooth and Wi-Fi is highly encouraged. Through single or multiple short-range communications among the mobile users in MSN, users' interested data can spread over MSNs in a delay tolerant way. However, for a given message with some interest, it is not easy to disseminate it to its targeted users (who are interested in) directly or indirectly in an efficient way. The reason is that data dissemination in MSNs suffers from the intermittent connectivity and unpredictable node mobility among mobile users, which makes the user contact opportunities very cherishable. Also, a successfully delivered message may possibly require multiple relays from the users that are not interested in it and the challenges is whether the intermediate users are willingly to carry such messages or not. Even if they do, what messages should they store, carry and forward is the problem since it is not realistic to store and carry all messages due to storage capacity and bandwidth of the mobile devices. In this work, we try to maximize the delivery ratio by optimizing the message forwarding through learning the mobility pattern of mobile users and interest distribution with a new defined overhead that we could control. Along with over a decade of research on MSNs, different architectures have been developed and considered in existing works. Here we introduce the main architectures to model MSNs. In summary, MSNs have three kinds of architectures: centralized, distributed and hybrid architectures [42]. In the centralized architecture, mobile

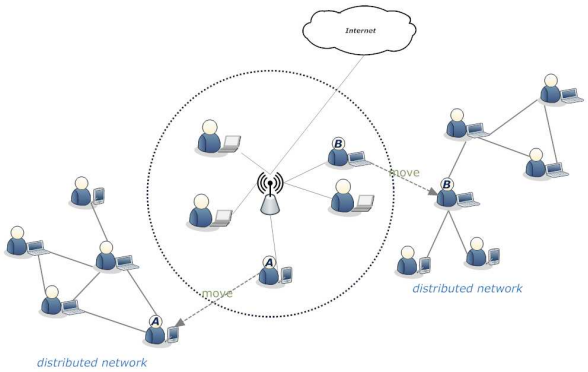


Figure 4.1. An MSN with the hybrid architecture.

nodes are all connected with Access Points (APs) and communicate with each other in a client-server manner. In the distributed architecture, there is no AP and mobile nodes only communicate with each other using short-range ratio. There is no message disseminator and messages are generated by all mobile nodes. The hybrid architecture is a mixture of the previous two architectures. In the hybrid architecture, most messages are disseminated from APs and only a certain number of mobile nodes have uncertain access to APs and these mobile nodes further disseminate messages to the rest of the nodes in a network using short-range radios. The hybrid architecture is the most realistic and commonly studied one in research and we also apply this architecture in this work. In the hybrid architecture, two mobile users forward messages to each other only when there is a direct contact between them, *i.e.,* two mobile nodes need to be in each other's short radio range to carry out a message exchange. A message is disseminated in a delay-tolerant manner and has high risks of delivery failure. A message may be dropped before it finally reaches its prospective receivers. Fig.4.1 shows an instance of data dissemination in hybrid architecture, where node A and node B have access to APs and when they move out of the access range of the APs, they further disseminate messages to other nodes in a distributed manner.

Considering the limited contact opportunity, contact duration, dynamic mobility and diversity of user interests, increasing overall data dissemination ratio becomes the primary goal for most previous works [80][81][82][83][36][87][88][35]. Epidemic data dissemination can achieve the highest dissemination ratio, while on the other hand, as well as highest

overhead for any given network. It is unrealistic since mobile nodes may quickly deplete their limited resources if they store and carry every single message. Except the epidemic model, many dissemination approaches are formulated as optimization problems, in which it is assumed that mobile users are cooperative [36][87][88][35]. Considering mobile nodes' selfish nature for message dissemination, some works discussed how to use credit or virtual money to stimulate nodes to be cooperative to help others disseminate their interested messages [79][78]. Although all previous works [92][35][80][81][82][83] provide an overall overhead for a given setting of network parameters, the overhead for a specific kind of messages is not well estimated. With the growing size of a network, the overhead may change a lot and become very uncertain. The overhead of data dissemination is not trivial to estimate and control considering the traditional definition of overhead in the data dissemination problem in MSNs. In general, the traditional definition of overhead in data dissemination problem is defined as the number of all the messages relayed and accepted in a network over the number of the messages accepted by the nodes with corresponding interests. Therefore, it is hard to control the specific number of message copies for a specific message. It becomes natural to define the overhead in this way when we treat all mobile nodes as normal store-and-relay nodes without distinctions. However, in the hybrid architecture, the group of nodes with access to APs can get their interested messages directly with relatively much smaller delay than the other nodes. Here, we denote them as the *first-level* nodes. For the nodes without access to APs, but with access to the *first-level* nodes, are denoted as the *secondary-level* nodes, they can only get their interested messages from the *first-level* nodes. For the *secondary-level* nodes, the *first-level* nodes are just like mobile APs. The difference is that the mobile APs may carry limited messages of some interests that have already been delayed to some extent. When some secondary-level nodes fetch uninterested messages from first-level nodes willingly, they can also play the role as mobile APs to further serve lower level nodes. If we treat the mobile nodes when they become mobile APs as mobile servers to disseminate messages, we can simply limit the number of active servers in a network to control overhead. For a given kind of interest, mobile nodes always store and carry their

interested messages and willingly forward messages to their contacts with same interests. Under this scenario, mobile nodes only carry-store-forward their own interested messages and Forwarding messages among users does not incur any overhead since every message forwarding delivers a message to its targeted users. This kind of data dissemination based on same interest is referred as "SelfCast"in this paper. When we consider a mobile node may store and carry an uninterested message, the mobile node can play two roles. When a mobile node carries an uninterested message, it plays as a server to disseminate this message. When mobile nodes get their interested messages from a server, they are considered as clients as to this interest. in this work, the concept of server is limited to the nodes who carries and spreads their uninterested messages to other nodes, e.g., when a client A receives an uninterested message from a server B, as to the message, A turns from a client to a server to further disseminate this message to other mobile nodes who are interested in the message. On the other hand, B turns into a normal client and meanwhile it stops serving this message anymore by removing the message from B's storage. As to a specific message, the number of servers in the network equals the number of message copies that co-exist in the network and are carries by the nodes who are not interested in the contents. Normally speaking, the larger the number of servers for a message is, the better the dissemination ratio can achieve. If the number of servers for a message is not limited, the dissemination for this message is as same as epidemic dissemination. Therefore, for a given message, how to select a certain number of nodes at different time to be serve as the message's servers is the main problem. Fig.4.2 shows one example how a server node may serve other nodes and the transition of the roles between client and server. There are several groups of users and individuals in this figure. Individuals marked with letter A, B, C and D are plotted with their moving track and how they meet others. Also, each group and individual are labeled with their interest. For simplicity in the example, only one interest is labeled for each group and individual. All groups and individuals in the figure have contacts with node A and therefore we starts from A's moving track. Along node A's moving track, A may meet different nodes with the same or different interests. A can disseminate its carried messages to the nodes who have the same

interest without any overhead. We assume A carries messages about movies when it starts. When A moves, it firstly meets the nodes which are interested in movies. A disseminates its carried movie message to them. After that, A meets node B and node C. Since A probably would not meet other nodes who are interested in movies through model prediction but node C would, A forwards the movie message copy to C for further dissemination. In this case, C becomes the server of this movie message and is responsible for further forwarding the movie message and A stops serving movie message by dropping this movie message. Meanwhile, A can become a health message server by fetching a health message from B. Similar scenario occurs when A fetches business messages from node D in order to better serve A's future encounters.

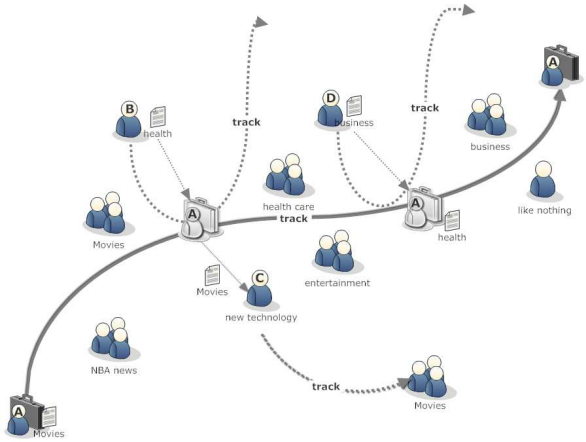


Figure 4.2. Example of how server nodes disseminate messages.

Considering the case that nodes may have two roles in a network for any given message, it is reasonable to define overhead based on the number of active servers for each given message. More formally, we denote m_i as the ith message and κ_{m_i} as the server copies that can co-exist in the network. Assume the message set is denoted as M, then the overhead is $overhead = \sum_{i=0}^{|M|} \kappa_{m_i}$. Unlike traditional overhead which highly depends on how the message disseminates, network mobility and interest distribution, the new defined overhead is fixed and is well controlled by limiting κ_{m_i} and will not overwhelm the network bandwidth and storage capacity. We denote this overhead as *service overhead* in the rest of the paper.

In section 6.3, the *service overhead* is discussed in details. We try to maximize the data dissemination ratio with controlled *service overhead*. Consider the following scenario, for message m_i, once server node A meets another node B which may serve as a better server for this message, A passes this message to B for future dissemination of m_i, and A terminates its service for m_i and drops m_i. A better evaluation metric may also consider how long a node serves others while consuming its own buffer resources. Limiting the number of alive copies of messages is easy to implement in a network. Moreover, when incorporating an incentive scheme [79][78], it is better and easier to decide the cost of an advertisement dissemination based on overhead it brings to the network. Compared with conventional overhead, our proposed overhead makes communication cost less important, while bringing more fairness to the messages that need to be disseminated. It may also become easier to implement incentive schemes with controlled overhead. A message being paid higher or with higher priority can simply be authorized more alive copies, *i.e.*, more alive servers to disseminate this message.

Considering that nodes may have two roles: servers and clients, in a network, the key issue to optimize data dissemination is the selection strategy of servers for the messages of a particular kind of interest. For message m_i, an ideal server is a mobile node which is not interested in m_i while most of its encounters are. When a server should terminate its service for m_i and pass it to the next server with better service ability is the key concern. In this paper, we adopt the time-homogeneous semi-Markov model to predict the service quality of each mobile node for a given kind of interest in a given time slot.

The contributions of this paper mainly includes the following aspects.

1. We pointed out the disadvantages of traditional overhead in data dissemination problem and come up with a fixed and controllable *service overhead* which is fair as to messages and avoid the risks of overwhelming the network.

2. According to the *service overhead* defined in this paper, we further use a semi-Markov model to define the neighbor interest transition probability functions for each mobile user to predict the most appropriate node to serve each message according to nodes'

mobility and interest condition in order to maximize the overall delivery ratio with the limited *service overhead*.

3. Both synthetic and real data traces are used to test our new data dissemination scheme. And the extensive simulation results show our new data dissemination model performs over 10% in average better than the existing data dissemination schemes which can incorporates *service overhead* as well.

4.2 System Model

In this section, we first formally define the *service overhead* and formulate the data dissemination problem under the limited *service overhead*. Then we propose two service utility functions which incorporates time-homogeneous semi-Markov model to maximize data dissemination service for each message.

4.2.1 Problem Formulation and Assumptions

We consider the data dissemination problem under the hybrid architecture. In general, the hybrid architecture of an MSN consists of APs and mobile users. Normally, the APs are understood as static places such as cafeteria. In this paper, we assume when mobile users are able to first disseminate messages to network, they can be considered as APs. We consider an MSN with a set of users V, a set of APs Λ with $\Lambda \subset V$, and a set of interest types Γ and $\rho =\mid \Gamma \mid$. Any two APs λ_i and λ_j in the AP set Λ share the same messages. Each message m_i only corresponds to one type of interest. Each user has zero or more interests. For user v_i, we denote its interests as a set $\Gamma_{v_i} = \{\gamma^0_{v_i}, \gamma^1_{v_i}, \cdots, \gamma^\rho_{v_i}\}$. If user v_i is interested in interest type j , $\gamma^j_{v_i} = 1$, otherwise, $\gamma^j_{v_i} = 0$. If message m_i's interest type is j and $\gamma^j_{v_i} = 1$, user v_i will receive, store and carry message m_i once it gets the chance. User v_i may further disseminate this message m_i to its encounters N_{v_i}. As to interest type j, the encounters who share the same interest type j are denoted as $N'_{v_i,j}$, the encounters who do not share the interest type j are denoted as $\bar{N}'_{v_i,j}$. In the following, we may refer to encounters as

neighbors as well. Moreover, for node v_i, the neighbors at different time slot may change to some extent or are totally different at two different time slots. Therefore, $N_{v_i}^t$, $N\prime_{v_i,j}^t$ and $\bar{N}\prime_{v_i,j}^t$ represent neighbor set, interested neighbor set and uninterested neighbor set at time slot t, respectively. For any message m_i, it will be dropped from all the users once its TTL expires. To simplify the problem, the buffers of nodes are assumed as unlimited and there is no difference in consuming buffers between storing its interested messages and its uninterested messages. Each message m_i has κ_{m_i} authorized copies and these copies of message m_i can only be first achieved from the APs. Assume m_i's interest type is j, if user v_i's interest type $\gamma_{v_i}^j = 1$, user v_i can get message m_i without consuming the authorized copies and this kind of forwarding does not incur any overhead. In contrast, if $\gamma_{v_i}^j = 0$, user v_i will get one authorized copy of message m_i in the first-come-first-server manner. Once a user carries an uninterested copy of message m_i, it becomes a server of message m_i and will help disseminate message m_i with interest type j to its encounter set $N\prime_{v_i,j}$. For message m_i, κ_{m_i} represents how many servers of message m_i can co-exist in the network, *i.e.,* it also means the potential *service overhead* of disseminating message m_i to the network. The benefits of defining overhead in this way is that the limited bandwidth of mobile network and buffer resources will not be overwhelmed and efficiency of bandwidth can be higher with limited server copies. More specifically, we define the overall *service overhead* as below:

$$G = \sum_{i=1} \kappa_{m_i} \tag{4.1}$$

For message m_i, κ_{m_i} means the maximum serving nodes in the network. Therefore, deciding which nodes playing the servers for message m_i is the key problem in this paper. From a perspective of a specific node who plays as a server for message m_i, when it meets another potential server, whether transferring the service to the new server is the problem. We refer to this as *service forwarding scheduling problem.* For any given time slot τ, the servers for this message is denoted as $\Phi_{m_i}^\tau$ and $\Phi_{m_i}^\tau \subset V$. Also, we define s_{m_i} as the publishing time of message m_i, then it is only meaningful to calculate the server set of message m_i for the time slot τ which satisfies $s_{m_i} < \tau < s_{m_i} + TTL$. For any given time slot τ, all contacts

happened at this time slot are denoted as a symmetrical matrix C^{τ}. $C^{\tau}_{v_l,v_k} = 1$ means at time τ, v_l and v_k has a contact and $C^{\tau}_{v_k,v_l} = 1$ has the same meaning. For any server node $v_l \in \Phi^{\tau}_{m_i}$, at time slot τ, for any other node v_k, if $c^{\tau}_{v_l,v_k} = 1$ and $\gamma^{j}_{v_k} = 0$, then v_k is the potential server for message m_i with interest type j. We define $U^{j}_{v_k}$ as v_k's service ability of disseminating messages with interest type j. Whether v_l will forward the service of disseminating message m_i with interest type j to v_k depends on their service ability $U^{j}_{v_l}$ and $U^{j}_{v_k}$. To further derive $U^{j}_{v_k}$ and deciding the manners to exchange servers in a contact, we propose the service utility model. The service utility model is based on a discrete time-homogeneous Markov model.

4.2.2 Service Utility Model

We evaluate the node service ability using a time-homogeneous semi-Markov model. We use ($S^{v_i}_n$, $T^{v_i}_n$) to represent the model. The state of $S^{v_i}_n$ stands for a standard Markov model state. The states of $S^{v_i}_n$ means the *appetite* of v_i's neighbors N_{v_i}, *i.e.*, the most interesting message type that most neighbors like. The number of interest types is ρ, therefore, there are totally ρ states for each node v_i. We refer to the state as service state. State transition from $S^{v_i}_n$ to $S^{v_i}_{n+1}$ is independent from $S^{v_i}_{n-1}$ to $S^{v_i}_n$. $T^{v_i}_n$ is the time slot of state $S^{v_i}_n$. $T^{v_i}_{n+1} - T^{v_i}_n$ is the duration of state $S^{v_i}_n$, *i.e.*, if $S^{v_i}_n = j$, $T^{v_i}_{n+1} - T^{v_i}_n$ means how long the node v_i is prone to provide service of interest type j to its neighbors.

We define the time-homogeneous Markov state transition process as below.

$$\phi^{v_i}_{j,k}(t) = P(S^{v_i}_{n+1} = k, T^{v_i}_{n+1} - T^{v_i}_n = t \mid S^{v_i}_n = j) \tag{4.2}$$

Equation.4.2 defines the that v_i's service preference changes from state j to k after staying at state j for time duration t.

From equation.4.2, the standard Markov process without considering time series can be derived as below.

$$\varphi^{v_i}_{j,k} = P(S^{v_i}_{n+1} = k \mid S^{v_i}_n = j) = \sum_{t=1}^{\infty} \phi^{v_i}_{j,k}(t) \tag{4.3}$$

Equation.4.3 can be described as a two dimensional probability transition matrix. Denoted as φ^{v_i}.

To quantify the duration of each state before it jumps to other states, we define the duration time as below.

$$D_j^{v_i}(t) = P(T_{n+1}^{v_i} - T_n^{v_i} = t \mid S_n^{v_i} = j) = \sum_{k=1}^{\rho} \phi_{j,k}^{v_i}(t) \tag{4.4}$$

Equation.4.4 means after staying at interest type j for t time slots, v_i finally changes its service preference to other interest types. We denote v_i's duration time distribution matrix as D^{v_i}.

Assuming that state probability transition process is independent from the duration time probability distribution, then we can represent $\varphi_{j,k}^{v_i}$ as in equation.4.5.

$$\begin{aligned} &\phi_{j,k}^{v_i}(t) = P(S_{n+1}^{v_i} = k, T_{n+1}^{v_i} - T_n^{v_i} = t \mid S_n^{v_i} = j) \\ &= P(S_{n+1}^{v_i} = k \mid S_n^{v_i} = j) \cdot P(T_{n+1}^{v_i} - T_n^{v_i} = t \mid S_n^{v_i} = j) \\ &= \varphi_{j,k}^{v_i} \cdot D_j^{v_i}(t) \end{aligned} \tag{4.5}$$

$\phi_{j,k}^{v_i}(t)$ depicts how long it takes to v_i's service preference from interest type j to k. Given that at a relative time slot $t = 0$, v_i's state is j, then the probability of changing v_i's state to k after t time slots is defined as below:

$$\psi_{j,k}^{v_i}(t) = \sum_{l=l}^{t} \sum_{q=1}^{\rho} \phi_{j,q}^{v_i}(l) * \psi_{q,k}^{v_i}(t-l) \tag{4.6}$$

In equation.4.6, $\psi_{j,k}^{v_i}(t)$ is an iterative function which means after zero or more transitions during time period t, v_i's state finally changes to state k. In following, we refer to equation.4.6 as service preference function. Note that when $k = j$, it means the state does not change in $\phi_{j,k}^{v_i}(t)$ or the state may change to others from j and finally change back to j in $\psi_{j,k}^{v_i}(t)$. Given that at time slot t_j, v_i's state changes to k from j through zero or more transitions can be represented as below.

$$\psi_{j,k}^{v_i}(t-t_j) = \sum_{l=t_j}^{t} \sum_{q=1}^{\rho} \phi_{j,q}^{v_i}(l) * \psi_{q,k}^{v_i}(t-l-t_j) \tag{4.7}$$

The difference between equation.4.6 and equation.4.7 is that equation.4.6 depicts the transitions from zero time slot and equation.4.7 depicts given any state j and its corresponding time slot t_j, the transitions from time t_j.

4.2.3 Metrics Estimation

Given a start time slot t_i, service preference function $\psi_{j,k}^{v_i}(t-t_i)$ can depict the service preference of v_i at time slot t, *i.e.*, we know what kind of interest type of messages that v_i would like to serve to its neighbors N_{v_i} at time slot t. In order to evaluate the service preference function, two parameters are important and should be first quantified.

The first parameter is a two dimensional probability transition matrix for each node φ^{v_i}. $\varphi_{j,k}^{v_i}$ is the probability that v_i's service preference changes from interest type j to interest type k.

Before defining $\varphi_{j,k}^{v_i}$, we first define $pre^{v_i}(t)$ in equation.4.8, which represents at time slot t, which interest type is most needed by v_i's neighbors.

$$pre^{v_i}(t) = \arg\max_{j \in \Gamma} \frac{\sum\limits_{n \in N_{v_i}^t} \gamma_n^j}{\sum\limits_{j \in \Gamma} \sum\limits_{n \in N_{v_i}^t} \gamma_n^j} \tag{4.8}$$

$pre^{v_i}(t) = j$ means at time t, v_i's service preference is interest type j. We further define a Kronecker delta function as below.

$$\delta_{jk}^{v_i}(t) = \begin{cases} 1 & \text{if } pre^{v_i}(t) = j \text{ and } pre^{v_i}(t+1) = k \\ 0 & \text{otherwise} \end{cases} \tag{4.9}$$

$\delta_{ij}^{v_i}(t)$ means at time t, whether v_i's service preference changes from interest type j to interest type k.

Now we can define the state transition probability matrix as below:

$$\varphi_{j,k}^{v_i} = \frac{\sum_{t \in T} \delta_{jk}^{v_i}(t)}{T} \tag{4.10}$$

In equation.4.10, T is a relatively long training time period, in the way of discrete time, containing T time slots, also meaning there are potential T state transition chances. Then in probability of statistics, when $T \to \infty$, $\varphi_{j,k}^{v_i}$ is the probability of v_i's service preference changing from interest type j to interest type k. In real application, using a period of training data, equation.4.10 can be easily computed and updated using an Exponentially Weighted Moving Average (EWMA) process.

Another parameter for calculating equation.4.2 is the duration time distribution matrix. D^{v_i} represents state transition probability in time scale. $D_j^{v_i}(t)$ depicts how long v_i will stay in state j before it leaves for other interest state. To quantify the duration time distribution matrix D^{v_i}, we can statistically calculate chances of each state for node v_i in a relative long period to represent its state transition probability. D^{v_i} can be calculated through the following equation.

$$D_j^{v_i}(t) = \prod_{n=1}^{t} P(pre^{v_i}(\tau + n) = j) \cdot P(pre^{v_i}(\tau + t + 1) \neq j), \forall \tau \in T \tag{4.11}$$

In statistics, the calculation process of equation.4.11 can be easily computed. T is a relatively long period. When calculating, we simply count all transitions that happened during time period T, *e.g.*, for node v_i, if there are 5 transitions from state j to other states during time period T and 2 of 5 happens after staying at state j for 3 time slots, $D_j^{v_i}(3) = 2/5 = 0.4$. Note that, the longer training period T is, the more accurate the distribution matrix D^{v_i} can be.

4.2.4 Service ability

In this section, we define two utility functions to evaluate each node's service ability. In section.4.2.2, we have defined the interest service function in equation.4.6 and equation.4.7.

Through the interest service function equation.4.7, given an initial state j at time slot t_j of node v_i, we can achieve v_i's state of interest type at any time slot t, *i.e.,* we know which kind of message of v_i is more willing to serve at a given time slot. Assuming m_i has κ_{m_i} authorized copies. To maximize the service of each message m_i, each copy should always be served by a better server in order to achieve a better QoS. From the perspective of message forwarding, for message m_i with interest type j, if v_i is currently serving this message, when v_i meets v_k with $\gamma^j_{v_k} = 0$, whether v_i should forward the copy of message m_i to v_k depends on the utility functions we defined. Below we define two utility functions to evaluate the service ability according to different time scale.

Utility Function 1 We define the utility function 1 to evaluate node v_i's service ability of interest type j at current and next one time slot. It can also be understood as evaluating v_i's immediate service ability towards interest type j.

