Incentive-Driven and Freshness-Aware Content Dissemination in Selfish Opportunistic Mobile Networks

Huan Zhou*†, Jie Wu†, Hongyang Zhao*, Shaojie Tang†, Canfeng Chen‡, and Jiming Chen*
*State Key Lab. of Industrial Control Technology, Zhejiang University, China.
†Department of Computer and Information Sciences, Temple University, USA.
‡Nokia Research Center, Beijing, China.

***Abstract*—Recently, the content-based publish/subscribe (pub/sub) paradigm is gaining popularity in opportunistic mobile networks (OppNets) for its flexibility and adaptability. Since nodes in OppNets is controlled by humans, they often behave selfishly with an aim to maximize their own revenues without considering the performance of others. Therefore, stimulating nodes in OppNets to collect, store, and share content efficiently is one of the key challenges under this scheme. Meanwhile, guaranteeing the freshness of content is also a big problem for content dissemination in OppNets. In this paper, in order to solve these problems, we propose an incentive-driven and freshness-aware pub/sub content dissemination scheme, called *ConDis* (Content Dissemination), for selfish OppNets. In *ConDis*, the Tit-For-Tat (TFT) scheme is employed to deal with selfish behaviors of nodes in OppNets. *ConDis* also implements a novel content exchange protocol when nodes are in contact. Specifically, during each contact, the exchange order is determined by the content utility, which is calculated by the direct subscribed value and the indirect subscribed value, and the objective of nodes is to maximize the utility of the content inventory stored in their buffer. Extensive realistic trace-driven simulation results show that *ConDis* is superior to other existing schemes in terms of total freshness value, total delivered contents, and total transmission cost.**

***Index Terms*—Publish/subscribe; content dissemination; opportunistic mobile networks; selfish behavior; freshness.**

I. Introduction

The content-based publish/subscribe (pub/sub) paradigm is a promising technology offering content-based services for nodes in opportunistic mobile networks (OppNets), which consist of a diversity of portable devices with the capability of ad hoc wireless communication, e.g., smart phones, PDAs, and laptops [1], [2], [3], [4], [5]. Specifically, end-to-end communication paths are hard to guarantee in OppNets, due to the time-varying network topology in OppNets. Thanks to the decoupling of the binding relationship between source and destination pairs, the pub/sub scheme has high flexibility and adaptability when dealing with highly dynamic network topologies, which bring a tremendous advantage for content dissemination in OppNets. Therefore, there is a great demand for studying such a scheme in OppNets.

Contents in the pub/sub scheme are organized into a set of channels, while a channel is a set of labels describing a type of content, and the goal is to deliver contents from publishers to subscribers [6], [7], [8]. The subscribers, namely the content consumers, express their interest without knowledge of the content generators' specific ID(s), and their interest is always stable in a long period of time; the publishers generate contents to the network without specifying the destination ID(s). Fig. 1 provides an example of pub/sub content dissemination in OppNets. Contents are organized into a set of channels, such as traffic news, sports news, weather forecast, and so on, and nodes can obtain subscribed contents through their opportunistic contacts with each other. For example, node 1, interested in job advertisement and financial news publishes "traffic jam in Temple University" to the network. Since node 2 subscribed to the traffic news channel is in contact with node 1, and because "traffic jam in Temple University" belongs to the traffic news channel, node 2 can obtain this content from node 1.

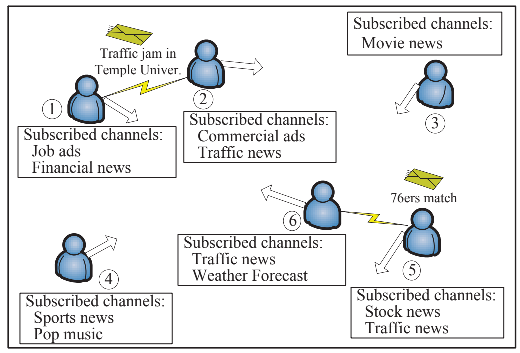


Fig. 1. An Example of Pub/sub Content Dissemination in OppNets.