Before we define *utility function 1*, for simplicity, we re-represent equation.4.5. With a given initial state j of v_i at time slot t_j, we can derive $\psi^{v_i}_{j,k}(t - t_j)$ at time slot $t - t_j$. Now we make t_j as the relative 0 time slot, then we simply use $\psi^{v_i}_k(t)$ to represent that interest serving preference of v_i is interest type k at relative time slot t. Then *utility function 1* comes as below.

$$U_1{}^j_{v_i}(t) = (\psi^{v_i}_j(t) + \psi^{v_i}_j(t+1)) * N\prime_{v_i,j} \tag{4.12}$$

$N\prime_{v_i,k}$ is the number of neighbors of v_i who are interested in interest type k. $N\prime_{v_i,k}$ can also be learned in the data training phase. Note that in our discrete time model, when node v_i meets node v_k at time slot t, it does not exclude other contacts that v_i may have. We address the contacts in one time slot sequentially, therefore the order of encounters matters. To roughly evaluate each node's immediate service ability, we combine the current time slot and next time slot.

Utility Function 2 *Utility function 1* is defined from the perspective of providing better service in short future. In order to evaluate the overall service ability in messages' residual TTL, we define another utility function as below:

$$U_{2_{v_i}}^{j}(t) = \sum_{n=0}^{TTL} (\psi_j^{v_i}(t+n)) * N\prime_{v_i,j} \tag{4.13}$$

Utility function 2 compares different servers according to their potential service before the disseminated message's TTL expires.

In comparison, *utility function 1* is the greedy way for short-term benefit. *Utility function 2* always seeks an overall better server during the message's lifetime. *Utility function 1* may be a better metric compared to *utility function 2* if node v_i's neighbors N_{v_i} interest type changes a lot in the time order and *Utility function 1* can effectively capture the changes and transfer the message service role more dynamically. However, *utility function 1* may suffer more from the prediction inaccuracy which my incur due to the dynamic mobility property of the network.

4.2.5 Data Dissemination Process

Using one of the utility functions, we can decide at a give time slot τ, which node can be a better server for a message with one interest. Therefore, during each contact at time τ, we can decide whether the server copy should be forwarded to another node for each pair of contacts. We describe the data dissemination process and node selection procedure in algorithm.1.

In the procedure, lines 5-11 describe how messages are first disseminated from APs. If the node who accesses the AP is interested in the message, the node will store and carry the message willingly, otherwise, the node becomes one server node who serves to disseminate this message to the rest of the network. There are totally κ copies and nodes get the copies in the first-come-first-get order. Lines 12-22 describe how to disseminate messages during each contact and how to transfer server role to another node for each pair of nodes in contact. Lines 23-26 checks whether a message has expired or not.

Algorithm 2: Data dissemination and Server node selection procedure

Input: Set of nodes V, set of APs Λ, message set M, original messages hold in APs Λ_M, time slot τ, $\forall v_i \in V - \Lambda$, Γ_{v_i}, $N^t_{v_i}$, $N\prime^t_{v_i,j}$, $\bar{N}\prime^t_{v_i,j}$, authorized message copies hold in all users v_{iM}, $\forall m_i \in M$, κ_{m_i}, s_{m_i}

Output: $\forall m_i \in M$ and $\forall \tau$ with $\tau - s_{m_i} < TTL$, the set of servers $\Phi^\tau_{m_i}$

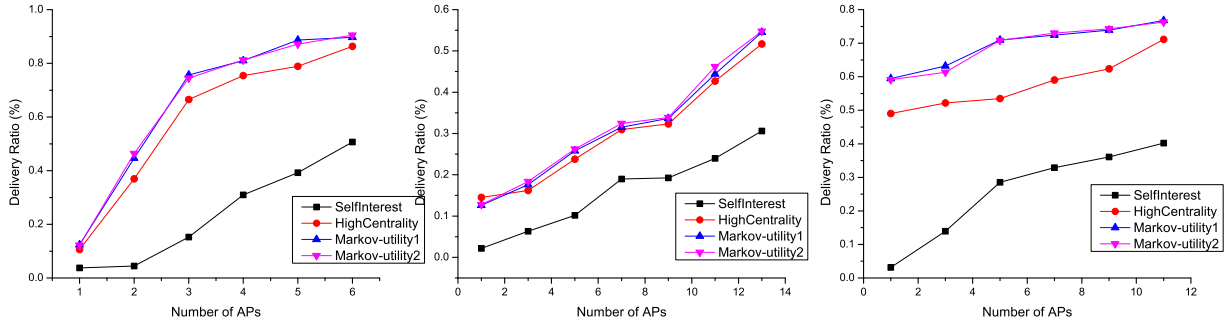
$\Phi_{m_i} = \emptyset, \forall m_i \in M$
for $m_i \in M$ **and** $\tau > s_{m_i}$ **do**
  $m_i \longrightarrow \Lambda_M$
  initialize κ_{m_i}
  for $m_k \in \Lambda_M$ **do**
    for $v_i \in V - \Lambda$ **do**
      if v_i **accesses** Λ **and** *(m_k's interest* $\in \Gamma_{v_i}$ **or** $\kappa_{m_k} > 0$*)* **then**
        copy m_k
        $m_k \longrightarrow v_{iM}$
        if m_k*'s interest* $\notin \Gamma_{v_i}$ **and** $\kappa_{m_k} > 0$ **then**
          $\kappa_{m_k} - -$
  for $v_i \in V - \Lambda$ **do**
    for $m_k \in v_{iM}$ **do**
      for $v_j \in N^\tau_{v_i}$ **do**
        if m_k*'s interest* $\in \Gamma_{v_j}$ **then**
          copy m_k
          $m_k \longrightarrow v_{iM}$
        if m_k*'s interest* $\notin \Gamma_{v_j}$ **then**
          compuate utility function $U_{v_i}(\tau)$ and $U_{v_j}(\tau)$
          if $U_{v_j}(\tau) > U_{v_i}(\tau)$ **then**
            $m_k \longrightarrow v_{jM}$
            delete m_k from v_{iM}
  for $v_i \in V - \Lambda$ **do**
    for $m_k \in v_{iM}$ **do**
      if $\tau - s_{m_i} > TTL$ **then**
        delete m_k from v_{iM}
  $\tau + +$

4.3 Performance Evaluation

In this section, we evaluate our semi-Markov data dissemination with two utility functions and compare the performance with several heuristic methods. Since we define the overhead in this paper from a new perspective and nodes relay the messages in a different way, we compare simply with the following heuristics. The first one is *SelfCast*, nodes will carry their own interested messages and disseminate the messages to the nodes who share the same interests. In *SelfCast*, all nodes are initially selfish and no one is willing to carry others' interested messages. Therefore, *SelfCast* provides a bottom line of performance for a given test-bed. The second one is *HighCentrality*. Nodes can and are willing to get server copies from APs and are responsible to improve data dissemination ratio. The criteria for a node changing the role from client to server as to a message depends on how active the node is. For a specific kind of interest, the more contacts the node has, the higher chance the node owns a server copy. Normally, *HighCentrality* gives not bad result for a given network since it takes advantage of the *high degree* nodes. Unlike existing works [80][81][82][83] [89][90], in which the overheads are not controlled before releasing the messages into the network, our goal of maximizing the data dissemination ratio subjects to the controlled *service overhead*. Therefore, to avoid unfairness, our social-aware data dissemination in this paper mainly compares with *SelfCast* and *HighCentrality* which both could be subjected to *service overhead* defined in this paper. We also implement our Markov data dissemination with two defined utility functions and use them to go against each other.

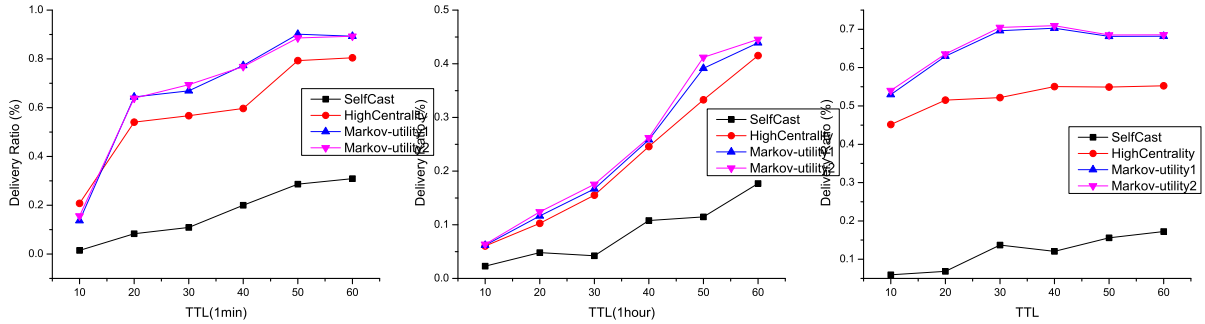
4.3.1 Simulation Setting And Data Set Prepocessing

We use three data sets including two real traces and a synthetic benchmark. Two real traces are UMassDieselNet [71] and SIGCOMM09 [40]. UMassDieselNet is a DTN testbed in which data is collected from buses running routes served by UmassTransit. In the introduction of UmassTransit, 40 buses covered more than 150 square miles and each bus is a highly mobile DTN node equipped with a small computer and communicates with each other in



(a) UmassDisel06 (b) SIGCOMM09 (c) Synthetic

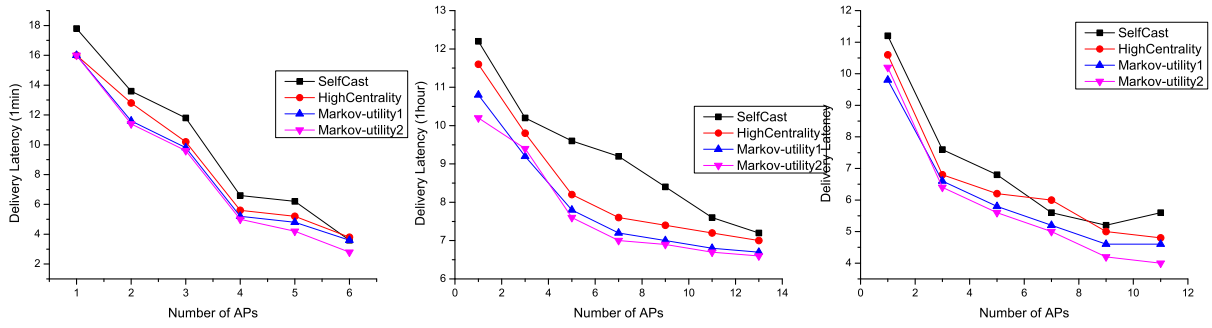
Figure 4.3. Comparison of delivery ratio on different data sets with variation of the number of APs.



(a) UmassDisel06 (b) INFOCOM06 (c) SIGCOMM09

Figure 4.4. Comparison of delivery ratio on different data sets with different TTLs.

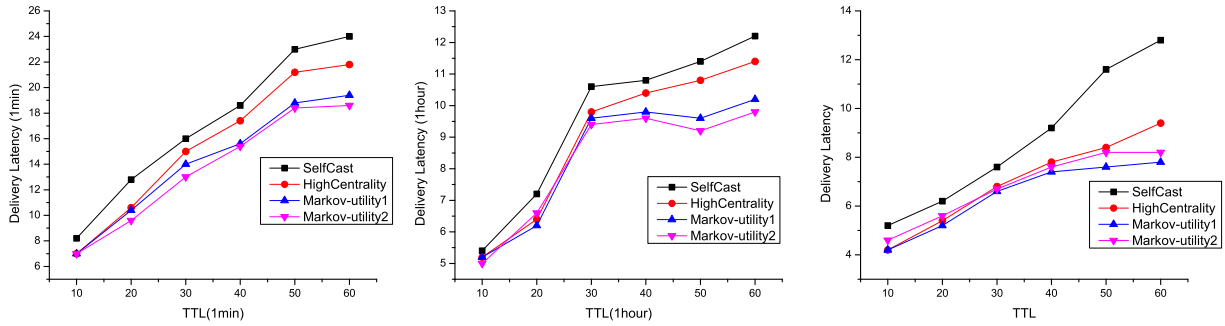
short Wi-Fi communication range. We preprocessed the UMassDieselNet dataset and found 36 nodes as a fact. We use discrete time to separate the continuous time to 398 time slots based on 1 hour interval. SIGCOMM09 consists of 76 nodes and the data is collected from volunteer attendants in SIGCOMM 2009 conference in Barcelona, Spain. Each volunteer carries a Bluetooth device to record the contacts in short range. We also preprocessed SIGCOMM09 data set in the discrete time slots and we get 350 time slots and each time slot is based on an interval of 1 hour in the real world. Since most real traces like UMassDieselNet and SIGCOMM09 are very limited on the number of nodes due to experimentation limitation in reality, but we still aim to find a larger scale of benchmark which simulates the contacts of people in mobile social networks. Therefore, the last data set we use in the implementation is a synthetic trace. We generate 200 nodes in the synthetic trace and the degree of nodes follows power-law distributions[1]. The contact frequency between two nodes are assigned a value in range $0 \sim 1$ and the frequency follows the normal distribution. In the simulation, there are 300 time slots in the synthetic trace. In all three traces of all simulation runs, we randomly assign 1000 messages to APs and each message has 3 limited server copies. In all simulations, the messages start to disseminate after a training period which takes about 20% of the total time length.



(a) UmassDisel06 (b) SIGCOMM09 (c) Synthetic

Figure 4.5. Comparison of delivery latency on different data sets with variation of the number of APs.

[1] Node degree follows power-law distribution in complex network [94]



(a) UmassDisel06 (b) SIGCOMM09 (c) Synthetic

Figure 4.6. Comparison of delivery latency on different data sets with different TTLs.

4.3.2 Comparison Result Analysis

In this subsection, we evaluate the results of the simulations on two real traces and on the synthetic trace. For each simulation, we collect the results over 30 runs. Fig.4.3 plots the data dissemination ratio of all methods on three datasets with increasing number of APs. TTL is set to 30 in both synthetic and SIGCOMM09 traces. TTL is set to 20 in UmassDisselNet. The data dissemination ratio will naturally increase when the number of APs becomes larger. From the results of all three datasets, we can see that our proposed Markov-based methods outperform *HighCentrality* and *SelfCast* methods. Our Markov-based utility functions can make nodes more effectively switch their roles between client and server for messages. In average, on dataset UMassDieselNet, both two Markov-based methods has 5% higher influence ratio than *HighCentrality* methods. On the synthetic trace, the advantages of Markov-based methods shows more obvious with about 10% higher influence ratio than *HighCentrality*. We believe Markov-based methods behaves better when the size of networks grows. From Fig.4.3, we can also find that the *Markov-utility2* has slightly higher influence ratio than *Markov-utility1* over all three datasets, which means considering multiple time slots of node's service ability can help decide the server node better to some extent. Fig.4.4 plots the data dissemination of all methods with increasing TTL. The number of APs is set to 3 for all three datasets in Fig.4.4. The results shows

similar conclusion as we have seen in Fig.4.3 that *Markov-utility2* has the best dissemination ratio and both *Markov-utility2* and *Markov-utility1* outperform *HighCentrality* on all three datasets.

Fig.4.5 plots the average data dissemination latency of all messages with increasing number of APs. TTL is set as the same as in Fig.4.3. Fig.4.6 plots the average data dissemination latency of all messages with increasing TTL. The number of APs is set 30 as well. Both Fig.4.5 and Fig.4.6 show that *SelfCast* has the longest latency. Because of lack of relays of messages for each other, *SelfCast* takes more time in average to disseminate messages to the people who are interested in. *Markov-utility2* has slightly shorter latency than *Markov-utility1* and both behave better than *HighCentrality* method.

From the comparison results, we can see that the proposed markov-based *Markov-utility1* and *Markov-utility2* can effectively increase the data dissemination ratio through taking advantage of analyzing interest transitions of nodes and deciding a better server node to disseminate messages with specific interests under limited overhead defined in this paper. In all simulations, the number of message copies is set to 3. Therefore, the limited network bandwidth will not be overwhelmed.

Notations	Definitions
V	the set of users
Λ	the set of APs
Γ	the set of interest types
ρ	the number of interest types
Γ_{v_i}	v_i's interest types
$\gamma_{v_i}^j$	whether v_i is interested in interest type j
N_{v_i}	the encounter (neighbor) set of node v_i
$N\prime_{v_i,j}$	the encounter (neighbor) set of of v_i which are interested in interest type j
$\bar{N}\prime_{v_i,j}$	the complimentary set of $N'_{v_i,j}$ regarding to N_{v_i}
m_i	the ith message
$N_{v_i}^t$	the encounter (neighbor) set of node v_i at time slot t
$N\prime_{v_i,j}^t$	the encounter (neighbor) set of of v_i which are interested in interest type j at time slot t
$\bar{N}\prime_{v_i,j}^t$	the complimentary set of $N'_{v_i,j}$ regarding to N_{v_i} at time slot t
m_i	the ith message
κ_{m_i}	the number of authorized copies of m_i
s_{m_i}	the publishing time or start time of message m_i
$\Phi_{m_i}^\tau$	the set of servers for message m_i at time slot τ
C^τ	the contact set of the users in time slot τ
c_{v_i,v_k}^τ	whether v_i and v_k has a contact at time slot τ
$U_{v_i}^j$	the service ability of v_i for interest type j

Table 4.1. Notations in the Problem Formulation and Service Utility Model

Chapter 5

INTRODUCE NEW INFORMATION DIFFUSION MODEL FOR INFLUENCE MAXIMIZATION

5.1 Introduction

With the emerging of online social networks such as Facebook, twitter, google+, and instagram, social netwofrks are playing an important role in people's interactions, idea spreading, influence propagation in human society. "word-of-mouth" [4] has an unprecedented influence through social networks. Nowadays, It becomes usual for a tweet, a photo or a video to be viewed and shared over a million times within a week in online social networks. People are enjoying commenting and sharing their friends' and followed celebrities's information to their friend circles. Sometimes a saying from nobody in network may get unexpected influence on millions of people. Moreover, With the exponentially increment of smart mobile devices with short-range communication such as Bluetooth and WiFi, mobile social networks also have an increasing influence on information diffusion and make it more convenient for people to communicate and exchange information. These social networks like Facebook has become a big platform for information dissemination and influence spread. How to take advantage of social networks to effectively spread influence becomes a challenge naturally for marketing companies who want to popularize its products.

The essence of marketing application is to spread information from a small group of people to as many people as possible in networks. Suppose such a scenario that a mobile application company develops a new game and needs to send limited free-trials to some individuals in the network, wishing people can spread the news about the game to their friends and then further propagates the news over the network to achieve a maximum influenced number of people knowing that new game. The key problem is to maximize the influence ability of the initial group of people. This problem, referred as *influence maximization* will

be addressed in this paper.

Motivated by viral marketing application, [4] and [13] first formulated the influence maximization problem as an algorithmic problem and studied the problem from perspective of probability. The problem was first addressed as an discrete optimization problem by Kempe et al. [10]. in [10], two stochastic diffusion models, namely, independent cascading (IC) and linear threshold (LT) models are proposed to describe the rules of information diffusion process. Kempe et al. formulated influence maximization problem as a network graph, under their proposed models IC and IT, through selecting a small number k vertices as an initial seed set to maximize the influence spread. Kempe et al. proved that influence maximization problem under two basic diffusion models IC and LT is both NP-hard and further proposed an approximation algorithm with approximation ratio $1 - 1/e$ to solve this problem for the first time. One of the contributions from Kempe et al. is that both IC and LT models provide model foundation for research afterwards. Diffusion models define the rules of information diffusion process and also determine whether diffusion process is as practical as real human influence propagation. In IC model, once a node becomes active, it will try to influence its adjacent nodes (also referred as neighbors later in this paper) with some probability once and only once. Each active node will influence others independently. In the LT model, with defining node *v's* active neighbor set as N'_v, then each node v becomes active when its active neighbors satisfy $\sum_{u \in N'_v} p_{uv} \geq \theta_v$, where θ_v is a threshold and $0 \leq \theta_v \leq 1$. From the definition of the models, we can see IC model assumes that each node v becomes active due to the independent effort of its active neighbors and LT model prefers more on the collective influence of all its active neighbors. Both these two models can depict real human influence propagation to some extent. Subsequently, researchers define more reasonable models to represent information diffusion process. weighted cascading (WC) model is proposed in work [2]. In WC model, given that node *v's* degree is d_v, then each active neighbor has an independent influence probability $1/d_v$ to get v influenced. Different from IC model, one active neighbor u may have more than one try to influence node v each round with probability $1/d_v$ until v got influenced. I.e., at round i, if a not yet active node v

has m active neighbors, then the probability that v get active at round i is $1-(1-1/d_v)^m$. A new round $i+1$ happens when a new active neighbor of v emerges. For the WC model, the physical meaning is that the influence of nodes should not be treated totally independently. If one node u has an early influence on v, even may fail at time t, but will always have an influence in future time slots $t+1, t+2, ...t+n$ until v get influenced at time $t+n$ or continue contributing every time when other active neighbor tries. Compared with IC model, WC model is believed to simulate the real word condition better. Consider a scenario in real world, one friend A tried to persuade you that basketball game is amazing sport but fails, on another day, one friend B did the same thing and you believed. In this case, it is more reasonable to consider A's implicit influence when B tried to influence. Therefore, WC model define more reasonable diffusion model. However, in WC model, each neighbor of v has the same influence probability which depends on the degree d_v, which contradicts the occasion in real world. Also, in all above models, they all fail to define the influence delay, i.e., the time dimension effect, especially, in WC model, which may result in different orders of $v's$ neighbors becoming active, further affect the influential ratio. Influence delay of time dimension can have an effect on the order of influence of nodes. Considering influence delay in diffusion model could be crucial when we try to propose a more real-world descriptive model. Based on the drawbacks of the most used models above, in order to define a better representation of real world human influence propagation process, we come up with a new diffusion model in this paper. Namely, sustaining cascading (SC) model, which considers the sustaining influence of each individual. The SC model is defined in section 7.2.

we use our proposed SC model to study influence maximization problem and prove that the optimization problem of maximizing the influence spread through selecting k seeds under SC model is a NP-hard problem. Besides, we also prove that the resulting influence function $\delta(\cdot)$ under SC model is submodular under some constraints and further we test the classic approximation algorithm and other heuristics under SC model. To increase time efficiency of the seed selection process, as well as achieving comparable influence ratio with approximation algorithm under the SC model, we also propose a new heuristic method which

takes advantage of properties of the SC model.

5.2 System model

In this section, we first propose the network model and our new diffusion model, namely sustaining cascading (SC) model. Then, we formulate the influence maximization problem under SC model formally. In the last subsection, we discuss the property of SC model in influence maximization problem.

5.2.1 Network Model

We model a social network graph as an undirected graph $G(V, E, D(E), P(E))$ where V is the set of nodes and the number of nodes $n = |V|$. Each node is denoted by u_i. i is the id of node and $0 \leq i < n$. Undirected edge $(u_i, u_j) \in E$ represents a social tie between node u_i and u_j. $P(E) = \{p_{ij}|(u_i, u_j) \in E, 0 \leq p_{ij} \leq 1\}$ where p_{ij} indicates the probability that node u_i activates u_j with assumption that u_i is active and vice versa. $D(E) = \{d_{ij}|(u_i, u_j) \in E, 0 \leq d_{ij} \leq \theta_d\}$ where d_{ij} indicates the time delay that when u_i and u_j tries to influence each other. $\theta_d = max(D(E))$ and is the maximum time delay in the graph G. For simplicity, we assume the edges are undirected, i.e., $\forall$ i and j, $p_{ij} = p_{ij}$ and $d_{ij} = d_{ji}$. For a node u_i, we also define its neighbors $N_{u_i} = \{u_j|(u_i, u_j) \in E\}$. For convenience, we also denote $N_{u_i} = \{\ell_{i1}, \ell_{i2}...\ell_{il}\}$ and $l = |N_{u_i}|$.

5.2.2 Diffusion Model: Sustaining Cascading Model

We define sustaining cascading (SC) model as follows. Each node has three states *neutral*, *pending* and *active*. For a node u_i, We define *neutral* as a status being inactive and has never been influenced by others. The initial status for all nodes in the graph is *neutral*. When node u_i is influenced by others successfully, u_i becomes active from *neutral*, otherwise, u_i becomes *pending* from *neutral*. If u_i is further influenced by others, u_i may become active from *pending* status and then try to influence its neighbors N_{u_i}. The transition process is depicted in Fig.7.1.

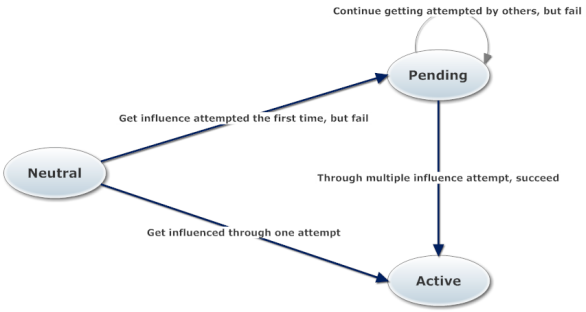


Figure 5.1. Node state transition diagram.

For each node $u_i \in V$, We also define two threshold parameter L_{u_i} and H_{u_i}, , namely lower overlapping influence trigger and higher overlapping influence trigger, respectively. Suppose u_j is active, u_i is *pending* and $u_j \in N_{u_i}$, if $p_{ij} < L_{u_i}$ or $p_{ij} < L_{u_i}$, u_j will try to influence node u_i independently, i.e., it will be the same as in IC model. If $L_{u_i} \leq p_{ij} \leq H_{u_i}$, and we assume before u_j tried to influence u_i, $n-1$ active neighbors of node u_i have tried, then whether u_i can be influenced successfully or not depends on both p_{ij} and previous comprehensive influence $P_{n-1}(u_i)$. The influence probability at each condition is defined in equation.7.1 where we choose $L_{u_i} = 0.2$ and $H_{u_i} = 0.7$. The meaning of equation.7.1 is that when a node u_j has very week or very strong influence ability to node u_i, we don't consider the previous accumulative influence from other nodes to u_i and this node will try to influence independently. Otherwise, we will consider the comprehensive sustaining influence $P_{n-1}(u_i)$ from previous nodes who tried to influence node u_i. If $p_{ij} > P_{n-1}(u_i)$, it will trigger a new round of influence with bigger probability to u_i, else the p_{ij} is not *strong* enough to trigger a new round influence and u_i will remain pending and $P_n(u_i) = P_{n-1}(u_i)$.

$$P_n(u_i) = \begin{cases} p_{ij}, \;\; p_{ij} \geq 0.7 \, or \, p_{ij} < 0.2 \\ 1 - (1 - P_{n-1}(u_i)) \cdot (1 - p_{ij}), \\ \qquad P_{n-1}(u_i) \leq p_{ij} < 0.7 \\ P_{n-1}(u_i), \;\; 0.2 \leq p_{ij} < P_{n-1}(u_i) \end{cases} \tag{5.1}$$

Where $P_n(u_i)$ is the probability that u_i becomes active because of u_j's influence attempt at *nth* round .

Every time a new active neighbor of N_{u_i} may trigger a new round that makes node u_i become active from *pending* status. Therefore, after n active neighbors become active, the accumulative probability that node u_i is active is:

$$\begin{aligned}
&\Gamma_n(u_i) = P_1(u_i) + (1 - P_1(u_i)) \cdot P_2(u_i) \\
&+ \ldots + \prod_{k=1}^{n-1} (1 - P_k(u_i)) \cdot P_n(u_i) \\
&= \Gamma_{n-1}(u_i) + \prod_{k=1}^{n-1} (1 - P_k(u_i)) \cdot P_n(u_i)
\end{aligned} \tag{5.2}$$

Once node u_i becomes active at time t, it will try to influence all its neutral and pending neighbors. The order of influence depends on the time delay d_{ij} between each node $u_j \in N_{u_i}$ and u_i.

Conceptual Justification of SC model Compared with IC, LT and WC models, we believe SC model reflects the human influence propagation mode of real world better. SC model is not a probabilistic independent model. Unlike IC model, which only allows nodes to try to activate their neighbors once and only once. For the past influences, yet not successfully, they still have future influence contribution under some constraints, i.e., SC model has memories of the influence. Every time a new active neighbor B tries to activate node A, it may trigger the comprehensive influence of all previous tries to influence A together. If we take node A as u_i, the comprehensive influence is referred as $P_n(u_i)$ in section 7.2.3. Consider a real world example. Charlie never watches basketball game. Sam tried to persuade Charlie that basketball game is interesting on Monday but failed. On Wednesday, Lucy tried Charlie again and succeeded in persuading Charlie to watch a basketball game, which made Charlie like it afterwards. Apparently, both Sam and Lucy contributes to the persuasion to some extent. In this case, how we determine whose persuasion is effective and how Sam and Lucy may influence Charlie are the problems. In SC model, it is committed that the result that Charlie became interested in basketball game was the result of both Sam's and Lucy's persuasion effort.

5.2.3 Problem Formulation

For a given social network $G(V, E, P(E), D(E))$, let $\sigma(S)$ denote the random process of the influence spread from seed set S under SC model where the seed set $S = \{s_1, s_2, ...s_k\}$. The output of $\sigma(S)$ is a set of nodes in V influenced by seed set S directly or indirectly. The objective of influence maximization problem is to select the seed set S containing k seeds to maximize the influence spread $\sigma(S)$. Different diffusion models yield to different influence propagation process and for the same seed set of S, different diffusion models can generate different result set $\sigma(S)$. The objective of this work is to study the properties of our defined SC model and propose a new heuristic algorithm to solve the influence maximization problem under the defined SC model efficiently.

5.2.4 Properties of SC model

Since SC model is not a probabilistic independent model, unlike IC and LT, the influence spread $\sigma(S)$ of seed set S does not simply equal the sum of each seed s_i's influence, i.e., $\sigma(S) \neq \bigcup_{i=0}^{k} \sigma(s_i)$. For a given seed set $S = s_1, s_2$, the relationship between $\sigma(S)$ and the individual influence $\sigma(s1)$ and $\sigma(s2)$ depends on how $\sigma(s1)$ and $\sigma(s2)$ overlaps. When $\sigma(s1)$ and $\sigma(s2)$ does not overlap at all, simply, $\sigma(S) = \sum_{i=0}^{k} \sigma(s_i)$. However, when $\sigma(s1)$ and $\sigma(s2)$ overlaps, it is not straightforward to get $\sigma(S)$. We further define the following two sets that exist under the SC model.

Definition 1. *Overlapping Loss Set (OLS)*: for two given selected seeds u_i and u_j, the influence spreads individually are $\sigma(u_i)$ and $\sigma(u_j)$, respectively, then $OLS(\{u_i, u_j\}) = \sigma(u_i) \cap \sigma(u_j)$. Similarly, for a seed set S with k seeds $\{s_1, s_2, ..., s_k\}$ where $s_i \in V$, and their individual influence spreads $\sigma(s_1), \sigma(s_2), ..., \sigma(s_k)$, respectively, $OLS(S) = \bigcup_{i=0, j=i+1}^{k} \sigma(s_i) \cap \sigma(s_j)$.

OLS measures how many nodes may get overlapped influence, which is a kind of *waste* if two seeds influence the same nodes. In our all existing models include IC, LT and WC model, OLS may happen between two selected nodes.

Definition 2. *Overlapping Gain Set (OGS)*: for two given seeds u_i and u_j, the influence spreads individually are $\sigma(u_i)$ and $\sigma(u_j)$, respectively, then for any node u_k with conditions

$u_k \in V$, $u_k \notin \sigma(u_i) \cup \sigma(u_j)$ but $u_k \in \sigma(\{u_i, u_j\})$, the node $u_k \in OGS(\{u_i, u_j\})$. Similarly, for a seed set S with k seeds $\{s_1, s_2, ..., s_l\}$ where $s_i \in V$, and their individual influence spreads $\sigma(s_1), \sigma(s_2), ..., \sigma(s_l)$, respectively, $OGS\,(S) = \{u_k | u_k \in \sigma(S) \wedge u_k \notin \sigma(T), T \subset S\}$

For node $u_k \in OGS(S)$ and $l = |S|$, it means node u_k can not be influenced independently by a single seed or a combination of m nodes from seed set S where $m < l$, i.e., it may only be influenced by selecting all l seeds. From the perspective of state transition, the nodes in $OGS(S)$ are all from state *pending* to *active*.

For a given seed set S and a graph $G(V, E, P(E), D(E))$, both $OLS(S)$ and $OGS(S)$ are abstract sets, which in the seed selection process, we use Monte-Carlo simulations to estimate $OLS(S)$ and $OGS(S)$.

5.3 Problem Hardness Analysis and Model Study

5.3.1 Problem Hardness Analysis

Theorem: The influence maximization problem under SC model is NP-hard.

Proof. Similarly with the proof that influence maximization problem is NP-hard under IC model [10], Consider a general instance of a set cover problem, which belongs to the category of NP-complete problems. Let the graph be $G(V, E)$ and a collection of subsets $S_1, S_2, ...S_m$ cover all nodes in $V = \{v_1, v_2, ...v_n\}$. Each set S_i covers zero nodes up to the complete node set V. The objective is that whether there exists a combination of selecting k subsets to cover the complete node set V. This is a set cover decision problem and we will show that the set cover problem can be considered as a special case of the influence maximization problem under SC model. Below is the process of many-to-one reduction from influence maximization problem to the set cover problem.

We first transform the influence maximization problem on SC model to the problem on IC model. Given an arbitrary instance of the influence maximization decision problem, we try to find whether there is a set S of k seeds that can successfully activate n nodes in the node set V. For each potential seed s_i, the individual spread is $\sigma(s_i)$. If an arbitrary node

$u_j \in \sigma(s_i)$, $p_{i,j} = 1$, otherwise, $p_{i,j} = 0$. If an arbitrary node is in the $OGS(s_i, s_j)$ of any two seeds s_i and s_j, then we simply remove the node from the node set V, which is a reduction process which will remove nodes in OGS. The activation process on the remaining nodes, denoted as set V', will only involve independent cascading, i.e., $\sigma(\{s_i, s_j\}) = \sigma(s_i) \cup \sigma(s_j)$. Whether we can find a set S of k seeds to activate all nodes with $\sigma(S) >\mid V' \mid +k$ in the set V' is equivalent to whether there exists k subsets that cover all nodes in the set V' , which is the set cover problem [10]. Solutions which could find set S with k seeds with influence spread $\sigma(S) >\mid V' \mid +k$, the set cover problem is also solved.

5.3.2 Submodularity of SC model

In this subsection, we prove that our diffusion SC model is submodular. Before giving the proof of submodularity under our new model, we first introduce what submodularity is and the formal definition of our model.

For an arbitrary function **F** that projects the finite set **U** to non-negative real number set $\mathbf{R}^+$, **F** is submodular if it satisfies one of the following inequalities.

$$F(S \cup \{u\}) - F(S) \geq F(T \cup \{u\}) - F(T) \tag{5.3}$$

where **u** is an arbitrary element of **U** and set $\mathbf{S} \subseteq \mathbf{T}$, or

$$F(S) + F(T) \geq F(S \cup T) + F(S \cap T) \tag{5.4}$$

for any set $S \subseteq U$ and $T \subseteq U$.

The submodular function F has the well-known property "diminishing return", which will reduce the gain gradually when continuously adding one element to a set. The "diminishing return"property is better expressed and understood in equation.5.3. If a submodular function F is monotone, since the codomain is R^+, each time adding a new element u to a set S will increase the gain of $F(S)$. Optimising the problem of selecting k elements for set S in order to maximize $F(S)$ is NP-hard [10]. However, Nemhauser, Wolsey and Fish-

er [5] shows a hill-climbing greedy algorithm by selecting the node u which maximizes the function $F(S \cup u)$ where u is later added to set S to make set S from empty set to a set with k elements. The greedy algorithm can approximate the optimum solution with a factor $1 - 1/e$.

The influence spread functions $\delta(\cdot)$ under classic IC and LT models in [10] are submodular. Therefore, by applying the hill-climbing algorithm in these models to solve the influence maximization problem can always achieve the approximation ratio of $1 - 1/e$. In fact, the hill-climbing algorithm leads all others so far in influence spread number, i.e., no other algorithm behaves better with respect to the influence spread. However, the shortage of the hill-climbing greedy algorithm is the time cost of selecting k seeds, which limits its usage in some large real networks and as the result, draws the need of other improved algorithms and heuristics with less computation time of selecting the seed set. The importance of submodularity for an influence diffusion model is that it enables the greedy algorithm which tells how well the influence spread can achieve under such a model. We will further discuss about the hill-climbing greedy algorithm and our proposed heuristic on our SC model in section 7.5.

Theorem 1. *The influence function $\delta(\cdot)$ is submodular under an arbitrary instance of SC model.*

A straightforward way to prove influence spread function $\delta(\cdot)$ is submodular is to seek function inequality $\delta(S \cup u) - \delta(S) > \delta(T \cup u) - \delta(T)$ from the definition for any set $S \subseteq T$ and an arbitrary node u. While this is not easy to quantify the spread set $\delta(S)$ for a given seed set S since the spread order under SC model is not specified exactly.

We come up with our proof from the perspective of each node's probability gain of being influenced. We try to prove for an arbitrary node u_i, the increment of probability of being influenced directly or indirectly from adding a new node u to seed set S is greater or equal than adding the same node u to seed set T where $S \subseteq T$. We first study that for an arbitrary node u_i, how the influence probability changes when its active neighbors change. Lemma 1 and its proof are given as below.

Lemma 1. *For an arbitrary node u_i with the neighbor set T where $n = |T|$ and $n \geq 2$, the increment of the accumulative probability function $\Gamma_n(u_i)$ from adding a new active neighbor u_j to u_i's active neighbor set $T1$ is greater or equal than adding the same active neighbor u_j to u_i's active neighbor set $T2$ w.r.t. $T1 \subseteq T2$ and $T2 \subseteq T$.*

Proof. We first study a case that $|T2| = |T1| + 1$, i.e., active neighbor set $T2$ has one more active neighbor than active neighbor set $T1$ and let $m = |T2|$ and $m - 1 = |T1|$. Since in equation.7.3 and equation.7.1, we don't specify the influence neighbor in each round, in order to distinguish the added node from the neighbors in set $T1$ and $T2$, we use node A as the notation for the new neighbor instead of u_j. And the individual influence probability from node A to node u_i is p_A. Since we know that when $p_A > H_{u_i}$ or $P_A < H_{u_i}$, the influence process is equal in IC model and therefore the influence function $\sigma(\cdot)$ is submodular [10]. In following, we consider only the condition when previous comprehensive influence $P^A_{n-1}(u_i)$ is involved. If A triggers the *nth* round of influence to u_i, instead of using $P_n(u_i)$, we denote *nth* round probability as $P^A_n(u_i) = 1 - (1 - P_{n-1}(u_i)) \cdot (1 - p_A)$. Suppose the neighbor sets are already sorted and new added neighbor is triggered in $T1 + 1$ round and $T2 + 1$ round, respectively. Or we can always shuffle the nodes to this scenario. we denote the increment of accumulative influence probability from adding node A to $T1$ as $G1$ and $G1 = \prod_{k=1}^{m-1} (1 - P_k(u_i)) \cdot P^A_m(u_i)$ where $P^A_m(u_i) = 1 - (1 - P_{m-1}(u_i)) \cdot (1 - p_A) = P_{m-1}(u_i) + p_A - P_{m-1}(u_i) \cdot p_A$. Similarly, the increment of accumulative influence probability from adding node A to $T2$ is $G2 = \prod_{k=1}^{m} (1 - P_k(u_i)) \cdot P^A_{m+1}(u_i)$ where $P^A_{m+1}(u_i) = 1 - (1 - P_m(u_i)) \cdot (1 - p_A) = P_m(u_i) + p_A - P_m(u_i) \cdot p_A$. It is obvious that both $G1 > 0$ and $G2 > 0$. Then we can get the following equation.

$$\begin{aligned} \frac{G2}{G1} &= \frac{(1-P_m(u_i)) \cdot P^A_{m+1}(u_i)}{P^A_m(u_i)} \\ &= \frac{(1-P_m(u_i)) \cdot (P_m(u_i) + p_A - P_m(u_i) \cdot p_A)}{P_{m-1}(u_i) + p_A - P_{m-1}(u_i) \cdot p_A} \end{aligned} \tag{5.5}$$

For simplicity, let $x = P_m(u_i)$ and $y = P_{m-1}(u_i)$ and $d = p_A$, and we assume that $\frac{G2}{G1} <= 1$. Then we derive the following equation.

$$\begin{aligned}&(1-x)(x+d-xd) > y+d-yd\\ &\Rightarrow (d-1)x^2+(1-2d)x+(d-1)y<0\end{aligned} \tag{5.6}$$

If d and y are treated as constants and if $d=1$, it satisfies. If $d<1$, when $x=\frac{2d-1}{2(d-1)}$, $(d-1)x^2+(1-2d)x+(d-1)y$ achieves the maximum value which is as below.

$$\frac{4y(d-1)^2-(2d-1)^2}{4(d-1)} \tag{5.7}$$

And since $d-1<0$, when $4y(d-1)^2-(2d-1)^2\geq 0$, it (the equation) satisfies. From the model definition, we know that $y\leq d$ and in order to satisfy $4y(d-1)^2-(2d-1)^2\geq 0$, the following inequality needs to be satisfied.

$$\begin{aligned}&4d\geq 4y\geq \tfrac{(2d-1)^2}{(d-1)^2}\\ &\Rightarrow 4d^3-12d^2+8d-1\geq 0\end{aligned} \tag{5.8}$$

Through solving the above inequality, we know that when $0.1625\leq d\leq 0.73$, it satisfies. In conclusion, if and only if $0.1625\leq d\leq 0.73$, $\frac{G2}{G1}<=1$. From the model definition, we know that $0.2\leq d\leq 0.7$. Therefore, the lemma 1 holds when $|T2|=|T1|+1$.

Now we consider a case that $|T2|=|T1|+2$, similarly, we can get fraction of increment of probability from adding new node A to $T2$ over the increment of probability from adding new node A to $T1$ as below.

$$\begin{aligned}&\tfrac{G2}{G1}=\tfrac{(1-P_{m-1}(u_i))(1-P_m(u_i))\cdot P^A_{m+1}(u_i)}{P^A_{m-1}(u_i)}\\ &\leq \tfrac{(1-P_{m-1}(u_i))\cdot P^A_m(u_i)}{P^A_{m-1}(u_i)}<=1\end{aligned} \tag{5.9}$$

Where the inequality holds when $0.1625\leq d\leq 0.73$, which is satisfied from model definition.

Similarly, we can extend to a general case that $|T2|=|T1|+t$ where $m<n$.

$$\frac{G2}{G1}=\frac{\prod_{k=m-t}^{m}\left(1-P_m\left(u_i\right)\right)\cdot P^A_{m+1}\left(u_i\right)}{P^A_{m-t+1}\left(u_i\right)}\leq 1 \tag{5.10}$$

Therefore, Lemma 1 holds. □

Let S and T denote the selected seed sets where $S \subseteq T$. To prove theorem 1, we first consider the nodes who is adjacent to set S and T, respectively. From lemma 1 we know that the increased probability of adding a new node A to set S is greater than adding the new node A to T, i.e., more nodes become active because of adding seed A to S than adding A to T. Since the Accumulative influence probability is non-decreasing, totally, the increment number of active nodes from adding A to S is greater than that from adding A to T, i.e., $\delta(S \cup A) - \delta(S) \geq \delta(T \cup A) - \delta(T)$. Therefore, theorem 1 holds.

5.4 Approximation and Heuristic Method

According to following theorem:

Theorem: [10] For a non-negative,monotone and submodular function f, let S be a set of size k obtained by selecting elements one at a time, each time choosing an element that provides the largest marginal increase in the function value. Let S^* be a set that maximizes the value of f over all k-element sets. Then $f(S) \geq (1-1/e)\dot{f}(S^*)$. In other words, S provides a $(1 - 1/e)$ approximation.

The influence spread funtion $\delta(\cdot)$ satisfies non-negative, monotone and submodular and therefore a hill-climbing algorithm can provide the approximation guarantee close to $1-1/e$.

The following greedy method can provide the approximation guarantee $1 - 1/e$.

Algorithm 3: Approximation method

```
1 S = ∅
2 for i = 1 to i = k do
3 |   Select u = arg max_{w∈V\S}(δ(S ∪ w) − δ(S))
4 |   S = S ∪ u
5 output S
```

The greedy method belongs to the hill-climbing algorithm. In each iteration, when we select a new node u to S and $S = S \cup u$, limited by the experimental environment, we

run $R = 100$ times to approximate monte carlo simulation to approach the real results. it can guarantee the approximation ratio $1 - 1/e$. However, the time complexity of the greedy method on our SC model is also high. If we use m rounds to approximate the cost of selecting each node, then the time complexity is $O(knmR)$. When the network size is relatively large, the time cost may surpass the computation ability of modern computers. Therefore, we propose another heuristic to speed up the seed set selection process while achieving comparable influence ratio compared with the greedy method.

Algorithm 4: HEURSISTIC:BESTOVERLAPPINGCOVERAGE

$S = \emptyset$, $R = 100$, $\delta_1 = 0.4$ 0.8, $delta_2 = 0.05$ 0.1, $delta_3 = 0.02$ 0.1
$PreSelection(M)$
for $i = 1$ *to* $i = M$ **do**
    for $r = 1$ *to* $r = R$ **do**
        calculate $Cover(v_i) += \delta(v_i)$
    for $j = 1$ *to* $j = N$ **do**
        if v_j *is covered by* $Cover(v_i)$ *more than* $R \cdot \delta_1$ **then**
            $MeanCover(v_i) = MeanCover(v_i) \cup v_j$
        if v_j *is covered by* $Cover(v_i)$ *more than* $R \cdot \delta_2$ *but less than* $R \cdot \delta_1$ **then**
            $MaxCover(v_i) = MaxCover(v_i) \cup v_j$
for $i = 1$ *to* $i = k$ **do**
    $OLS(S \cup u) = MeanCover(S) \cap MeanCover(u))$
    $OGS(S \cup u) = \delta_3 * MaxCover(S \cup u)$
    select $u = \arg\max_{u \in V \setminus S}(MeanCover(u) + OGS(S \cup u) - OLS(S \cup u)$
    $S = S \cup u$
output S

Our new heuristics is described as below in Algorithm.4.

Since in each round of selecting a seed, the coverage of the seeds are partially repeatedly calculated. Also, with the consideration of SC model property, we can improve the efficiency of calculation. According to the definition of OLS and OGS in subsection 5.2.4, it may be a good idea to calculate the $OLS(S \cup u)$ and $OGS(S \cup u)$ for each potential seed $u \in V/S$. And the best seed for the current round will be the one that maximizes $\sigma(u) + OGS(S \cup u) - OLS(S \cup u)$. However, in such a process, for each round of adding new seed node to the seed

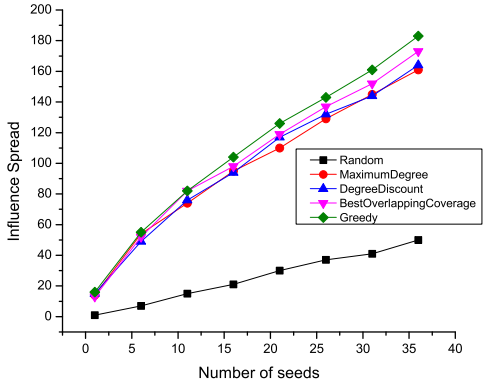
set, we may still need to calculate each node $u \in V \backslash S$. Therefore, we try to use another way to estimate the OLS and the OGS. As shown in Algorithm.4, we first do a a pre-selection process $PreSelection(M)$ to select the top M nodes according to the node degree since we observe that most seeds selected in the hill-climbing greedy method have relatively big degrees. Then we calculate for each node in the pre-selected set, how many nodes and for each node v_i, the number of times it has been covered, which is represented by $Cover(v_i)$. For a node v_j, if $v_j \in MeanCover(v_i)$, it means a strong likelihood of influencing node v_j through selecting v_i as the seed. While if $v_j \in MaxCover(v_i)$, it means selecting v_i as the seed may be able to influence v_j, but the chance is very limited. Then if a node $v_k \in MeanCover(v_i)$ and $v_k \in MeanCover(v_j)$, then we simply count that v_k is in $OLS(v_i, v_j)$. Similarly, if a node $v_k \in MaxCover(v_i)$ and and $v_k \in MaxCover(v_j)$, we simply count that v_k is in $OGS(v_i, v_j)$ and a factor $delta_3$ is multiplied to estimate the strengthened likelihood of comprehensive influence from both v_i and v_j. With the estimated OLS and OGS, we can now decide the seed u for each round to maximize the influence in line 14 in Algorithm.4. The parameters used in this algorithm may vary according to different data sets.