The above example only illustrates the situation that nodes in the network can obtain subscribed contents from each other without giving anything back to the counterpart, which means they are cooperative. In this paper, we consider that nodes in the network are to be inherently selfish rather than cooperative. This is because nodes in OppNets are controlled by rational humans, who by nature often exhibit selfish behavior in reality [9]. If a node is selfish, it only aims to maximize its own revenues, and will not be willing to share its resources (e.g., memory space, transmission bandwidth, energy) to provide network services to others. Therefore, appropriate incentive schemes in OppNets need to be designed, in order to stimulate cooperation between nodes.

Incentive schemes have been extensively studied in the

Internet, mobile ad hoc networks (MANETs), and peer-to-peer (P2P) networks [10], [11]. Most existing works addressing selfishness can be classified into three categories: reputation-based [10], credit-based [11], and Tit-For-Tat (TFT)-based schemes [12], [13], [14]. Note that the TFT-based scheme does not require the detection of trusted nodes, secure hardware, or a centralized credit center; it only requires the principle of equal amounts of service, which is easy to achieve in intermittently-connected OppNets. Therefore, in this paper, we propose an incentive-driven and freshness-aware pub/sub content dissemination scheme, *ConDis* (Content Dissemination), for selfish OppNets, in which TFT is chosen as the incentive scheme to deal with selfish behaviors of nodes. It is worth noticing that *ConDis* also takes the freshness of content into consideration, this is because the satisfaction of a published content for a certain subscribed node is different at different times. For example, if node 2 obtains "traffic jam in Temple University" when it is published by node 2, then the satisfaction will be very high. However, if node 1 obtains it when it is going to expire, the satisfaction will be very low. This is because fresh contents are more useful for nodes.

Under the TFT scheme, any pair of nodes must offer each other equal amount of contents while they are trading with each other. Therefore, in order to trade with others, nodes have to use their storage to store content for others [9], [15]. Since the storage space of nodes is limited, and the importance of content is different for different nodes, nodes in the network have to choose some contents which are useful for them to store. An interesting optimization problem then arises: in order to get as many subscribed contents as possible, and guarantee the freshness of these contents, how should nodes act in the presence of TFT when the storage space is limited?

To answer this question, our proposed scheme, *ConDis*, introduces a novel content exchange protocol when nodes are in contact. Specifically, during each contact, the exchange order is decided by the content utility, which represents the usefulness of a content for a certain node. Intuitively, the utility of a certain content for a certain node should depend on two kinds of the current node's one-hop neighbors. The first kind are the nodes which are interested in this content, yet have not obtained this content; the second kind are the nodes which are not interested in this content, and have not obtained this content. From the perspective of real life trading, the first kind of nodes will absolutely choose to trade with the current node, because they are interested in this content and have not obtained it. The second kind of nodes may also choose to trade with the current node if many of their one-hop neighbors are interested in this content. Here, we denote the value contributed by the first kind as the direct subscribed value, and the value contributed by the second kind as the indirect subscribed value. Therefore, in the definition of content utility, we only need to take these two kinds of nodes into consideration, and the objective of nodes in the network is to maximize the utility of the content inventory stored in their buffer.

To summarize, the novelty and contributions of this paper are as follows:

1) We propose *ConDis*, an incentive-driven and freshness-aware pub/sub content dissemination scheme for selfish OppNets, in which TFT is employed as the incentive scheme to deal with selfish behaviors of nodes in OppNets.
2) In *ConDis*, a novel content exchange protocol is proposed when nodes are in contact. Specifically, during each contact, the exchange order is decided by the content utility, which includes the direct subscribed value and the indirect subscribed value, and the objective of nodes in the network is to maximize the utility of the content inventory stored in their buffer.
3) Extensive realistic trace-driven simulations are conducted to evaluate the performance of our proposed scheme, *ConDis*. The simulation result shows that *ConDis* outperforms other existing schemes.

The remainder of this paper is organized as follows. Section II summarizes the related work. Section III introduces the node contact model, channel and content model, and assumptions. System architecture of *ConDis* (Section IV-A), details about how to compute the content utility in *ConDis* (Section IV-B), and the content exchange protocol (Section IV-C) are introduced in Section IV. Section V evaluates the performance of our proposed scheme, *ConDis*, and other existing schemes through extensive realistic trace-driven simulations. Section VI concludes the paper.