5.5 Experiments

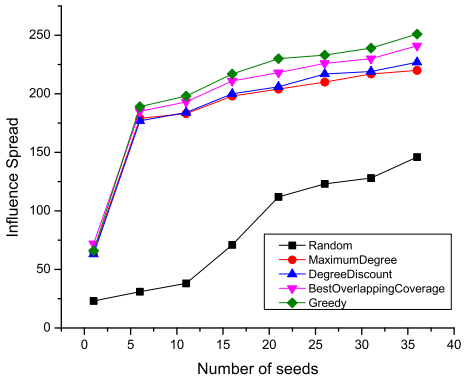
We apply the new proposed SC model to several real network data sets. Also we test out various seed selection algorithms including our proposed heuristic algorithm under the SC model on these real network data sets. In this section, we first introduce the experimental setup, then we discuss the experimental results of all algorithms through the influence spread of SC model on different networks.

5.5.1 Experimental setup

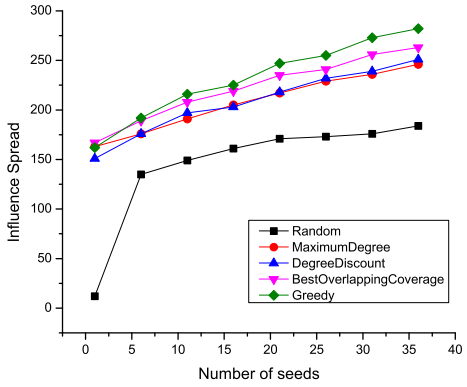
Data set introduction We use four real network data sets as test-beds. We introduce them in the order of increasing number of nodes in the networks. The first network we use is called *NetHEPT* which contains $15,233$ nodes and $58,891$ edges. *NetHEPT* is a collaboration network which represents the collaboration relationship among authors writing papers, which



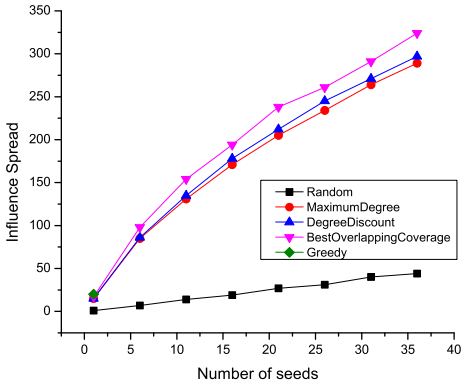
(a) NetHEPT



(b) Ego-Facebook



(c) Wiki-vote



(d) Amazon0302

Figure 5.2. Comparison of influence spread of different algorithms on different data sets with increasing number of seeds.

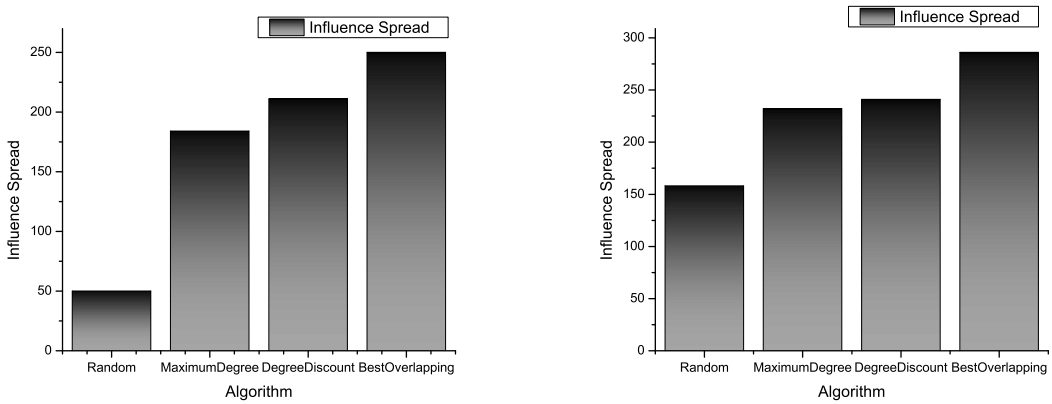
is an undirect network. *NetHEPT* is also used by [2] and [10] and it was downloaded from http://research.microsoft.com/enus/people/weic/graphdata.zip. The second network data is called *ego-Facebook* data set [12], which consists of $4,039$ nodes and $88,234$ edges and it is an undirect network. The Facebook data was collected from survey participants using an online application which could provide users' basic information. The third network data is *Wiki-vote* [6], which consists of $7,115$ nodes and $103,689$ edges. According to [6], the network contains all the Wikipedia voting data from the inception of Wikipedia till January 2008. The edge (i, j) represent user i votes user j. The network is a directed network but is addressed as an undirect network in this experiment. Every edge in Wiki-vote will represent nodes voting each other since currently we only consider undirect cases under the SC model. The forth network data is Amazon data set [7], which was collected by crawling Amazon website on March 02, 2003. The Amazon data set consists of $262,111$ nodes and $1234,877$ edges, which is the largest data set we use in this experiment.

Propagation Probability Model **Trivalency model**. Trivalency model was first used in [3], which randomly select a probability for each edge from an array containing three probabilities. We use the probability array $\{0.01, 0.02, 0.05\}$ in this model.

Server specification The experiments run on a cluster server, the node we use is equipped with Quad-Core AMD Opteron(tm) Processor 2376 and can access up to 264G system memories. Since each node has eight processors including both physical and logical ones, to increase output efficiency, we run eight threads simultaneously, each thread is responsible for selecting k seeds for a specific algorithm and $1 \leq k \leq 8$.

Algorithms introduction We run the following algorithms under the SC model on all four network data sets.

- **Random**: All the seeds are selected randomly from the node set. This algorithm takes constant time to calculate the seed set.



(a) NetHEPTBig (b) WikiBig

Figure 5.3. Comparison of running time of different algorithms when number of seeds = 100

- **MaximumDegree**: This heuristic simply selects k seeds with the largest degrees, which is first used to go against other algorithms in [10].

- **DegreeDiscount**: This is an algorithm based on **Maximum Degree**. Instead of selecting k seeds with largest degrees directly from the node set. This Algorithm adds a degree discount to the nodes whose neighbors are already fully or partially influenced. For a specific node, the degree discount depends on how many neighbors of this node have been influenced by previous picked seeds. The algorithm was first proposed in [2]. The original algorithm in [2] was designed under IC model. In this paper, we calculate the degree discount similarly with the original version in [2], but just under the SC model.

- **Greedy**: This is the approximation algorithm with approximation ratio $1 - 1/e$ we introduced in Algorithm.6. The greedy algorithm was first introduced and proved with the approximation ratio in [10] and since it has the best influence spread, it is used to mark the best influence spread a network graph can achieve with k seeds in nearly all works targeted on influence maximization problem.

- **BestOverlappingCoverage**: This is Algorithm.4 we introduced above. It is named

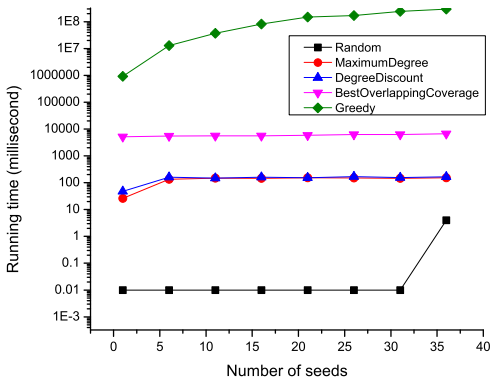
BestOverlappingCoverage since it calculates the overlapping gains and overlapping loss between each two potential nodes.

Certainly there are still other heuristics in the existing works under other models. We don't modify them to adjust under the SC model since most of the algorithms are designed on a specific model and may be too complicate under the proposed SC model. For all the above algorithms, once a seed set is chosen, the influence spread running all real data sets are ran 1000 times and output the average spread value. For the *Greedy* algorithm, we use $R = 100$ to select each seed in the seed set. In the *BestOverlappingCoverage* algorithm, we use $R = 100$ to calculate the overlapping gain and overlapping loss. Besides, we use $M = 300$ to pre-select the potential seed list.

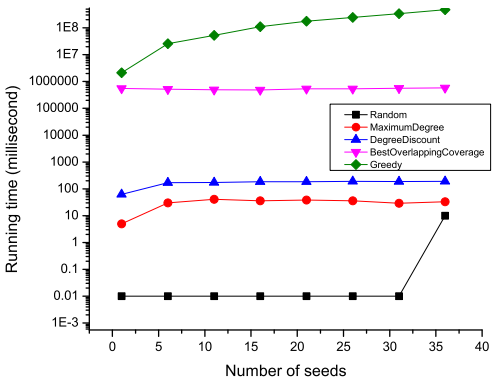
5.5.2 Experimental results

We discuss the experimental results of different algorithms on different data sets in this subsection. We evaluate the results based on the influence spread and the running time.

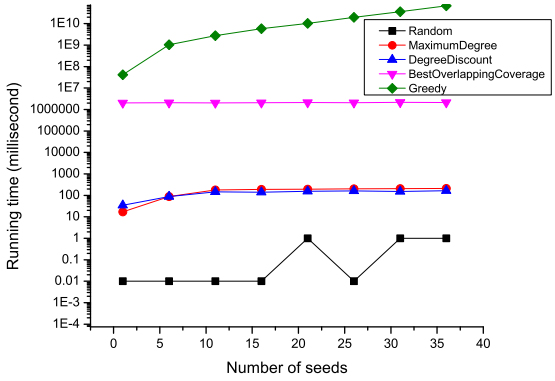
We first take a loot at the influence spread of different algorithms on all four real data sets in Fig.5.2. *Random* performs badly as the baseline for the spread number on a specific data set. Different from *Random*, *Greedy* should behave best from the perspective of the spread number. For *MaximumDegree* and *DegreeDiscount*, as we can see, in average, we get similar results compared with the one in [2] that *DegreeDiscount* is slightly better than *MaximumDegree*. Also, we can see that the proposed *BestOverlappingCoverage* performs much better than both *DegreeDiscount* and *MaximumDegree*. Though *BestOverlappingCoverage* can't surpass *Greedy* algorithm in spread number, *BestOverlappingCoverage* still achieves a comparable spread number in all data sets except for *Amazon0302* cause we did not get the result *Amazon0302* due to its big size. Also, we can see a trend that the more seeds we choose, the bigger difference there will be among these five algorithms, i.e., *BestOverlappingCoverage* performs better with a bigger seed set. Fig.5.3 shows the influence spread on *NetHEPT* and *Wiki-vote* where the number of seeds = 100. We can see that *BestOverlappingCoverage* is more than 10% better than *DegreeDiscount* in this two cases. We couldn't



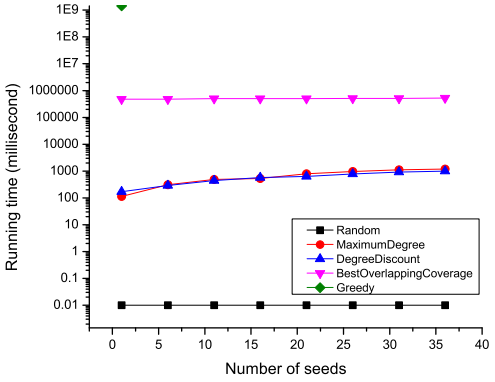
(a) NetHEPT



(b) Ego-Facebook



(c) Wiki-vote



(d) Amazon0302

Figure 5.4. Comparison of running time of different algorithms on different data sets with increasing number of seeds.

get the spread of *Greedy* algorithm due to time cost.

Fig.5.4 plots the running time of different algorithms on these four data sets. We only calculate the time cost of calculating the seed set, which does not include the influence spread time. The y-axis of all four plots in Fig.5.4 are in Log10 scale since there is a big running time difference between *Greedy* or *BestOverlappingCoverage* and other algorithms. For each algorithm in each data set, the trend of running time is increasing according to the increasing number of seeds in all algorithm, which does not show clearly in Log10 scale. We can see that *Greedy* consumes significant time while all others can finish in acceptable time. As we can see in Fig.5.4, the running time of *Random* algorithm does not depend on the network while other algorithms do. We use nanosecond in the program to catch the running time cost. However it is still hard to guarantee the exact time cost. Usually, for *Random* algorithm, when the number of seeds < 31, 0 millisecond was recorded. Since we plot y-axis in Log10 scale, we put 0.01 millisecond instead of 0 for *Random* algorithm for some number of seeds. For *MaximumDegree* and *DegreeDiscount* algorithms, in all four data sets, the running time cost are around 100 milliseconds. *DegreeDiscount* slightly cost more than *MaximumDegree*. For the proposed *BestOverlappingCoverage*, it can finish within a minute with number of seeds < 36 on *NetHEPT*. It can finish within half an hour on *ego-Facebook* as well as on *Amazon0302* and within an hour on *Wiki-vote* separately. However, *Greedy* algorithm usually cost hours to days to finish the calculation of choosing the seed set.

Through the discussion on Fig.5.2 and Fig.5.4, We can see that *BestOverlappingCoverage* algorithm has comparable influence spread with reasonable running time cost under the SC model and through testing different algorithms on the SC model, we can see that the SC model can be used for information diffusion effectively.

Chapter 6

STRENGTHEN NODAL COOPERATION FOR DATA DISSEMINATION IN MOBILE SOCIAL NETWORKS

6.1 Introduction

Most existing works which aim at improving dissemination ratio and reducing delay in MSNs assume that all the nodes are completely cooperative. However, the mobile users in reality can be either cooperative or selfish. More precisely, some mobile users may be cooperative if they have extra resources such as abundant AP access time, enough bandwidth and storage buffers. In the meanwhile, most of them are selfish naturally and resources are always limited at most time. Moreover, if resources are limited, no matter whether mobile users are cooperative or not, they also need to be smart to choose the messages considering both the prospective of the whole network performance and each individual's own benefits. Therefore, a practical incentive scheme is essential to encourage nodes to be wisely cooperative.

The incentive schemes in wireless networks fall into three categories: reputation, barter (Tit-for-Tat), and credit (virtual money) based schemes. We will discuss the three categories in Section II. In this work, we take credit as the stimulus to encourage nodes to be more cooperative. We assume all the nodes are selfish initially but rational and can be motivated by benefits. We propose a credit-based incentive scheme to strengthen nodal cooperations. The messages fall into a set of interests and they are provided by content provider and can be reached through APs. Each mobile user has one or multiple interests. Initially, each user is assigned some credits and users only carry and share their own interested messages. With the statistical analysis of growing number of contacts with each other and the user interest information, each node can evaluate their neighbors' (the nodes they encounter) ability to fetch messages of a specific kind of interest. Knowing the number of available

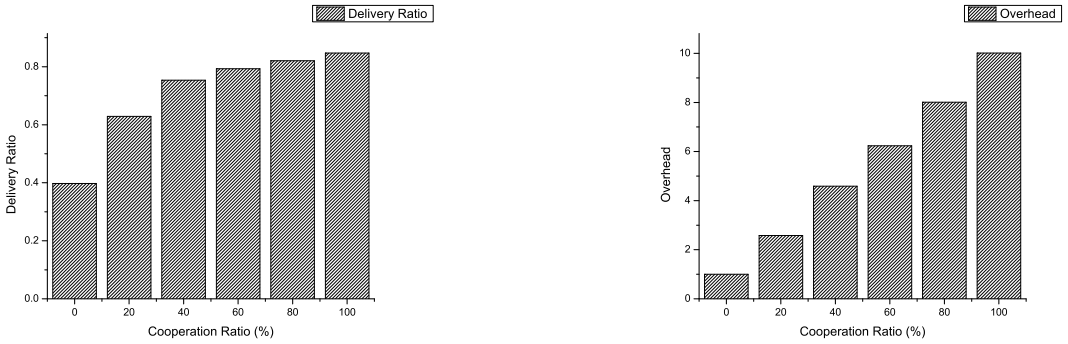
credits, the neighbors' contact patterns and all the fetching abilities of messages of all kinds of interests this node has, the node then decides which neighbors to rent to help get messages of which type of interest in order to get his own optimization goal of fetching more interested messages while paying less. The nodes being rent are paid with credits. Such a system is like a business market with fluent currency (credits). To simplify the problem, we do not consider the limited resources such as bandwidth and contact duration. Moreover, the credits in the networks are cryptic messages issued from authenticated APs. For other security issues, we leave them in future work.

We summarize our contributions as followings. To the best of our knowledge, our work is the first credit-based scheme in stimulating nodes to be cooperative in data dissemination problem in mobile social networks. Our credited-based scheme can effectively strengthen nodal cooperation in order to increase packet delivery ratio and decrease the delivery latency. Our scheme is based on fairness and all nodes are stimulated according to the maximization of their own benefits, which at the same time, increase the cooperative level of the whole network.

The followings are the challenges. First of all, how to analyze the mobility patterns to evaluate a node's fetching ability of messages of a specific kind of interest quickly and dynamically considering the fact that mobile devices are computation-limited. Second, how to define a reasonable and fair optimization function for all the nodes in order to stimulate them. Third, how to reward the credits and control the currency flow. The last but the most important, how to test and maintain the credit-based network with a healthy health state, that is, how to avoid the credits to be accumulated in a small portion of nodes such that the fluency of the credits do not become weak and performance of system is consistently good.

6.2 PRELIMINARY ANALYSIS

Before we propose our system model and incentive scheme, we first investigate the following two questions through experimental analysis as follows: "How important the cooperative nodes are in the network?" and "Why totally cooperative in network is not practical?". A



(a) Dissemination Ratio

(b) Overhead

Figure 6.1. The influence of different cooperation ratios to dissemination ratio and overhead on UMassDieselNet [71].

set of cooperative nodes from all nodes, which carry all kinds of messages, are selected to test the data dissemination performance. We take the real trace UMassDieselNet [71] as the testbed. UMassDieselNet is a bus-based delay tolerant network testbed consisting of 36 buses. In the experiment, hybrid architecture is applied. There are 3 access points that are randomly selected and 3000 messages are disseminated over the network. Each time slot in simulation corresponds to 60 minutes in the real trace. The experimental results are shown in Fig.2. Cooperation ratio denotes the ratio of cooperative nodes selected from all network nodes. Fig.2(a) shows that a higher cooperation ratio results in a higher dissemination ratio. Actually the delivery ratio of complete cooperation is double that of no cooperation, which indicates cooperation is crucial to improve the delivery ratio. Next, we will investigate the relationship between the cooperation ratio and overhead, which is shown in Fig.2 (b). With the increment of cooperation ratio, the increment of dissemination ratio grows slower, but the increment of overhead is growing linearly, i.e., high degree of cooperation may incur more overhead but only achieve limited data dissemination ratio improvement. Furthermore, mobile users are more likely selfish than cooperative [78]. The aforementioned reasons demonstrate the complete cooperation is not practical. Similar results can also be obtained in another two DTN real traces INFOCOM06 [69] and SIGCOMM09 [70]. In the following sections, we try to propose a new practical credit-based incentive scheme stimulate

nodes to become more cooperative in order to increase data dissemination ratio and also keep a relatively low overhead.

6.3 SYSTEM MODEL

We consider an MSN with hybrid architecture as shown in Fig.1. All the users use short-range radio to communicate. Some nodes have chances to access APs. Since MSNs can be considered as a special kind of delay tolerant network, given any snapshot of the network, only some nodes are connected and the whole network is disconnected. The links among all the nodes are highly dynamic and the whole network graph is sparse.

In general, the hybrid architecture of an MSN consists of APs and mobile users. APs are normally static places such as library, cafeteria and office buildings that can provide Internet services. If mobile nodes can also provide connectivity and downloading service between mobile users and content providers, they can also be treated as APs. Mobile users can be classified into two kinds depending on whether they can access the APs. Users without access to APs can only get interested messages through message propagation by other nodes in the network.

More specifically, an MSN consists of n mobile nodes and m APs. We assume all kinds of messages are equally important in value and each message contains only one interest. T is the set of interests with $t = |T|$.Each mobile node has one or multiple interests and maintains an interest array to mark whether an interest type is cared about by this node. For node a, the interest array is denoted as $\langle \delta_1^a, \delta_2^a, ..., \delta_t^a \rangle$. $\delta_i^a = 1$ if interest type i is cared about by node a, otherwise, $\delta_i^a = 0$. We define the dissemination overhead as the number of all messages relayed and accepted in the network over the number of messages accepted by the nodes with corresponding interests.

6.4 CREDIT-BASED INCENTIVE SCHEME

In this section, we propose a new credit-based incentive scheme to stimulate nodal cooperation for data dissemination in MSNs. We aim at improving the performance of data

dissemination ratio and delay while keeping the overhead in a relatively low level. We first evaluate the fetching abilities of messages of all kinds of interests for each single mobile node, then we formulate an optimization function for mobile nodes to help deciding how to rent other nodes to achieve the locally optimized goal of getting more interested messages while paying less.

6.4.1 Definitions

In order to evaluate how useful a node is in helping other nodes that it may meet, we start with the following definitions.

Definition 1. *Neighbors*: for any given node a, the nodes which have encountered node a are defined as *neighbors* of node a, denoted as N_a. Given an interest i, if the nodes in N_a have interest i, they are denoted as $N'_{a,i}$, otherwise, denoted as $\bar{N}'_{a,i}$.

Definition 2. *Interest Fetching Ability (IFA)*: for any given node a, a's neighbors N_a and a given interest i, *IFA* of node a represents the ability that node a get messages of interest i from its neighbors N_a directly, denoted as $P^a(i)$. In the following, if the node is not specified, we may simply use $P(i)$ instead.

Considering the high-dynamic connectivity among the nodes and the variation of the neighbors as well as the computation complexity, and inspired by [78], we also apply the statistical quality control tools to compute and update *IFA*. Exponentially Weighted Moving Average (EWMA) chart is one of the quality control charts that is widely applied since it is simple and easily computed. Moreover, the EWMA chart is relatively robust on the base of Poisson distribution. Meanwhile, most contact processes in DTNs are Poisson processes [68] and MSNs are a special kind of DTNs. We evaluate IFA by dividing it to the following two parts, both of which are calculated using EWMA.

IFA-AP Since we consider the hybrid architecture of MSNs, mobile nodes have two ways to get the interested messages. If nodes get the interested messages from APs directly, then we denote the interest fetching ability from APs as IFA-AP. We define each time slot

T as the updating interval. The computing and updating functions are as below.

$$Y_n(i) = \begin{cases} (1-\lambda_1) Y_{n-1}(i) + \lambda_1 & Contact \\ (1-\lambda_1) Y_{n-1}(i) & Timeout \end{cases}, n = 0, 1, 2, ... \tag{6.1}$$

$Y_n(i)$ represents the IFA-AP at time slot n for interest i and is initialized to zero. Before the time slot expires, if there are contacts between the current node and any AP, the upper one of Formula 6.1 is carried out. Otherwise, the lower function holds. λ_1 is a constant and must satisfy $0 \leq \lambda_1 \leq 1$.

IFA-Prop When mobile nodes meet each other, they can also get interested messages from each other. We define the interest fetching ability from propagation among the mobile nodes as IFA-Prop. Similar with IFA-AP, updating happens in each interval T.

$$Z_n(i) = \begin{cases} (1-\lambda_2) Z_{n-1}(i) + \lambda_2 & Contact \\ (1-\lambda_2) Z_{n-1}(i) & Timeout \end{cases}, n = 0, 1, 2, ... \tag{6.2}$$

Similar with IFA-AP, $Z_n(i)$ represents the IFA-Prop at time slot n for interest i. Before the time slot expires, if there are contacts between the current node and other mobile nodes with interest i, the upper one of Formula 6.2 is carried out. Otherwise, the lower function holds. λ_2 is a constant and must satisfy $0 \leq \lambda_2 \leq 1$.

IFA IFA-AP and IFA-Prop measure the interest fetching ability from the APs and propagation, respectively. Some nodes in the hybrid architecture may have both abilities and the question is how to evaluate which ability is more important. By applying another EWMA chart, we try to find a balance point between IFA-AP and IFA-Prop. We believe with the updating of the EWMA chart, the balance point first varies then approaches to a relatively steady point.

$$P_n(i) = (1 - \gamma_n(i)) \cdot Y_n(i) + \gamma_n(i) \cdot Z_n(i), n = 0, 1, ... \tag{6.3}$$

$P_n(i)$ represents the interest fetching ability at time slot n for interest i. $Y_n(i)$ and $Z_n(i)$ are IFA-AP and IFA-Prop, respectively. $\gamma_n(i)$ satisfies $0 \leq \gamma_n(i) \leq 1$ and is defined as

$$\gamma_n(i) = \begin{cases} (1-\tau)\,\gamma_{n-1}(i) + \tau & Prop \\ (1-\tau)\,\gamma_{n-1}(i) & AP \end{cases}, n = 0, 1, ... \tag{6.4}$$

$\gamma_n(i)$ is a feedback value of the practical data dissemination and is updated in each time slot. $\gamma_n(i)$ updates when the node fetches interested messages either from APs or from propagation. Initially, $\gamma_n(i)$ is set to 0.5, representing that IFA-AP and IFA-Prop are equally important to the comprehensive IFA.

Definition 3. *Interest Absorbing Ability (IAA)*: for any given node a, N_a and interest i, $\bar{N}'_a$ is a set without interest i. If node j belongs to N_a, then IAA of node a for interest i from node j represents the ability that node a gets messages of interest i from node j directly. To calculate IAA, we need the contact frequency of each pair of nodes. We define f_j^a as the contact frequency between a and j. $P^j(i)$ is the IFA that neighbor j has as to interest i. We can define IAA as below.

$$u_{i,j}^a = f_j^a \cdot P^j(i) \tag{6.5}$$

All the mobile users are supposed to be selfish but rational. They are willing to carry and share their interested messages with each other. Therefore, for a given interest i, if neighbor j of node a belongs to $N'_{a,i}$, we denote IAA of node a for interest i from j as $u_{i,j}^a$. If neighbor j of node a belongs to $\bar{N}'_{a,i}$, we denote IAA of node a for interest i from j as $u'^a_{i,j}$. In the following, if the node is not specified, we simply use $u_{i,j}$ and $u'_{i,j}$ instead.

6.4.2 Rental decision

When the system runs a while, each node has a list of neighbors and has calculated IAA from each neighbor for each interest i that this node has. Then we can define a utility function for each mobile node. The utility function helps a node to decide which nodes to

"rent" and what kinds of interested messages to fetch. The utility function is defined as

$$\begin{aligned} U_a = & \sum_{i=0}^{t} \left(1 - \prod_{j \in N'_{a,i}} \left(1 - u^a_{i,j}\right) \prod_{j \in \bar{N}'_{a,i}} \left(1 - u'^a_{i,j} \delta^a_{i,j}\right) \right) \delta^a_i \\ & - \sum_{i=0}^{t} \left(1 - \prod_{j \in N'_{a,i}} \left(1 - u^a_{i,j}\right) \right) \delta^a_i \end{aligned} \tag{6.6}$$

subject to

$$f_a(x) \leq C_a \tag{6.7}$$

$$x = \sum_{i=0}^{t} \sum_{j \in \bar{N}'_{a,i}} \delta^a_{i,j} \tag{6.8}$$

$$\delta^a_i \in \{0, 1\} \tag{6.9}$$

$$\delta^a_{i,j} \in \{0, 1\} \tag{6.10}$$

U_a represents the difference between renting other nodes to get node a's interested messages and it totally depends on the neighbors' common interests. There are totally t kinds of interest types in the network. For each interest i, if node a has the interest, $\delta^a_i = 1$, otherwise, $\delta^a_i = 0$. For each neighbor $j \in N'_{a,i}$, if $\delta^a_i = 1$ and j is rent by node a to help with obtaining the messages of interest i, $\delta^a_{i,j} = 1$, otherwise, $\delta^a_{i,j} = 0$. In Formula 6.7, $f(x)$ is the prepay function which is used to decide how much to pay to rent a node and the concrete functions are discussed in Section 6.6. C_a is the number of credits that node a has when the decision is made.

From the perspective of increasing data dissemination ratio and reducing data dissemination delay, we can simply set our optimization goal as maximizing U_a by finding the optimal rental set δ^*. However, in this work, our goal is not purely targeted at improving data dissemination ratio and reducing delay. Instead, our goal is to provide a practical incentive scheme to stimulate nodal cooperation in order to improve the performance of the two metrics while still have a good control of the overhead. Therefore, we define our optimization

goal as below.

$$\Delta\left(\frac{U_a}{f_a}\right) \geq \theta \tag{6.11}$$

where θ is a threshold which approaches to 0. $\Delta\left(\frac{U_a}{f_a}\right)$ represents the increment of U_a per credit spent.

There are two specific reasons why we use Formula 6.11 as our optimization goal. The first reason is that from the observation of the IAA distribution of most nodes, we find that for each interest i, the IAA of node a from his neighbors $\bar{N}'_{a,i}$ may vary a lot. Fore each interest i, since we select $\delta^a_{i,j}$ in the decreasing order, the curve of U_a increases more like logarithmical growth. For the rational node a, when spending more credits on getting nearly no more utility gain, it is meaningless to spend more credits in this decision period. Forcing nodes to spend the credits on the little gain can increase the data dissemination ratio to some extent but may harm the fairness and disobey the assumption of selfish nature of the nodes. The second reason is that from the perspective of overhead control, spending more credits to get small utility gains may help improve the delivery ratio and delay, but on the other hand, it more likely leads to a much higher overhead.

6.4.3 Process of the credit-based incentive scheme

With all the above definitions, in order to make the incentive process more clear, we use the following steps to illustrate how the incentive scheme works.

Initialization All the mobile nodes are initialized with the number of credits C_a. Each node has one or multiple interests from the t kinds of interests in the network.

User-centric information updating Node a moves in the MSN and when node a meets node j, if j is a new neighbor, it is added to a's neighbor list. Otherwise, a updates the contact frequency with j and fetches user-centric information from j including j's IFA. For each interest i that a has, update the IAA of interest i from node j.

Rental process We define a rent cycle period as Δ. After each system period Δ, node a makes decisions to get the rental set of neighbors according to Formula 6.6. However, even though for each interest i that a has, the rental set is selected, the renting process is not finished and the rent nodes are not working until the next contacts with the neighbors in the rental set occur. E.g., node j is chosen by node a to help fetch messages of interest i, and c_j credits are needed to be paid to node j. Before a meets j next time, c_j credits are put into *credits_processing* which is a discrete account used to store the credits that are decided to be spent but not paid yet. If a meets j before Δ expires, j is rent by a and get c_j credits paid from a's *credits_processing*. Rent by node a for interest i, node j will add interest i to his temporary interest array. If a fails to meet j before Δ expires, c_j credits in *credits_processing* cannot be paid and will be put back into a's total credits C_a.

Message delivery If node j is rent by a for interest i successfully, node j adds interest i to j's temporary interest array and then starts to carry and share messages of interest i with all its neighbors. The rent will last a period Δ and then the interest i goes off j's temporary interest array and at the same time, all the messages of interest i are dropped. From the perspective of a, if j is successfully rent by a but still fails to get new messages of interest i for a or j fails to meet a with messages of interest i before Δ expires. Then a will decrease the IAA from node j in interest i also by using a EWMA chart. From the perspective of the whole network, the more rent nodes, the better data dissemination ratio and delay. Meantime, the overhead also increases.

6.5 APPROXIMATION ALGORITHM

In this section, we present our algorithm to find the feasible solution of the optimization problem for each single mobile user. We first analyze U_a by dividing it into two parts: the new gain by renting other nodes $\sum_{i=0}^{t}\left(1-\prod_{j\in N'_{a,i}}\left(1-u^a_{i,j}\right)\prod_{j\in \bar{N}'_{a,i}}\left(1-u'^a_{i,j}\delta^a_{i,j}\right)\right)\delta^a_i$ and the original gain by sharing messages with neighbors with common interests $\sum_{i=0}^{t}\left(1-\prod_{j\in N'_{a,i}}\left(1-u^a_{i,j}\right)\right)\delta^a_i$.

From the observation of these two parts, we can simply find that the original gain by sharing messages with neighbors with common interests can be treated as a constant. For the first part, we can solve it by the following greedy algorithm. In each iteration, the algorithm finds $(i,q) = \underset{i,q \in \bar{N}'_{a,i} - L_i}{\arg\max}\ U_a^{i,q}$.

$$\begin{aligned} U_a^{i,q} = & \left(1 - \left(1 - {u'}^a_{i,q}\delta^a_{i,q}\right) \prod_{j \in {N'}_{a,i}} \left(1 - u^a_{i,j}\right) \prod_{j \in \bar{N}'_{a,i} \cap L_i} \left(1 - {u'}^a_{i,j}\delta^a_{i,j}\right)\right)\delta^a_i \\ & - \left(1 - \prod_{j \in {N'}_{a,i}} \left(1 - u^a_{i,j}\right)\right)\delta^a_i, \end{aligned} \tag{6.12}$$

where $j \in L_i$ if $\delta^a_{i,j}$ is assigned to 1 by previous iterations. $\delta^a_{i,q}$ is set to 1 and $L_i = L_i \cup \{q\}$ at next round if

$$\Delta\left(\frac{U_a^{i,q}}{f_a^{i,q}}\right) \geq \theta \tag{6.13}$$

where $f_a^{i,q}$ is the spent credit at the current iteration when setting $\delta^a_{i,q} = 1$. The algorithm then keeps searching the next feasible neighbor and interest. The algorithm stops until no more feasible neighbors and interests can be detected or the node is running out of credit.

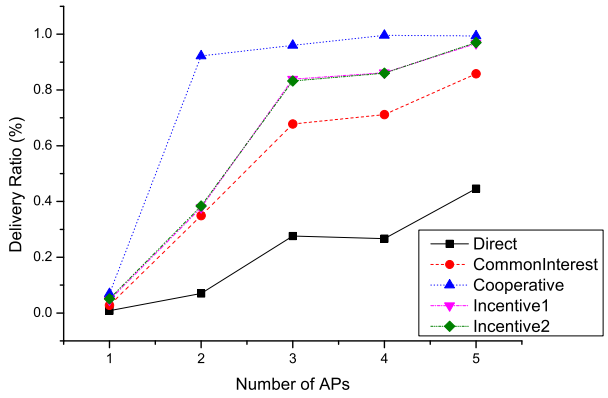
Algorithm 5: Optimization algorithm

Input: Node a, N_a, ${N'}_{a,i}$, $\bar{N}'_{a,i}$, t interests, the IAA matrix with format $u^a_{i,j}$ where i is the interest type and j is one of a's neighbors, the decision period Δ, and the number of node a's credits C_a

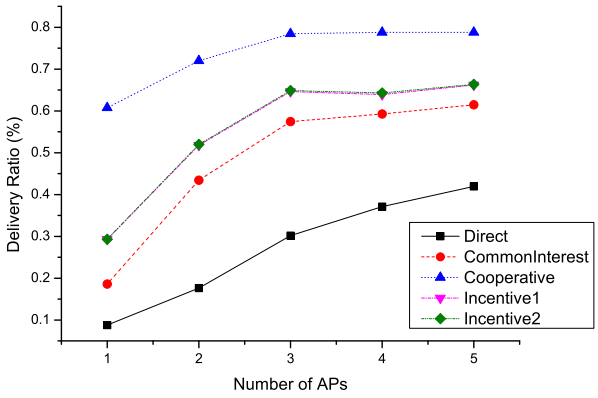
Output: Node a's rental decision set δ

1 $L_i = \emptyset$ for all $i \in T$.
2 Update $u^a_{i,j}$ and ${u'}^a_{i,j}$ for all $i, j \in N_a$.
3 At the beginning of each decision period Δ, calculate the maximum value of x such that $f(x) \leq C_a$. Note that it counts twice if a neighbor is rent to help fetch two kinds of interests.
4 For each round of renting, find $(i,q) = \underset{i,q \in \bar{N}'_{a,i} - L_i}{\arg\max}\ U_a^{i,q}$. If $\Delta\left(\frac{U_a^{i,q}}{f_a^{i,q}}\right) \geq \theta$, $L_i = L_i \cup \{q\}$. Otherwise, the algorithm terminates.
5 x=x-1.
6 If $x > 0$, repeat Step 2 through 5.
7 The algorithm terminates when $x \leq 0$ or for every interest i with $\delta^a_i = 1$, all $\delta^a_{i,j} = 1$ or there are no neighbors satisfying Formula 6.13 anymore.

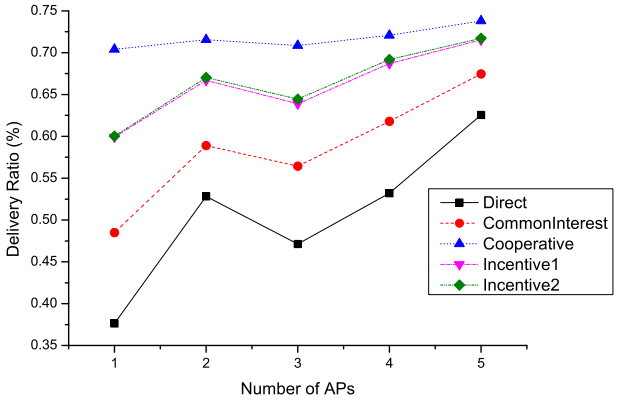
The complexity of the algorithm is $O(td)$ for each single decision, where t is number



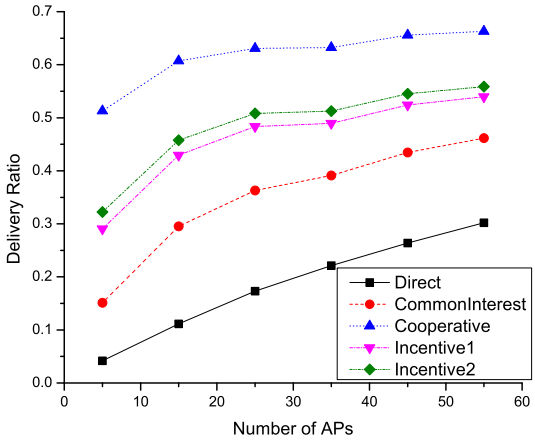
(a) UmassDisel06



(b) INFOCOM06

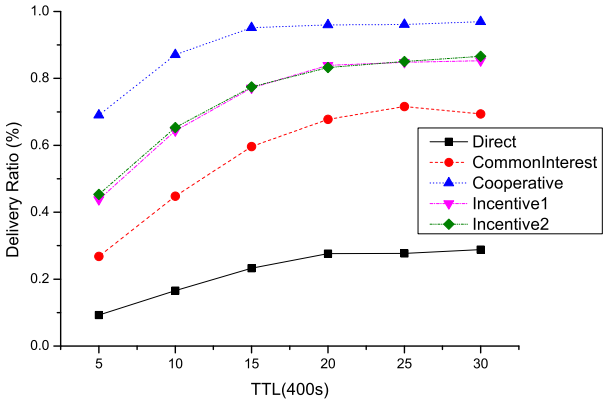


(c) SIGCOMM09

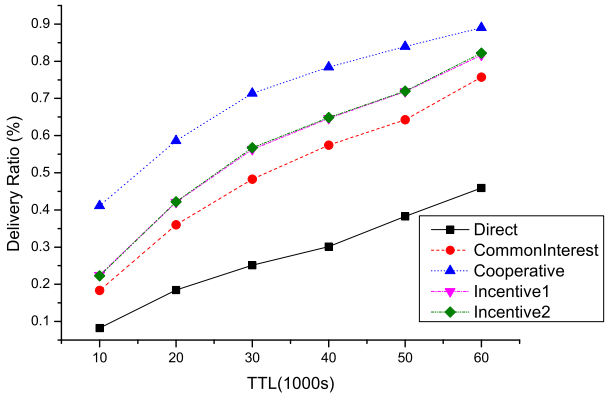


(d) MOBICOM06

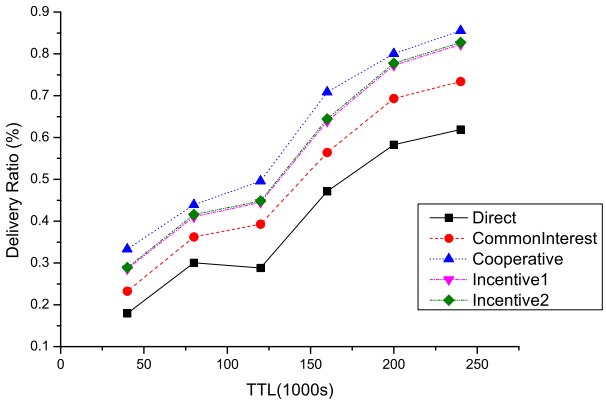
Figure 6.2. Comparison of delivery ratio on different data sets with variation of the number of APs.



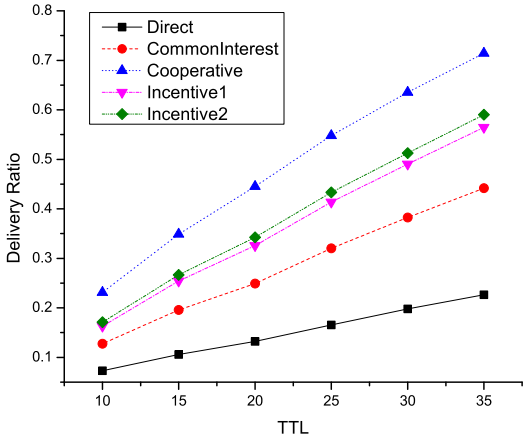
(a) UmassDisel06



(b) INFOCOM06



(c) SIGCOMM09



(d) MOBICOM06

Figure 6.3. Comparison of delivery ratio on different data sets with different TTLs.

of the interests and d is maximum number of neighbors. Even though the greedy algorithm dose not find the optimal rent set δ^*. It satisfies our expectation since computation resources are still limited in mobile devices and our solution requires low computation on each single device and can response quickly even though the decision period Δ is small.

6.6 ANALYSIS AND COMPLEMENT OF THE INCENTIVE SCHEME

6.6.1 Prepay function

Credit-based incentive scheme is based on fluent credits (virtual currency). The payment function is the core of the credit-based incentive scheme. It has significant influence on the overhead control and the health state of the system. As to a given node a, we already know $f(x)$ in Formula 6.7 is the total prepay function for node a and x in Formula 6.8 is the total number of renting times. We give the two payment functions as below:

$$f_1(x) = d \cdot x \tag{6.14}$$

where d is a constant number. Formula 6.14 means each rent costs the same number of credits.

$$f_2(x) = \sum_{t=1}^{x} p(t) \tag{6.15}$$

$$p(t) = \theta \cdot \frac{C_a - \sum_{s=1}^{t-1} p(s)}{C/n} \tag{6.16}$$

where θ is a constant and satisfies $0 < \theta < 1$, c_a is the number of available credits at node a, C is the total number of issued credits in the network, and n is the total number of nodes joined in the incentive scheme.

The difference between Formula 6.14 and Formula 6.15 is that Formula 6.15 increases the scale of each payment for the "rich" nodes and decreases the scale of each payment for the "poor" nodes. If we compare the system with the business market, our approach in

Formula 6.15 is similar to the tax strategy that balances the wealth in the society. The reason why we need Formula 6.15 is to maintain the health state of the system. When the credits are accumulated at some nodes with the running of the system, the fluency of the credits becomes weak and the system can hardly work, which indicates the system is unhealthy. The dynamic adjustment of Formula 6.15 can effectively improve the fluency of the credits.

In subsection 6.6.2, we analyze the influence of the two defined prepay functions on credits flow. In Section 6.7, we will further compare the performance of these two payment functions.

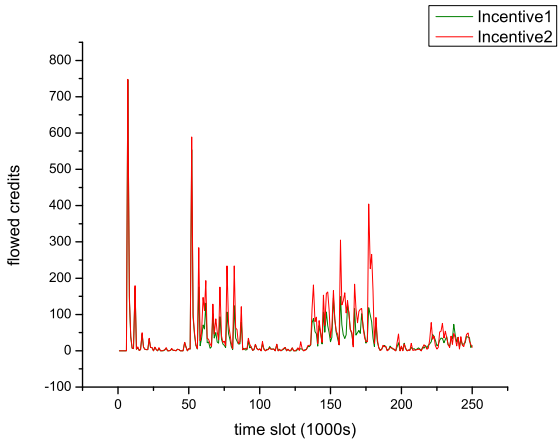
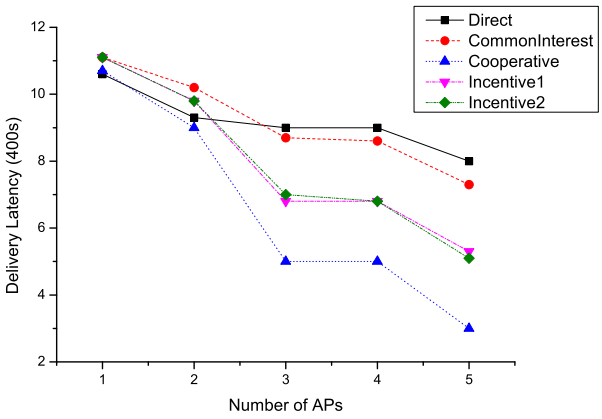


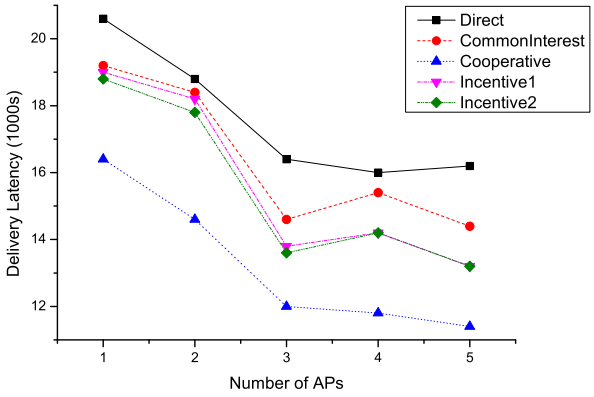
Figure 6.4. Credit flow analysis on real trace INFOCOM06.

6.6.2 Analysis of credits flow

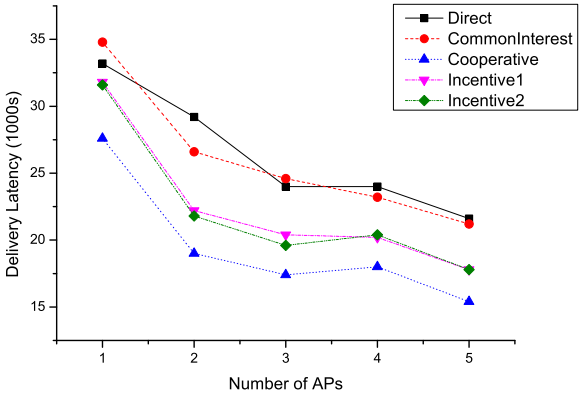
In the credit-based incentive scheme, the credit flow is indispensable to keep stimulating nodes to be cooperative. We analyze the credit flow of the two prepay functions under our proposed incentive scheme. The credit flow is the total transactions between all pairs of nodes at each time slot. Fig.5 is the analysis result on trace INFOCOM06. Each node is assigned with 15 credits before the simulation starts and 1000 messages are disseminated in 250 time slots. In Fig.5, we can see that "Incentive2" has a much active credits flow



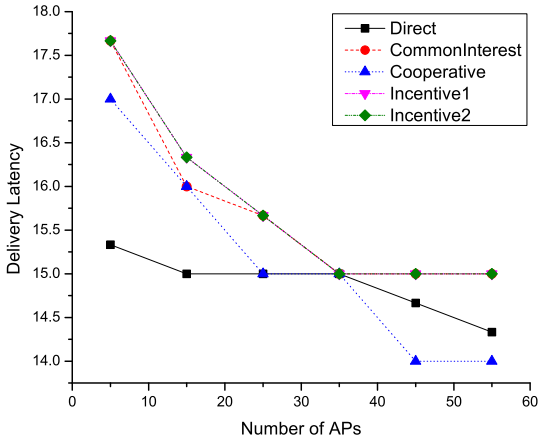
(a) UmassDisel06



(b) INFOCOM06

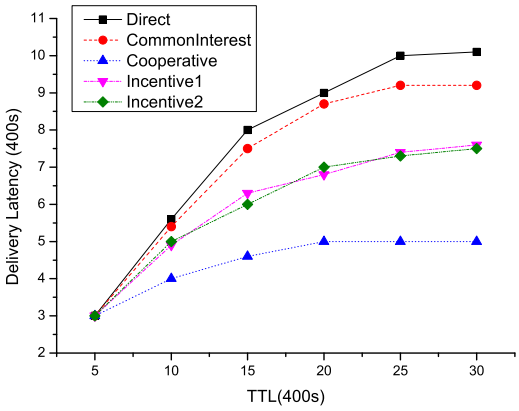


(c) SIGCOMM09

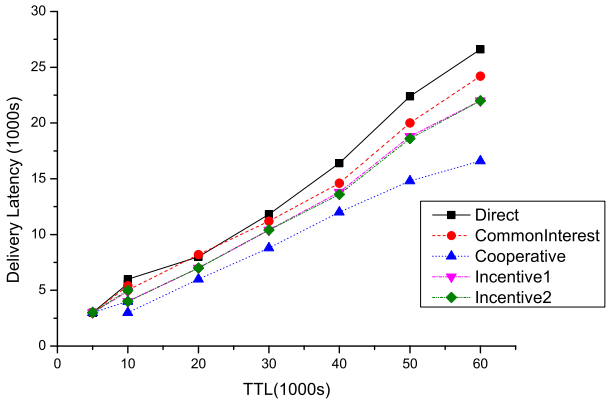


(d) MOBICOM06

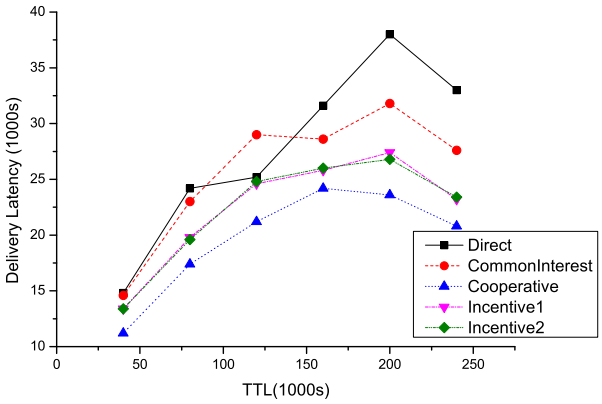
Figure 6.5. Comparison of delivery latency on different data sets with variation of the number of APs.



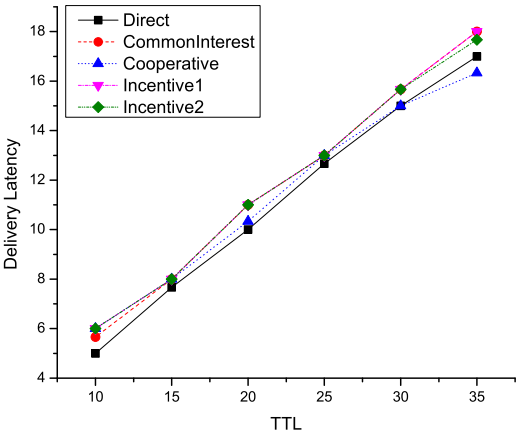
(a) UmassDisel06



(b) INFOCOM06



(c) SIGCOMM09



(d) MOBICOM06

Figure 6.6. Comparison of delivery latency on different data sets with different TTLs.

than “Incentive1”, indicating “Incentive2” can stimulate more nodes to be cooperative to help other nodes to get their interested messages. In section 6.7, we can learn further that “Incentive2” can achieve a better data dissemination ratio with a small gain of overhead than “Incentive1”. From Fig.5, we can find that there are several peaks of credit flow due to the contact pattern of the traces. During the peak period, there are more contacts happening. Another observation is that with the running of the simulation, the credit flow becomes lower and lower and finally maintains at a relative low level. This is also caused by the contact pattern of the trace. In reality, the contact frequency of the nodes in DTNs and MSNs also follows the power-law distribution. Even though “Incentive2” may increase the degree of credit flow to some extent, but it still cannot hold the credit flow at a high level. In our simulation, other prepay functions which further balance the difference of credits between the “rich” and “poor” nodes may increase the level of credit flow to some extent, but may suffer from the following two reasons. One is that they may violate the fairness and the assumption that nodes are selfish but rational. The other is that they incur more overhead over data dissemination since more poor nodes with lower IAA may be rented. In the future work, other than prepay, an auction way can be studied to see whether there is an increment of overall credit flow without incurring too much overhead.

6.6.3 Selfishness and misbehavior proofing

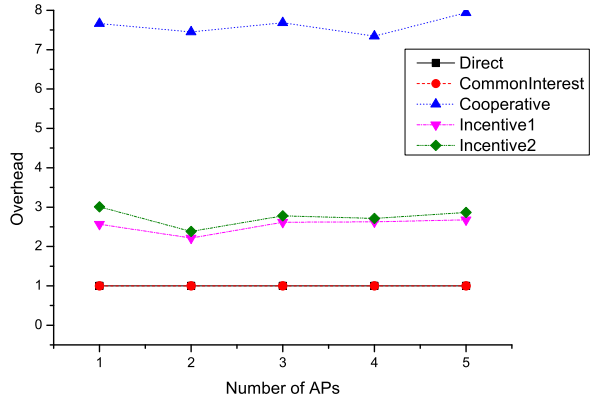
The proposed incentive scheme can stimulate selfish nodes and prevent malicious nodal collusion effectively. All the credits are cryptic messages that are issued by authenticated APs. If a hanode is extremely selfish and does not cooperate with other nodes, it has no chance to gain more credits once the initially issued credits are spent. The collusion among a small number of nodes also fails in cheating others. All the nodes evaluate other nodes through interest absorbing abilities and if the nodes lie its own records through the EWMA chart by decreasing the order of nodes which always fail to fetch interested messages. Moveover, no reputation is used in the network, and the collusion behavior can only work in their own cheating circle.

6.7 PERFORMANCE EVALUATION

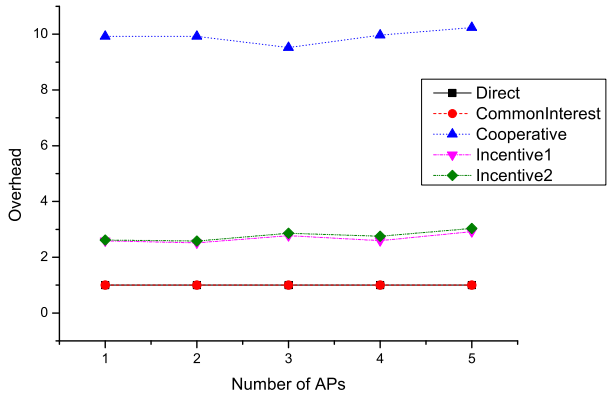
With the consideration of the prepay functions in Section 6.6.1, we name our credit-based schemes as "Incentive1" and "Incentive2" according to Formula 6.14 and Formula 6.15, respectively. For comparison, we also implement other three schemes: "Direct", "Common-Interest" and "Cooperative". In the "Direct" scheme, nodes can only fetch their interested messages directly from APs. In the "CommonInterest" scheme, nodes can get their interested messages from APs and nodes only interact with each other about the messages of common interest. In the "Cooperative" scheme, nodes are completely cooperative such that each node fetches all the messages no matter whether it is interested in or not. When a contact happens, two nodes will share all the messages they have. Since we do not limit the buffer size, "Cooperative" can be considered as the ground truth when we evaluate dissemination ratio and delay. Meanwhile, the number of copies of messages in "Cooperative" is also much higher than that of others, which leads to high overhead.

6.7.1 Simulation settings

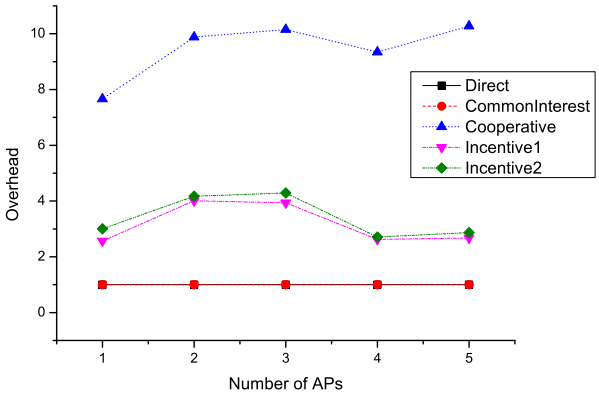
In this section, we evaluate our credit-based incentive scheme using three real traces INFOCOM06 [69], SIGCOMM09 [70] UMassDieselNet [71] and MOBICOM06 [72]. Discrete time is used in our simulations. E.g., we set 1000s as one time slot in the INFOCOM06 trace. The real trace INFOCOM06 consists of 78 users who are student volunteers in the conference INFOCOM 2006. Each student volunteer carried a mobile sensor equipped with a short-range radio. The contacts between each pair of nodes were recorded during four days. Similarly, we set 1000s for SIGCOMM09 and 400s for UmassDieselNet, separately. SIGCOM09 consists of 76 users in the conference SIGCOMM 2009. UMassDieselNet is a bus-based DTN testbed which consists of 36 buses. MOBICOM06 trace contains contact patterns among students, collected during the spring semester of 2006 in National University of Singapore. MOBICOM06 traces consists of 22341 students and 4885 class sessions. Students in each class session can contact with each other. MOBICOM06 trace is much



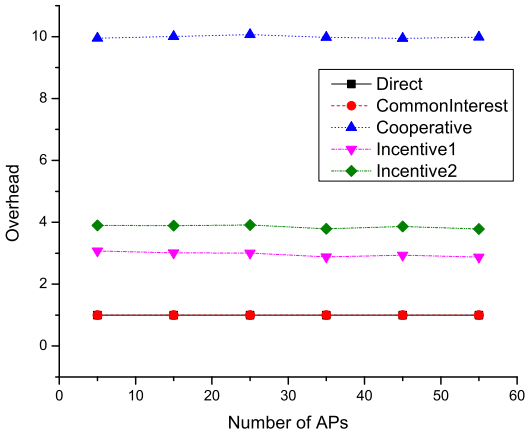
(a) UmassDisel06



(b) INFOCOM06

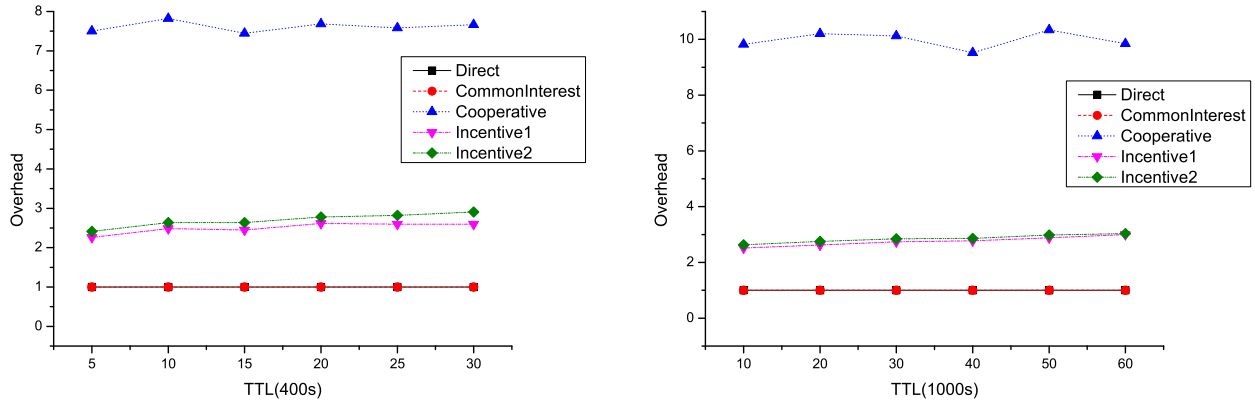


(c) SIGCOMM09

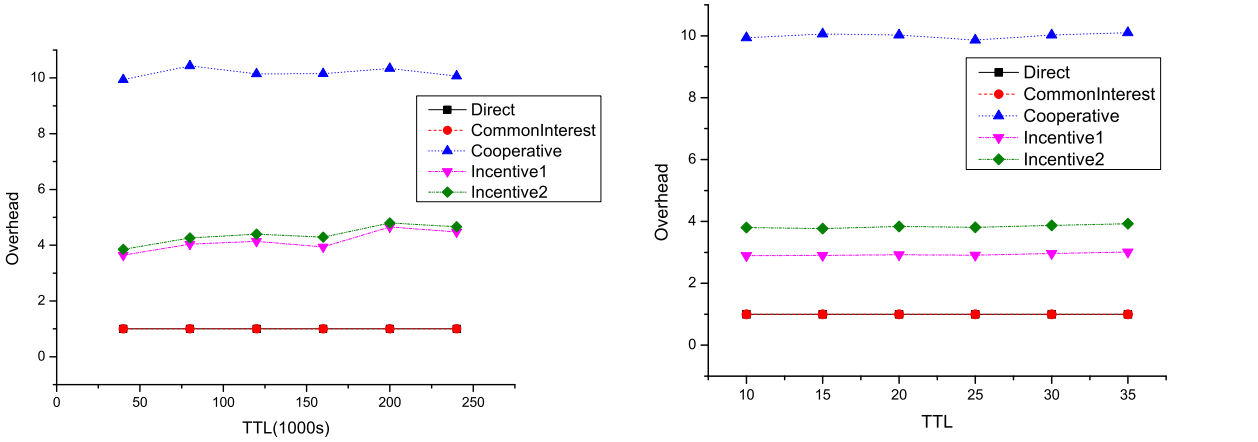


(d) MOBICOM06

Figure 6.7. Comparison of overhead on different data sets with variation of the number of APs.



(a) UmassDisel06 (b) INFOCOM06



(c) SIGCOMM09 (d) MOBICOM06

Figure 6.8. Comparison of overhead on different data sets with different TTLs.

bigger in terms of node size and duration. The original graph is very sparse. We compresses the trace on MOBICOM06 in order to speed up the simulation and make it a little denser. We only sample 5000 students and group each 5 students into one unit. Therefore, in the simulation, there are 1000 nodes. The first 4500 class sessions are selected and also group each 5 sessions into one time slot in the simulation. Compared with the other three real traces INFOCOM06, SIGCOMM09 and UMassDieselNet, the MOBICOM06 trace after preprocess is still much bigger, which is used to test the performance between "Incentive1" and "Incentive2" methods since larger datasets with larger time slots can better differentiate this two methods.

The default settings of simulations are as followings. We set the number of interest types to 15. Each node has 5 random interests on average. Initially, for traces INFOCOM06, SIGCOMM09 and UMassDieselNet, each node is assigned with 15 credits, for trace MOBICOM06, each node is assigned with 50 credtis. Messages are randomly generated in the content center and nodes can fetch messages from any AP. Totally there are 3000-5000 messages generated in the simulations in the three data sets. Each message has a Time-to-Live (TTL). When one message's TTL expires, it will be discarded from the nodes and the dissemination process of this expired message terminates. λ_1, λ_2 and τ in Formula 6.1, Formula 6.2 and Formula 6.4 are set to 0.01, 0.005 and 0.1, respectively. We evaluate our credit-based incentive scheme by varying the number of APs and the TTL of messages.

6.7.2 Comparison results

In this subsection, we evaluate the average results of the simulations on four real traces over 20 runs. Fig.3 plots the delivery ratio (data dissemination ratio) on three datasets with variation of the number of APs. All the three data sets show that "Cooperative" performs the best and our incentive schemes "Incentive1" and "Incentive2" outperform "CommonInterest" and "Direct". With the increment of the number of APs, the delivery ratio of all the schemes increases and gradually tends to be steady at some point. E.g., the delivery ratio of "Cooperative" on dataset UMassDieselNet barely increases after the number of APs

equals 4 and the delivery ratios of "Incentive1" and "Incentive2" on data set INFOCOM06 also tend to be steady after the number of APs reaches 3. The reason is that more APs bring more chances to the nodes to get interested messages directly and indirectly increasing the number of messages among the propagation. Fig.3 plots the delivery ratio on the three data sets with different TTLs. Longer TTLs can help the messages reach more nodes with more hops during the propagation increasing the delivery ratio. In both Fig.3 and Fig.4, "Cooperative" behaves the best and can be considered as the ground truth. Our incentive schemes "Incentive1" and "Incentive2" are about 10 percent better than "CommonInterest" on average. From the results on data set SIGCOMM09, we can see that "Incentive2" is slightly better than "Incentive1". With the limitation of the computation ability of the machine that the simulation runs on, the selected three data sets INFOCOM06, SIGCOMM09 and UMassDieselNet are not big enough to show the obvious difference between the two incentive schemes. In Fig.3 (d), we can see the the obvious advantage of "Incentive2" over "Incentive1". Also, we tried to run the simulations on our synthetic trace with 200 nodes and "Incentive2" outperforms "Incentive1" about 5 to 10 percent. Since up to now there is still no criteria for what kind of benchmark that can be qualified to be used as an MSN testbed for data dissemination, we do not show the results on our synthetic results.

Fig.6 and Fig.7 show the average delivery latency on the four datasets with the variation of the number of APs and different TTLs, respectively. "Cooperative" as the ground truth has the least latency since all the nodes are completely willing to carry all the messages they meet. Our two incentive schemes perform similarly and both outperform "CommonInterest". Which also means cooperation can effectively decrease the delivery latency.

Fig.8 and Fig.9 show the overhead on the four data sets with the variation of the number of APs and different TTLs, respectively. We only consider the number of copies as the overhead. When a message is delivered to the nodes who are not interested in, the overhead increases. It is very obvious that the overhead of "Cooperative" is much higher than all the other schemes. E.g., in Fig.8(b), the overhead of "Cooperative" is about 10 and the overhead of our two incentive schemes are under 3, the overhead of "CommonInterest" and

"Direct" are both 1 since no cooperation are allowed among the propagation and therefore no extra copy exists. On average, the overhead of "Incentive2" is a bit higher than "incentive1". From the observation of all the results for all the four data sets both in Fig.8 and Fig.9, we keep the overhead of our incentive schemes at a low level.

From all the above results, we can see that our incentive schemes can stimulate nodes to be more cooperative and therefore increase the data dissemination ratio and at the meantime control the overhead, which is very important in real world where resources are limited.

Chapter 7

MINIMIZE INFORMATION DIFFUSION TIME IN SOCIAL NETWORKS WITH ESTIMATED INFLUENCE COVERAGE

7.1 Introduction

The growing online network platform and potential application development of mobile social networks open doors for large-scale viral marketing. "word-of-mouth" effect can spread faster and reach a much larger influence of people compared to conventional advertising approaches. Viral marketing on social networks is to spread the information from a small group of people and the goal is to maximize the influence spread in a fast speed. Consider the following scenario as one example. A game company needs to test and advertise its new game product before finally releases it to the market. The way is to send limited free test samples to some users in its game network and let the users to spread news to influence others, The company wants to release the game as soon as the spread number reaches 2,000. then how to select users as the sample receivers from the network to minimize the release date is the problem that needs to be addressed in this situation, which is a case of the *information diffusion time minimization* problem with subject to a threshold of influence spread estimation. The problem will be formally defined in Section 7.2.

As we know there could be a great potential application need of considering diffusion time for some viral marketing, however, considering time factor to measure how fast the influence can spread in social networks has received little attention and study in all research fields. [17] addresses the information diffusion problem from the speed perspective for the first time. However, the goal of minimizing diffusion time in [17] does not consider the influence spread as a constraint, which may result into fast diffusion but lacks of influence spread number under some scenarios. A fast spread algorithm is meaningless if it lacks of the measurement of influence spread. The goal of influence diffusion time minimization problem

should correlate with expected influence spread. In related work, we will further discuss the shortage of work [17] with examples. In order to solve the *information diffusion time minimization* problem proposed in the paper, we need to define a new information diffusion model which incorporates time delays. In this situation, the classic probabilistic information diffusion models like IC and LT and other derivatives [10] [9][11][1][14] are not appropriate since they lack of considering time delays on edges between nodes. Directly adding a delay model may neither reflect influence propagation process in people's interactions with a meaningful combinations. Therefore, we define a new influence diffusion model named as sustaining cascading (SC) model which also defines diffusion delays in the model definitions. To simply analyze the hardness of the *information diffusion time minimization* problem, we may find that the problem may not even be solvable if the expected influence spread is not achievable with limited seeds. Telling whether the problem is solvable is a decision problem and can be simply answered by solving the *influence maximization* problem [10] to see whether the maximum spread is larger than the expected diffusion spread number. In this paper, we won't propose new algorithms to answer the decision problem since there are already many doing so. Through analysis of the network properties, a threshold is defined to limit the expected influence spread considered in this problem to make sure the problem is solvable. With a given network and the defined SC model, we prove the *information diffusion time minimization* problem is a NP-hard problem and we propose an approximation algorithm to solve the problem with an approximation factor. Due to the poor scalability of the approximation algorithm on large scale network graph, an heuristic algorithm is also designed to select the seed set with comparable diffusion time as well as with an acceptable computation running time.

7.2 System model

In this section, we first define the network model. Then we define a new information diffusion model, referred as sustaining cascading (SC) model. The *information diffusion time minimization* problem is also formally defined in this section.

7.2.1 Network Model

In our problem, a social network is modeled as an undirected graph, denoted as $G(V, E, D(E), P(E))$ where V is the set of nodes representing the individuals in the social network and $n = |V|$. Each individual node is denoted by u_i, where i marks the ith node. Each edge $e_{ij} = (u_i, u_j) \in E$ represents there is a social tie between node u_i and u_j. Each edge e_{ij} has two values p_{ij} and d_{ij}, representing the information diffusion probability and information diffusion delay, respectively. Therefore, $P(E)$ is the set of influence probability of each edge and $P(E)$ is defined as $P(E) = \{p_{ij}|(u_i, u_j) \in E, 0 \leq p_{ij} \leq 1, i < n, j < n\}$. Where p_{ij} indicates the probability that node u_i can successfully activate node u_j when u_i is active and vice versa. $D(E)$ is the set of information diffusion delay of each edge and $D(E) = \{d_{ij}|(u_i, u_j) \in E, 0 \leq d_{ij} \leq \theta_d, i < n, j < n\}$, where d_{ij} indicates the time cost when either u_i or u_j tries to influence each other and θ_d is a delay threshold in the information delay model, which is introduced in the subsection 7.2.4. Since the graph is modeled as undirected, e_{ij} is identical to e_{ji}, i.e., $p_{ij} = p_{ji}$ and $d_{ij} = d_{ji}$. For an ordinary node u_i, we rename the adjacent nodes of node u_i as neighbors of node u_i, denoted by $N_{u_i} = \{u_j|(u_i, u_j) \in E\}$.

7.2.2 Problem Formulation

For a given social network denoted as $G(V, E, P(E), D(E))$, let $\sigma(S, T)$ denote the arbitrary process of the influence spread from seed set S under the defined SC model within time T, where the seed set $S = \{s_1, s_2, ...s_k\}$ and diffusion time cost $0 \leq T \leq \infty$. The output of $\sigma(S, T)$ is a set of nodes in V influenced by seed set S directly or indirectly. The objective of the *information diffusion time minimization* problem is to select the seed set S containing k seeds to minimize diffusion time T while satisfying $|\sigma(S, T)| > \lambda$ and λ is a positive number $0 \leq \lambda \leq |V|$. Considering the dual problem of the *information diffusion time minimization* problem, which maximizes the influence spread $\sigma(S, T)$ with subject to a time limitation T, when $T = \infty$, for a chosen seed set S, the influence spread reaches the maximum, denoted as $\sigma^*(S, T)$. Therefore, when $\sigma^*(S, T) < \lambda$, there is no seed set S with k seeds that could satisfy the condition, i.e., there is no solution. In this paper, we consider the

condition where $\lambda < \sigma^*(S,T)$ to make the *information diffusion time minimization* problem solvable. In next section, we also design a method with consideration of specific network properties to limit the upper bound of λ.

7.2.3 Diffusion Model: Sustaining Cascading Model

In this paper, we propose a new information diffusion model, named the sustaining cascading model, which implies an successful influence to node u_i is due to the correlated influence from its neighbors N_{u_i}. The sustaining cascading (SC) model is described as follow.

For a given social network, an ordinary node u_i has three states *neutral*, *pending* and *active*. For a node u_i, We define *neutral* as a state being inactive and has never been influenced by others. The initial state for all nodes in the graph is *neutral*, except for the nodes that are first selected as seeds. The seeds are initialized as *active* and spread influence originally. When node u_i is influenced by other active neighbors successfully, u_i becomes *active* from *neutral*, otherwise, u_i becomes *pending* from *neutral*. If u_i is in *pending* state and is further influenced by others, u_i may become active from *pending* state and then try to influence its neighbors N_{u_i}. The transition process is depicted in Fig.7.1.

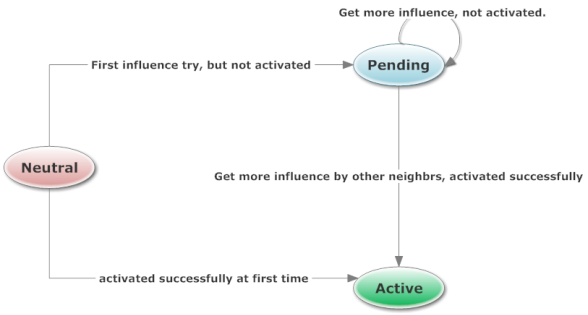


Figure 7.1. Node state transition diagram.

Compared with independent cascading (IC) model, the defined sustaining cascading model is expected to admit and reflect the influences from the neighbors are correlated under some constraint. Below we define two threshold parameters L_{u_i} and H_{u_i} for each node u_i, which represent the lower overlapping influence trigger and higher overlapping influence

trigger, respectively. Only when an attempt of influence from a neighbor u_j falls in the range $L_{u_i} < p_{ij} < H_{u_i}$, it can be considered to influence u_i with other neighbors together, otherwise, the node u_j will try to influence node u_i independently just like in IC model. By using the threshold constraint in SC model, We simply filter out the minor and dominant influences from the consideration of overlapping influence in SC model to strengthen the effect of overlapping influence. And it reflects the real scenario as well where people's state is not simply influenced and determined by a bunch of strangers and people may value their important friends exclusively. The sustaining cascading model is defined formally by equation.7.1 and equation.7.2. Equation.7.1 is referred as active influence probability and equation.7.2 is referred as passive influence probability, respectively later in this paper. $P_n(u_i)$ means node u_i is taking the nth round influence from its neighbors. In the active influence probability equation, the neighbor u_j will try to activate node u_i with probability $P_n(u_i)$ if p_{ij} satisfies the condition in equation.7.1. If node u_i is not activated by node u_j in the current round, $P_n(u_i)$ will be a threshold for later other neighbors. Only a neighbor in round $n+1$ with edge influence probability higher than $P_n(u_i)$, will trigger an active influence on node u_i, otherwise, the passive influence probability applies, where the activation threshold holds. In the active influence probability equation.7.1, when p_{ij} satisfies $p_{ij} > P_{n-1}(u_i)$ and $p_{ij} < L_{u_i}$ or $p_{ij} > H_{u_i}$, the independent cascading applies. When p_{ij} satisfies $p_{ij} > P_{n-1}(u_i)$ and p_{ij} falls in the range (L_{u_i}, H_{u_i}), the overlapping probability is considered to strengthen the influence probability at nth round. The thresholds we use in equation.7.1 and equation.7.2 are chosen as $L_{u_i} = 0.2$ and $H_{u_i} = 0.7$. As for node u_i, if it is not selected as seed node, $P_0(u_i) = 0$. To better understand how sustaining cascading model works, let us look into the following example, Node A and B are neighbors of node C. when node A tries to influence node C with edge probability 0.3, if node C is not activated and changes to pending state, later when node B tries to influence node C with probability 0.5, then at this round, node C may get activated with probability $0.65 = 1-(1-0.3)(1-0.5)$, which is strengthened and has considered the previous influences. In another case, if node B tries to influence node C before node A does, node C first has probability 0.5 to be influenced, if node

C is not active, when A tries with probability 0.3, node C will not be influenced at all cause the edge probability P_{AC} is not enough to trigger the influence. Afterwards, node C can only get a chance to become active if it meets other active neighbors with higher probability than 0.5. In the first scenario, node C has an overall higher probability to become active. While in the second scenario, node C has a higher probability to become active sooner.

$$P_n(u_i) = \begin{cases} p_{ij}, \;\; p_{ij} \geq \max(P_{n-1}(u_i), H_{u_i}) \; or \\ \qquad P_{n-1}(u_i) \leq p_{ij} \leq L_{u_i} \\ 1 - (1 - P_{n-1}(u_i)) \cdot (1 - p_{ij}), \\ \qquad \max(P_{n-1}(u_i), L_{u_i}) \leq p_{ij} < H_{u_i} \end{cases} \tag{7.1}$$

$$P_n(u_i) = P_{n-1}(u_i), \;\; p_{ij} < P_{n-1}(u_i) \tag{7.2}$$

Where $P_n(u_i)$ is the probability that u_i becomes active because of u_j's influence attempt at *nth* round .

From equation.7.1, we can get the accumulative probability that node u_i is active after n rounds of influence from its neighbors as below.

$$\begin{aligned} &\Gamma_n(u_i) = P_1(u_i) + (1 - P_1(u_i)) \cdot P_2(u_i) \\ &+ \ldots + \prod_{k=1}^{n-1} (1 - P_k(u_i)) \cdot P_n(u_i) \\ &= \Gamma_{n-1}(u_i) + \prod_{k=1}^{n-1} (1 - P_k(u_i)) \cdot P_n(u_i) \end{aligned} \tag{7.3}$$

Once node u_i becomes active at time t, it will try to influence all its neutral and pending neighbors according to equation.7.1. The order of influence depends on the activation order of its neighbors N_{u_i} and the time delay d_{ij} between each node u_j and u_i.

Compared with IC, LT and WC models, SC model better reflects the human influence propagation in real world. SC model is not a pure probabilistic independent model. For the past influences, yet not successfully, they may still contribute to the future influences under some constraints, i.e., SC model has memories of the influence. E.g., every time a new active neighbor B tries to activate node A, it may trigger the comprehensive influence of all

previous influences to influence A together. Consider a real world example. Matt has two friends Sam and Lucy. Matt does not like baseball game. Sam tried to persuade Matt that baseball game is interesting on Monday but failed. On Thursday, Lucy tried Matt again and succeeded in persuading Matt to watch a baseball game and Matt loves it afterwards. The question is because of whom, Matt starts to like baseball games. In this case, apparently, both Sam and Lucy contribute to the persuasion to some extent. How we determine whose persuasion is effective and how Sam and Lucy may influence Matt are the problems. In SC model, it is committed the result that Matt becomes interested in baseball games was due to both Sam's and Lucy's persuasion effort. Sam's influence has sustaining effect in Matt's opinion in baseball games.

7.2.4 Delay Model

Since most real world data sets only has the topology information, which usually lack of delay information on edges, We use the *Random Delay* model to assign time delay on each edge $e_{ij} \in E$, where the delay on each edge is assigned a value between (d_l, d_h) and d_l and d_h are two delay thresholds.

7.2.5 Edge probability model

If network data sets lack of influence probability information, we apply pentavalency probability model to assign edge probability for each $e_{ij} \in E$.

$$p_{ij} = RAND(p_1, p_2, p_3, p_4, p_5),\ e_{ij} \in E \tag{7.4}$$

where $RAND()$ is a random function that returns a random value from the five probabilities. Note that in experiment, we may also use a trivalency probability model which selects a random value from three probabilites.

7.3 Problem Hardness Analysis

Theorem: The influence minimization problem under SC model is NP-hard.

Proof. We mentioned in subsection.7.2.2, instead of proving the influence minimization problem under SC model, we could consider the dual problem of the *information diffusion time minimization* problem with the goal of maximizing the influence spread $\sigma(S,T)$ and subject to the time cost limitation T, which is referred as *influence maximization problem with time limitation* problem. To prove *influence maximization problem with time limitation* is a NP-hard problem, we apply a many-to-one reduction from *influence maximization problem with time limitation* to a set cover problem, which is known as a NP-Complete problem. For any given seed set S, When consider a case when $T = \infty$, the influence spread $\sigma^*(S,T)$ reaches the maximum. We try to find whether there is a seed set S with $k = |S|$ seeds that could get a influence spread $\sigma(S,T) > \lambda$ and we denote the influenced node set as V, where $\lambda = |V|$. For each potential seed s_i, the spread is $\sigma(\{s_i\},T)$ individually. We consider one case that if an arbitrary node $u_j \in \sigma(\{s_i\},T)$, $p_{ij} = 1$, which means the influence is always successful as long as there is an edge e_{ij}. When we consider two seeds s_i and s_k and their individual processes of influence $\sigma^*(\{s_i\},T)$ and $\sigma^*(\{s_k\},T)$, respectively, it is easy to conclude that $\sigma^*(\{s_i,s_k\},T) \geq \sigma^*(\{s_i\},T) \cup \sigma^*(\{s_k\},T)$ because of SC model considers overlapping influence and previous influence effect. We define the overlapping gain spread of seed s_i and s_k as $\mu(\{s_i,s_j\}) = \sigma^*(\{s_i,s_k\},T) - \sigma^*(\{s_i\},T) \cup \sigma^*(\{s_k\},T)$. Then if an arbitrary node $u_j \in \mu(\{s_i,s_j\})$, we could simply remove the node from the node set V, i.e., remove node u_j out of consideration. Then the remaining nodes on the activation process is denoted as set V', will only involvse independent cascading and it is a scenario in IC model. Whether we can find a seed set S to activate the seed set V', i.e., $\sigma(S) > |V'| + k$, is equivalent to whether there exists k subsets that cover all nodes in the node set V', which is the set cover problem. If solutions for the set cover problem exists, then the *influence maximization problem with time limitation* problem is solvable as well. Therefore, The influence minimization problem subject to a influence spread under SC model is NP-hard.

7.4 Submodularity of SC model

We first defined SC model in our previous work (under review), where we already proved that the influence function $\delta(S, V)$ is submodular under an arbitrary instance of SC model. Studying submodularity of a information diffusion model is crucial to decide whether we could find an approximation algorithm and mark the potential maximum influence spread $\sigma^*(S, T)$ for an arbitrary seed set S. A submodular function $F(S)$ has the well-known property "diminishing return", which means the gain by adding one more element to set S will decrease gradually. With "diminishing return"property, a hill-climbing algorithm can be applied to achieve approximation ratio of $1 - 1/e$. In the scenario of applying SC model in information diffusion problems, we could apply a Monte-Carlo greedy algorithms to maximize the influence spread with limited time constraint with approximation ratio of $1 - 1/e$. We will introduce the greedy algorithm in section 7.5. In fact, there is no other algorithm so far can beat the approximation algorithm. Therefore, the nice fact of submodularity of SC model enables the approximation algorithm that could tell us the potential spread a given seed set can achieve under SC model. I.e., we could avoid setting an unreachable influence spread $\lambda > \sigma^*(S, T)$ as an constraint when applying to real world scenarios.

7.5 Approximation Algorithm

According to the following theorem:

Theorem: [10] For a non-negative, monotone and submodular function f, let S be a set of size k obtained by selecting elements one at a time, each time choosing an element that provides the largest marginal increase in the function value. Let S^* be a set that maximizes the value of f over all k-element sets. Then $f(S) \geq (1-1/e)\dot{f}(S^*)$. In other words, S provides a $(1 - 1/e)$ approximation.

We know we could design a Monte-Carlo simulation based hill-climbing greedy algorithm satisfying that the influence spread function $\sigma(S, T)$ is a non-negative, monotone and submodular function and therefore achieve the approximation ratio $1 - 1/e$.

The approximation algorithm is defined as below.

Algorithm 6: Approximation Algorithm

```
1 S = ∅
2 for i = 1 to k do
3 |   Select u = arg max_{w∈V\S}(σ(S ∪ {w}, T) − σ(S, T))
4 |   S = S ∪ {u}
5 output S
```

The nodes are selected into the seed set one by one in the greedy way. Optimally, if $|S| = 1$, the solution is optimal. In each round, we select the node that will maximize the influence spread gain if adding the node into the current seed set. Due to the limitation of experiment conditions, We run $R = 100$ times to approximate Monte-Carlo simulation for line 3 to approach the ground truth.

Now with the approximation algorithm, we could possibly answer the question what the spread coverage threshold λ should be set to be reasonable for the *influence minimization problem*. We denote the seed set selected from the approximation algorithm as S^*, then setting the spread coverage threshold $\lambda > \sigma(S^*, T)$ may not guarantee that we could find a possible solution. Therefore, $\sigma(S^*, T)$ can be considered as the upper bound for the coverage threshold λ. In the section 7.7, the spread coverage threshold λ is selected within the range $(0, \sigma(S^*, T))$ through testing.

7.6 Heuristic Algorithm on SC model

7.6.1 Preliminary

Given two nodes u and v, there are many simple paths. The set of simple paths is denoted as $P(G, u, v) = \{p1, p2, \cdots, p_n\}$. We consider each simple path separately. For each path $p_i = \langle e_1, e_2, \cdots, e_m \rangle$, where u is the end point of e_1 and v is the end point of e_m, the delays over the path p_i is denoted as $\langle d_1, d_2, \cdots, d_m \rangle$ and the propagation probability over the path p_i is denoted as $\langle p_1, p_2, \cdots, p_m \rangle$. Therefore, the diffusion delay of path p_i is

denoted as $d(p_i) = \sum_{i=1}^{m} d_m$ and the influence probability of successfully activating node v from u through path p_i is denoted as $p(p_i) = \prod_{i=1}^{m} p_i$. Further we define *fastest influence path* and *maximum influence path* to approximate the actual expected influence in the social network.

Definition 7.6.1. ***(Fastest Influence Path)*** *For a graph G, we define the fastest influence path between node u and v as*

$$p_s(u, v) = \arg\min_{p_i} \{d(p_i) | p_i \in P(G, u, v)\} \tag{7.5}$$

Definition 7.6.2. ***(Maximum Influence Path)*** *For a graph G, we define the maximum influence path between node u and v as*

$$p_m(u, v) = \arg\max_{p_i} \{p(p_i) | p_i \in P(G, u, v) \text{ and } p(p_i) \geq p(p_s(u, v))\} \tag{7.6}$$

The *fastest influence path* between node u and v is actually the shortest path between u and v with respect to the edge diffusion delays in a weighted graph. It means the minimum time required to possibly activate node v from node u. While the *maximum influence path* between node u and v is the path with the maximum influence probability. Note that *fastest influence path* could be the same with the *maximum influence path* when a path with the maximum influence probability costs the least time. When two paths are different, since the *maximum influence path* is expected to have influence effect on node v later than the *fastest influence path*, the path $p_m(u, v)$ is strengthened as $1 - (1 - p_s(u, v))(1 - p_m(u, v))$ when satisfying equation.7.1 in SC model.

For an arbitrary node u, we define the *fastest influence arborescence(FIA)*, which is the union of the fast influence paths from u and the *maximum influence abrorescence(MIA)*, which is the union of the maximum influence paths from u. We use *FIA* and *MIA* to evaluate the potential of node u as a seed. We user an influence probability threshold θ to filter *FIP* and *MIP* that have too small propagation probabilities.

Definition 7.6.3. ***(Fastest Influence Arborescence)*** *For a graph G, we define the*

fastest influence arborescence between node u and v as

$$FIA(u) = \bigcup_{v \in V, p(p_s(u,v)) > \theta} p_s(u, v) \tag{7.7}$$

Definition 7.6.4. ***(Maximum Influence Arborescence)*** *For a graph G, we define the maximum influence arborescence between node u and v as*

$$MIA(u) = \bigcup_{v \in V, p(p_m(u,v)) > \theta} p_m(u, v) \tag{7.8}$$

Both *Fastest Influence Arborescence* and *Maximum Influence Arborescence* give the local influence regions of node u, the probability threshold θ controls the size of two arborescence. The *Maximum Influence Arborescence* depends on the *Fastest Influence Arborescence*.

From the definition of *MIA*, we can find the *MIA* model is a simplified SC model, which underestimates the influence propagation between two given nodes u and v. However, With the *Maximum Influence Arborescence*, we can easily estimate a reasonable spread coverage threshold λ by applying the influence spread function $\sigma(S, T)$ on the MIA instead of running Monte-Carlo simulations on the SC model.

The influence spread over $MIA(u)$ is calculated as

$$\sigma(u) = \sum_{p_i \in MIA(u)} p(p_i) \tag{7.9}$$

And maximum time cost over $MIA(u)$ is calculated as

$$T(u) = \max_{p_i \in MIA(u)} d(p_i) \tag{7.10}$$

7.6.2 The Heuristic Algorithm

In this section, we design a heuristic algorithms to minimize the diffusion time subject to an expected influence coverage guarantee.

The idea of the heuristic algorithm relies on the *Maximum Influence Arborescence*. We

first calculate the *Maximum Influence Arborescence* for each arbitrary node $u \in V$. We pre-select top $\omega \cdot k$ nodes from the node set V by ranking the expected influence spread number in the *Maximum Influence Arborescence*, where k is the number of seeds and ω is a constant number. Then we greedily select nodes with highest ratio of influence spread number over time cost. After selecting all the seeds, if the influence spread $\sigma(S,T) < \lambda$. It means some seeds with highest ratio of influence spread number over time cost may not have a fair total influence spread. Therefore, we could adjust the seed set by removing the seed with smallest coverage and continue to find next node with highest ratio of influence spread number over time cost.

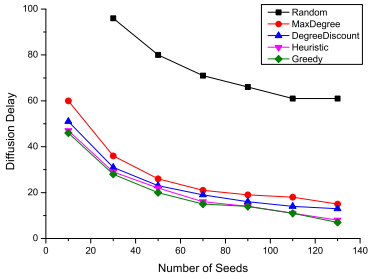
The heuristic algorithm is defined as below.

Algorithm 7: HEURISTIC ALGORITHM

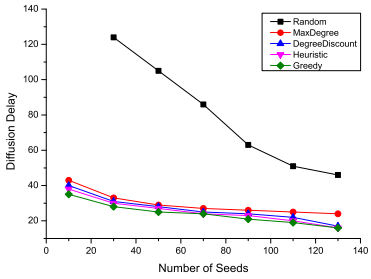
```
S = ∅, H = ∅, R = ∅, n = |V|, k = |S|, initialization θ, ω, λ, T
for u in V do
    calculate MIA(u);
for i = 1 to ω · k do
    select u = max_{w∈V\H}(σ(w))
    H = H ∪ {u}
for i = 1 to k do
    select u = max_{w∈H\S}(σ(w)/T(w))
    S = S ∪ {u}
while σ(S,T) < λ do
    select u = min_{w∈S}(σ(w))
    S = S\{u}
    R = R ∪ {u}
    select v = max_{w∈H\(S∪R)}(σ(w)/T(w))
    S = S ∪ {v}
output S
```

In Algorithm.7, we first filter the nodes to get the potential seed set H, in which, all nodes have fair spread coverage. We prefer the seeds with fast growing spread (influence spread/diffusion time cost) in the seed set H. If the selected seed set H satisfies the coverage guarantee λ, we output the seed set S, otherwise, we drop the seed with the minimum spread coverage in seed set S and pursue next fast-growing seed. Set R takes the dropped seeds.

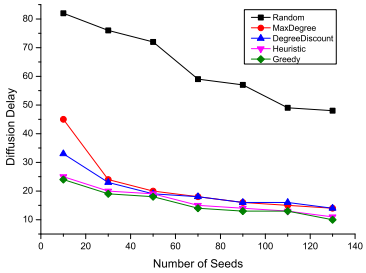
As long as we chose a reasonable spread threshold λ using the defined *MIA*, we could always find a solution.



(a) NetHEPT



(b) Ego-Facebook



(c) Wiki-vote

Figure 7.2. Comparison of diffusion delay of different algorithms on different data sets with increasing number of seeds.

7.7 Performance Evaluation

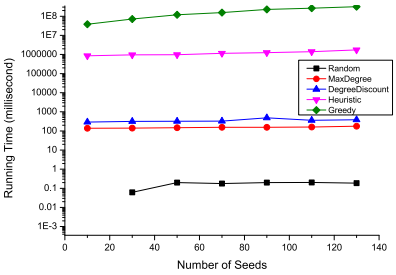
In this section, we apply the heuristic algorithm against several other algorithms including the approximation algorithm to test on the sustaining cascading model. We use two metrics to evaluate the algorithms. One is diffusion delay. Given an influence coverage threshold, the best algorithm should have minimum diffusion delay. Another metric is running time of seed selection process.

7.7.1 Experimental Setup

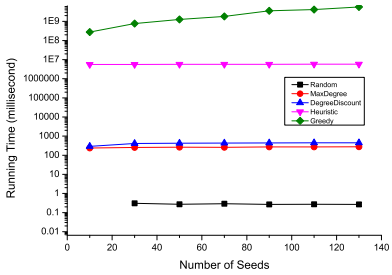
We use three real network graph data sets as test-beds. The first data set we use is *NetHEPT* which contains $15,233$ nodes and $58,891$ edges. *NetHEPT* is a snapshot of a collaboration network among authors writing papers. The second network graph is called *ego-Facebook* data set [12], which consists of $4,039$ nodes and $88,234$ edges. The number of Triangles in the network is $1,612,010$. The data was collected from survey participants using an online application which could provide users' basic information in 2012. The thrid network data is *Wiki-vote* [6], which consists of $7,115$ nodes and $103,689$ edges. The number of Triangles in the network is $608,389$. According to [6], the network contains all the Wikipedia voting data from the inception of Wikipedia till January 2008. The network is addressed as an undirect network in this paper. For all the data sets, we use the *Random Delay* model to assign a random delay value between 0 and 30 to each edge in the graphs. For *NetHEPT*, the influence probability on each edge is chosen randomly from $\{0.01, 0.05, 0.1, 0.2, 0.3\}$. For *Wiki-vote* and *ego-Facebook*, these two graph data sets are more dense than *NetHEPT*. Then for these two graph data sets, the influence probability on each edge is chosen randomly from $\{0.01, 0.05, 0.1\}$. The influence spread threshold is 600 for all three data sets (Note that 600 does not include the initial seed set).

7.7.2 Algorithms Introduction

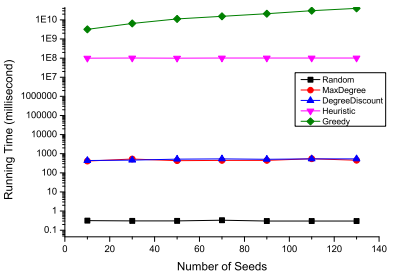
We run the following algorithms under the SC model against our heuristic algorithm on the introduced network data sets. The first one is is **Random**, which takes constant time to randomly select the seed set. It can be used as a base line for evaluating our heuristic algorithm. The second one is **MaxDegree**. This simple heuristic selects top k seeds with the largest degrees and first used in [10]. The time cost is linear and only related to the number of nodes in the network. The influence spread number in existing approaches are much better than **Random**. Considering often two seeds may cover the same spread, which is considered as bad in independent cascading model, An improved Degree-counting based algorithm called **DegreeDiscount** is proposed in [2] to add a discount of spread gain on a node depending on



(a) NetHEPT



(b) Ego-Facebook



(c) Wiki-vote

Figure 7.3. Comparison of running time of different algorithms on different data sets with increasing number of seeds.

how many neighbors of the node has been influenced by other picked seeds. Also, we apply the **Greedy** algorithm (It is an approximation algorithm) in Algorithm.6 and our heuristic algorithm 7.

For all the above algorithms, once a seed set is chosen, the influence spread running all real data sets are ran 100 times and output the average spread value.

7.7.3 Experimental results

We discuss the experimental results of different algorithms on different data sets in this subsection.

We first look at the diffusion delays of different algorithms on all three real data sets in Fig.7.2. The influence spread number is set to 600. The diffusion delays reveal which algorithm could successfully influence 600 nodes with shortest time. From Fig.7.2, we can see that *Random* uses the most time since it does not evaluate the influence abilities, neither the influence efficiencies of different nodes. *MaximumDegree* and *DegreeDiscount* spreads much faster than *Random* and *DegreeDiscount* is slightly better than *MaximumDegree*. These two methods evaluates the influence abilities from the perspective of degree centrality, which prefer the nodes with high degrees or discounted high degrees. Both *DegreeDiscount* and *MaximumDegree* do not consider the influence delay on each edges. Our defined *Heuristic* algorithm further reduces the diffusion delay. The *Greedy* method always look for the most influential node to add to the seed set and is slightly better than our *Heuristic* algorithm. However, the *Greedy* method may have unacceptable running time cost.

Fig.7.3 plots the running time of different algorithms. Only the calculation of the seed set is counted in the running time cost. The y-axis in Fig.7.3 is in Log10 scale since *Heuristic* and *Greedy* costs magnitudes more than the degree centrality based algorithms and *Random* algorithm. For each data set, the running time is increasing according to the increasing number of seeds in all data sets, though not clear in Fig.7.3. *Random* algorithm theoretically does not cost time to select seeds, in this simulation, it costs around 0.1 milliseconds. Both *MaxDegree* and *DegreeDiscount* costs less than 1 second overall in all three data sets since

they only calculate the degrees of nodes and are linear to the size the network. Our *Heuristic* may need hours or about a day to select the seed sets, which is still acceptable. However, the *Greedy* algorithm may costs days to months to finish the seed selection process, which is unrealistic. The *Greedy* algorithm is not scalable and may never finishes the seed set selection process when facing a even larger network with millions of nodes and tens of millions of edges. Compared with the *Greedy* algorithm, our defined *Heuristic* algorithm is relatively scalable because seeds are only selected and replaced from a preselected seed set.

Through the study on Fig.7.2 and Fig.7.3, We can see that *Heuristic* algorithm has the most fast influence spread speed with reaching an achievable influence coverage threshold in an acceptable running time.

Chapter 8

CONCLUSION

In this dissertation, we propose network models and algorithms in the fields of data dissemination and information diffusion in social networks. We briefly introduced two kinds of social networks including mobile social networks and large-scale online social networks. We introduce all model design and algorithms to resolve the most important issues in these social networks. Most of all approaches are optimization-based. Below, We briefly conclude each of our works.

In chapter 3, we propose a prediction-based routing protocol with packet scheduling. Considering the time constraint of the messages, we add a dose of altruism to the protocol in order to increase the delivery ratio. We first model the contact graph and then derive the ability graph which provides the probabilistic foundation for the decision of forwarding messages during each contact. We also propose a greedy method and the optimal forwarding schedule. We formulate the optimal forwarding problem as a maximum utility forwarding scheduling model and then transform it to the maximum bipartite matching problem. The simulation results show that our approach improves delivery ratio. Meanwhile, the overall delivery latency is reduced compared with the method without considering scheduling the packets under time constraint. Considering the traditional data dissertation which expands from one-to-one packet delivery to one-to-multiple data propagation, this work could potentially help increase delivery ratio.

In chapter 4, we study the data dissemination problem with controlled overhead which is defined from the perspective of assigned authorized server copies for each message from the APs. We use a time-homogeneous Markov model to analyze the interest transition of every node's neighbors and further define two utility functions to evaluate the service ability of nodes for a specific kind of interest. The simulations show our proposed methods can

effectively increase the data dissemination ratio. Compared with most existing works in literature, this work is the first one to increase data dissemination ratio more efficiently from the perspective of controlling the network resources.

in chapter 6, we study how to strengthen nodal cooperation for data dissemination in MSNs. We propose a new credit-based incentive scheme to stimulate selfish nodes to be more cooperative to disseminate the messages even if they are not interested in. The proposed incentive scheme is fair to all the nodes and can effectively prevent malicious nodal collusion. We define an optimization function and each node in the incentive scheme tries to maximize their own benefits by gaining more chances to get their interested messages while paying less. We implement our incentive scheme in four real traces with different scales. As a result, each selfish node is well motivated by their own interest and meanwhile the data dissemination ratio and delay of the whole network are improved with a small increment of overhead. This paper contributes to making delay-tolerant network protocols more applicable.

In chapter 5 and 7, In these works, we propose a new information diffusion model, namely, the sustaining cascading (SC) model. We believe SC model has a better representation of the real world information diffusion process than the existing models. We prove the influence maximization problem under SC model is NP-hard and the diffusion function under SC model is submodular. The classic hill-climbing greedy method with $1-1/e$ approximation ratio can also be applied under the SC model. To improve the greedy method, with consideration of SC model property, we propose a new heuristic algorithm. We also conduct extensive experiments to test out the SC model and the new heuristic algorithm. Based on the SC model, we could address the *information diffusion time minimization* problem. The SC model could incorporate time delays. We adopt the classic degree-based algorithms as well as the *Approximation* algorithm in the SC model to go against the new designed *Heuristic* algorithm in metrics of diffusion delays and running time costs. We conduct extensive experiments to test out the SC model and the new *Heuristic* algorithm. The new designed *Heuristic* algorithm has the minimum diffusion delays on the tested data set with acceptable running time costs. In future, this SC model makes it possible to propose more heuristic

algorithms that could be designed to either further minimize the diffusion delay or reduce the running time costs of seed selections.

Bibliography

[1] L. V. S. Lakshmanan A. Goyal, F. Bonchi and S. Venkatasubramanian. On minimizing budget and time in influence propagation over social networks. *Social Network Analysis and Mining, 2(1)*, 2012.

[2] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208. ACM, 2009.

[3] Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1029–1038. ACM, 2010.

[4] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 57–66. ACM, 2001.

[5] M. Fisher G. Nemhauser, L. Wolsey. An analysis of the approximations for maximizing submodular set functions. In *Mathematical Programming*, pages 265–294, 1978.

[6] J. Kleinberg J. Leskovec, D. Huttenlocher. Predicting positive and negative links in online social networks. *WWW*, 2010.

[7] L. Adamic J. Leskovec and B. Adamic. The dynamics of viral marketing. *ACM Transactions on the Web (ACM TWEB)*, 2007.

[8] Kyomin Jung, Wooram Heo, and Wei Chen. Irie: Scalable and robust influence maximization in social networks. *arXiv preprint arXiv:1111.4795*, 2011.

[9] K. Ohara K. Saito, M. Kimura and H. Motoda. Selecting information diffusion models over social networks for behavioral analysis. *ECML PKDD*, 2010.

[10] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[11] K. Ohara M. Kimura, K. Saito and H. Motoda. Learning information diffusion model in a social network for predicting influence of nodes. *Intell. Data Anal.15(4):633 - 652*, 2011.

[12] J. McAuley and J. Leskovec. Learning to discover social circles in ego networks. *Advances in Neural Information Processing Systems*, 2012.

[13] Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 61–70. ACM, 2002.

[14] M. J. Streeter and D. Golovin. An online algorithm for maximizing submodular functions. In *Proceedings of Neural Information Processing Systems*, pages 1577–1584, 2008.

[15] Yu Wang, Gao Cong, Guojie Song, and Kunqing Xie. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1039–1048. ACM, 2010.

[16] Honglei Zhuang, Yihan Sun, Jie Tang, Jialin Zhang, and Xiaoming Sun. Influence maximization in dynamic social networks. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 1313–1318. IEEE, 2013.

[17] Guohong Cao Zongqing Lu, Yonggang Wen. Information diffusion in mobile social networks: The speed perspective. In *INFOCOM*, pages 1932 – 1940. IEEE, 2014.

[18] Evan P.C. Jones and Paul A.S. Ward, *Routing Strategies for Delay-Tolerant Networks*, submitted to Computer Communication Review, http://www.ccng.uwaterloo.ca/ pasward/Publications/dtn-routing-survey.pdf.

[19] M. Mauve, A. Widmer, and H. Hartenstein, *A survey on position-based routing in mobile ad hoc networks*, IEEE Network, vol. 15, no. 6, pp. 3039, 2001.

[20] R. D. Poor, *Gradient routing in ad hoc networks*, MIT Media Laboratory, unpublished manuscript, http://www.media.mit.edu/pia/Research/ESP/texts/poorieeepaper.pdf, 2000.

[21] Q. Yuan, I. Cardei, and J. Wu, *An Efficient Prediction-Based Routing in Disruption-Tolerant Networks*, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 23, NO. 1, JANUARY 2012.

[22] J. Wu, M. Xiao, and L. Huang, *Homing Spread: Community Home-based Multi-copy Routing in Mobile Social Networks*, accepted to appear in Proc. of INFOCOM 2013.

[23] N. Ristanovic, G. Theodorakopoulos and J-Y Le Boudec, *Traps and Pitfalls of Using Contact Traces in Performance Studies of Opportunistic Networks*, In INFOCOM 2012.

[24] L. Ma, X. Cheng, F. Liu, F. An and J. Rivera, *iPAK: an in-situ pairwise key bootstrapping scheme for wireless sensor networks*, IEEE Transactions on Parallel and Distributed Systems, vol. 18(8), 1174-1184. 2007.

[25] B. Zhang, X. Cheng, N. Zhang, Y. Cui, Y. Li and Q. Liang, *Sparse target counting and localization in sensor networks based on compressive sensing*, INFOCOM, 2011, pp. 2255-2263.

[26] L. Ma, AY. Teymorian and X. Cheng, *A hybrid rogue access point protection framework for commodity Wi-Fi networks*, INFOCOM, 2008, pp. 1220-1228.

[27] S. Ji, Z. Cai, Y. Li and X. Jia. *Continuous Data Collection Capacity of Dual-Radio Multichannel Wireless Sensor Networks*, IEEE Transactions on Parallel and Distributed Systems, 23 (10), 2012, pp 1844-1855.

[28] Z. Cai, S. Ji and J. Li. *Data caching based query processing in multi-sink wireless sensor networks*, International Journal of Sensor Networks 12(2), 2012, pp 109-125.

[29] S. Ji and Z. Cai *Distributed data collection in large-scale asynchronous wireless sensor networks under the generalized physical interference model*, IEEE/ACM Transactions on Networking, 21(4), 2013, pp 1270-1283.

[30] S. Cheng, J. Li and Z. Cai. *O(ϵ)-Approximation to Physical World by Sensor Networks*, INFOCOM, 2013, pp. 3084-3092.

[31] C. Ai, L. Guo, Z. Cai and Y. Li. *Processing area queries in wireless sensor networks*, Mobile Ad-hoc and Sensor Networks, 2009, pp 1-8.

[32] Z Cai, G. Lin and G. Xue. *Improved approximation algorithms for the capacitated multicast routing problem*, COCOON, 2005, pp 136-145.

[33] Jing (Selena)He, Shouling Ji, Yi Pan, and Yingshu Li, *Constructing Load-Balanced Data Aggregation Trees in Probabilistic Wireless Sensor Networks*, to appear in the IEEE Transactions on Parallel and Distributed Systems (TPDS), 2013.

[34] W. Gao, Q. Li, B. Zhao, and G. Cao, *Multicasting in Delay Tolerant Networks: A Social Network Perspective*, in ACM MobiHoc, 2009.

[35] K. J. Lin, C. Chen and C. F Chou, *Preference-Aware Content Dissemination in Opportunistic Mobile Social Networks*, In INFOCOM 2012.

[36] S. Ioannidis, A. Chaintreau, and L. Massoulie, *Optimal and Scalable Distribution of Content Updates over a Mobile Social Network*, in INFOCOM, 2009.

[37] D. B. West, *Introduction to Graph Theory (2nd Edition)*. PrenticeHall, 1999.

[38] A. Itai, Y. Perl, and Y. Shiloah, *The complexity of finding maximum disjoint paths with length constraints. Networks*, 277-286, 1982.

[39] H. Kuhn, *The hungarian method for the assignment problem*, Naval research logistics quarterly, vol. 2, no. 1-2, pp. 83-87, 1955.

[40] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, *Pocket Switched Networks and Human Mobility in Conference Environments*, in ACM SIGCOMM workshop on Delay-tolerant networking (WDTN), 2005.

[41] A. Pietilainen and C. Diot, *CRAWDAD data set thlab/sigcomm2009 (v. 2012-07-15)*, Downloaded from http://crawdad.cs.dartmouth.edu/thlab/sigcomm2009, July, 2012.

[42] N. Kayastha, D. Niyato, P. Wang, and E. Hossain, "Applications, architectures, and protocol design issues for mobile social networks: A survey,"Proc. of the IEEE, vol. 99, no. 12, pp. 2130 - 2158, dec. 2011.

[43] G. Liu, S. Ji, J. Wu and Z. Cai, "Credit-Based Incentive Data Dissemination in Mobile Social Networks,"IIKI 2013

[44] Y. Zhu, B. Xu, X. Shi, and Y. Wang, "A survey of social-based routing in delay tolerant networks: positive and negative social effects,"IEEE Communications Surveys and Tutorials.

[45] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks,"in MobiCom '00: Proc. of the 6th annual international conference on Mobile computing and networking. New York, NY, USA: ACM, 2000.

[46] S. Buchegger and J.-Y. L. Boudec, "Performance analysis of the CONFIDANT protocol: Cooperation Of Nodes Fairness In Dynamic Ad-hoc NeTworks,"in MobiHoc 02: Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing, June 2002.

[47] S. Wang, M Liu, X. Cheng, Z. Li, J. Huang and B. Chen, "Opportunistic Routing in Intermittently Connected Mobile P2P Networks, "IEEE Journal on Selected Areas in Communications (JSAC). Accepted.

[48] L. Ma, AY. Teymorian and X. Cheng, "A hybrid rogue access point protection framework for commodity Wi-Fi networks, "INFOCOM, 2008, pp. 1220-1228.

[49] S. Ji, R. Beyah, and Z.Cai, "Minimum-latency broadcast scheduling for cognitive radio networks, "SECON, 2013, pp. 389-397.

[50] S. Cheng, J. Li and Z. Cai. "O(ϵ)-Approximation to Physical World by Sensor Networks, "INFOCOM, 2013, pp. 3084-3092.

[51] J. Li, S. Cheng, H. Gao and Z. Cai. "Approximate Physical World Reconstruction Algorithms in Sensor Networks, "IEEE Transactions on Parallel and Distributed Systems.

[52] C Ai, L Guo, Z Cai and Y Li. "Processing area queries in wireless sensor networks, "Mobile Ad-hoc and Sensor Networks, 2009, pp 1-8.

[53] Y. Sun, R. Bie, X. Yu, S. Wang. "Semantic Link Networks: Theory, Applications, and Future Trends, "Journal of Internet Technology, 2013, 13 (3):365-378.

[54] Z Cai, G. Lin and G. Xue. "Improved approximation algorithms for the capacitated multicast routing problem, "COCOON, 2005, pp 136-145.

[55] P. Michiardi and R. Molva, "Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks,"in Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security. Kluwer, B.V., 2002.

[56] Q. He, D. Wu, and P. Khosla, "SORI: A secure and objective reputationbased incentive scheme for ad hoc networks,"in Proc. WCNC, Atlanta, GA, Mar. 2004, pp. 825 - 830.

[57] K. Balakrishnan, J. Deng, and V. Varshney, "Twoack: preventing selfishness in mobile ad hoc networks,"in Wir. Comm. and Net. Conf., 2005 IEEE, vol. 4, March 2005.

[58] Y. Zhang and Y. Fang, "A fine-grained reputation system for reliable service selection in peer-to-peer networks,"IEEE Trans. Parallel Distrib. Syst., vol. 18, no. 8, pp. 1134 - 1145, Aug. 2007.

[59] U. Shevade, H. Song, L. Qiu, and Y. Zhang, "Incentive-aware Routing in DTNs,"in Proc. of ICNP, pp. 238 - 247, October 2008.

[60] L. Buttyan, L. Dora, M. Felegyhazi, and I. Vajda, "Barter trade improves message delivery in opportunistic networks,"Ad Hoc Networks, vol. 8, no. 1, pp. 1 - 14, 2010.

[61] A. Mei and J. Stefa, "Give2Get: Forwarding in Social Mobile Wireless Networks of Selfish Individuals,"in Proc. of ICDCS, pp. 488 - 497, 2010.

[62] L. Buttyan and J. P. Hubaux, "Enforcing Service Availability in Mobile Ad-hoc WANs,"in Proc. of MoBiHoc, pp. 87 - 96, 2000.

[63] S. Zhong, J. Chen, and Y. R. Yang, "Sprite, A Simple, Cheat-proof, Credit-based System For Mobile Ad-hoc Networks,"in Proc. of INFOCOM, pp. 1987 - 1997, 2003.

[64] N. B. Salem, L. Buttyan, J. P. Hubaux, and M. Jakobsson, "A Charging and Rewarding Scheme for Packet Forwarding in Multi-hop Cellular Networks,"in Proc. of MoBiHoc, pp. 13 - 24, 2003.

[65] M. Jakobsson, J. P. Hubaux, and L. Buttyan, "A Micropayment Scheme Encouraging Collaboration in Multi-hop Cellular Networks,"in Proc. of Financial Cryptography, pp. 15 - 33, 2003.

[66] B. Chen and M.C.Chan, "MobiCent: a Credit-Based Incentive System for Disruption Tolerant Network,"in Proc. of INFOCOM, pp. 1 - 9, 2010

[67] Ting Ning, Zhipeng Yang, Xiaojuan Xie, Wu, H. "Incentive-Aware Data Dissemination in Delay-Tolerant Mobile Networks,"SECON, 2011

[68] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in Delay Tolerant Networks: A Social Network Perspective,"in ACM MobiHoc, 2009.

[69] Anna-Kaisa Pietilainen and Christophe Diot, "CRAWDAD data set thlab/sigcomm2009 (v. 2012-07-15), Downloaded from http://crawdad.cs.dartmouth.edu/thlab/sigcomm2009,"July, 2012.

[70] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket Switched Networks and Human Mobility in Conference Environments,"in ACM SIGCOMM workshop on Delay-tolerant networking (WDTN), 2005.

[71] J. Burgess, B. N. Levine, R. Mahajan, J. Zahorjan, A. Balasubramanian, A. Venkataramani, Y. Zhou, B. Croft, N. Banerjee, M. Corner, and D. Towsley, "CRAWDAD data set umass/diesel (v. 2008-09-14)."Downloaded from http://crawdad.cs.dartmouth.edu/umass/diesel, Sept. 2008.

[72] Vikram Srinivasan, Mehul Motani, Wei Tsang Ooi, "Analysis and implications of student contact patterns derived from campus schedules,"in ACM SIGCOMM workshop on Delay-tolerant networking, proceedings of the 12th annual international conference on Mobile computing and networking, pages 86-97 (Mobicom06), 2006.

[73] Ting Ning, Zhipeng Yang, Hongyi Wu and Zhu Han, "Self-Interest-Driven incentives for ad dissemination in autonomous mobile social networks, "in INFOCOM, 2013 Proceedings IEEE.

[74] Y. Zhu, B. Xu, X. Shi, and Y. Wang, "A survey of social-based routing in delay tolerant networks: positive and negative social effects,"IEEE Communications Surveys and Tutorials.

[75] A. Vahdat and D. Becker, "Epidemic Routing for Partially Connected Ad Hoc Networks,"Tech. Rep. CS-200006, Duke University, 2000.

[76] W. Zhao, M. Ammar, and E. Zegura, "Multicasting in Delay Tolerant Networks: Semantic Models and Routing Algorithms, "sigcomm workshop, 2005.

[77] U. Lee, S. Y. Oh, K.-W. Lee, and M. Gerla, "RelayCast: Scalable Multicast Routing in Delay Tolerant Networks, "ICNP, 2008

[78] T. Ning, Z. Yang, X. Xie, and H. Wu, "Incentive-Aware Data Dissemination in Delay-Tolerant Mobile Networks,"SECON, 2011

[79] G. Liu, S. Ji and Z. Cai,"Strengthen nodal cooperation for data dissemination in mobile social networks,"PUC, 2014

[80] C. Boldrini, M. Conti, and A. Passarella, "Contentplace: Social-aware data dissemination in opportunistic networks,"in Proc. ACM Int. Symp. Model. Anal. Simul. Wireless Mobile Syst., pp. 203-210. 2008.

[81] E. Yoneki, P. Hui, S. Chan, and J. Crowcroft, "Contentplace: A socio-aware overlay for publish/subscribe communication in delay tolerant networks,"in Proc. ACM Int. Symp. Model. Anal. Simul. Wireless Mobile Syst., pp. 225-234. 2007.

[82] K. Kwong, A. Chaintreau, and R. Guerin, "Quantifying content consistency improvements through opportunistic contacts,"in Proc. 4th ACM Workshop Challenged Netw., 2009,

[83] R. Cabaniss, S. Madria, G. Rush, A. Trotta, and S. S. Vulli, "Dynamic social grouping based routing in a mobile ad-hoc network, "in Proc. 11th Int. Conf. Mobile Data Manage., pp. 295-296, 2010.

[84] S. Ioannidis and A. Chaintreau, "On the strength of weak ties in mobile social networks, "in Proc. ACM EuroSys. Workshop Social Netw. Syst., pp. 19-25, 2009.

[85] A. Miklas, K. Gollu, K. Chan, S. Saroiu, K. Gummadi, and E. de Lara, "Exploiting social interactions in mobile systems, "in Proc. Ubiquitous Comput., pp. 409-428, 2007.

[86] K. Kawarabayashi, F. Nazir, and H. Prendinger, "Message duplication reduction in dense mobile social networks,"infocom, 2010.

[87] D. Niyato, P. Wang, E. Hossain, and Y. Li, "Optimal content transmission policy in publish-subscribe mobile social networks,"globecom, 2010.

[88] D. Niyato, Z. Han, W. Saad, and A. Hjorungnes, "A controlled coalitional game for wireless connection sharing and bandwidth allocation in mobile social networks,"globecom, 2010.

[89] A. Beach, M. Gartrell, X. Xing, R. Han, Q. Lv, S. Mishra, and K. Seada, "Fusing mobile, sensor, and social data to fully enable context-aware computing,"in Proc. 11th Workshop Mobile Comput. Syst. Appl., 2010.

[90] J. Rana, J. Kristiansson, J. Hallberg, and K. Synnes, "An architecture for mobile social networking applications,"in Proc. Int. Conf. Comput. Intell. Commun. Syst. pp. 241-246, 2009.

[91] J. An, Y. Ko, and D. Lee, "A social relation aware routing protocol for mobile ad hoc networks,"in Proc. IEEE Int. Conf. Pervasive Comput. Commun., pp. 1 - 6, 2009,

[92] E. C.-H. Ngai, M. B. Srivastava and J. Liu, "Context-Aware Sensor Data Dissemination for Mobile Users in Remote Areas,"INFOCOM, pp. 2711 - 2715, 2012.

[93] Q. Yuan, I. Cardei and J. Wu, "An Efficient Prediction-Based Routing in Disruption-Tolerant Networks,"Parallel and Distributed Systems, pp. 19 - 31, 2012.

[94] M. E. J. Newman, "The structure and function of complex networks,"in SIAM REVIEW, vol. no. 45, pp. 167 - 256, 2003.