II. Related Work

In this section, we first introduce the related work about content dissemination in OppNets, and then introduce the related work about the incentive schemes in OppNets.

A. Content Dissemination in OppNets

Early research on content dissemination in OppNets mostly relies on existing infrastructure. TACO-DTN [16] is a related solution exploiting a hybrid architecture composed of a fixed backbone and mobile nodes. Specifically, TACO-DTN introduces a concept of temporal interest expressed using profiles with corresponding temporal utility for data forwarding and buffer management. Peoplenet [17] is a hybrid system that first publishes and matches information queries over infrastructure, and then uses opportunistic contacts among mobile nodes to forward them further.

Later studies on content dissemination among mobile nodes without the help of infrastructure are closely related to the pub/sub paradigm. Research on the pub/sub paradigm in OppNets is initiated by the PodNet project [6], [18], which proposes a Podcasting architecture for OppNets. In the first version of PodNet [6], nodes only retrieve contents of channels they subscribe to. To improve the overall performance, another version was proposed in [18], in which some strategies for nodes are designed to also store contents associated to other channels. ContentPlace [15] aims to improve the performance of Podcasting using explicit knowledge of social relationships among nodes. SocialCast [19] employs social links of participants as well, which investigates the "homophily" phenomenon [20], and assumes that users with common interests have more frequent contacts with each other. Furthermore, since the cached data may be refreshed periodically and is subject to expiration, a novel scheme was proposed in [21] to efficiently maintain freshness of the cached data.

Different from existing studies on content dissemination in OppNets, in our proposed scheme, *ConDis*, we consider nodes in the network to be inherently selfish rather than cooperative.

B. Incentive Schemes in OppNets

Recently, there are several incentive schemes proposed for OppNets. In [22], an incentive-aware routing protocol for OppNets was proposed to adaptively optimize individual performance while conforming to TFT constraints. In [23], a credit-based incentive scheme was proposed to deal with selfishness in OppNets, which incorporates the incentive scheme into content dissemination in selfish OppNets with multiple interest types. An incentive content cooperation scheme was proposed in [24], to encourage node participation in OppNets; however, this work does not take the resource constraints like buffers into consideration. In [9], a utility-driven trading system, called MobiTrade, was proposed to optimize the content sharing strategy in OppNets, and derived an optimal policy to split the buffer of a node into zones allocated to each channel. This work chooses TFT as the incentive scheme to deal with selfish behaviors of nodes in OppNets, however, it does not take into account the TTL of contents. Indeed, TTL has a significant impact on the definition of content utility. If the TTL of a certain content is going to expire, the contact probability to meet those nodes interested in this content will be low; hence the utility of this content will also be low. Therefore, an incentive-based pub/sub content dissemination scheme, called ConSub, was proposed in [8], in which the content utility is determined by the contact probability and cooperation level between the current node and its one-hop neighbors subscribing to the associated channel.

Different from the above studies, our proposed scheme, *ConDis*, takes both the direct subscribed value and the indirect subscribed value into consideration from the perspective of real life trading. Furthermore, *ConDis* also takes into account the freshness of contents.

III. Models and Assumptions

This section gives a brief introduction to the node contact model, the channel and content model, and several assumptions in our proposed scheme.

A. Node Contact Model

Recently, some studies [25], [26] in OppNets found that the pair-wise inter-contact time in realistic traces follows an exponential distribution. Specifically, authors in [25] conduct the χ^2 hypothesis test on each contacted node pair in the *Infocom 06* [27] and *MIT Reality* [28] traces, to test whether "the pair-wise inter-contact time of nodes follows an exponential distribution." Their results show that when a large enough number of test intervals ($\geq$ 10) is used, over 85% of the contacted node pairs in the above traces pass the test. Based on these results, in this paper we also assume that the distribution of the pair-wise inter-contact time in OppNets follows the exponential distribution. Hence, the contact between nodes i and j becomes a homogeneous Poisson process with a contact frequency of λ_{ij}.

B. Channel and Content Model

Nodes in the network need to express their interests towards different kinds of contents in a certain way, and accordingly subscribe to those channels. They may subscribe to one or several channels. Here, we identify contents of those channels based on the matching between the key words of subscription and the description of the channel. The matching model for these key words is suitable to the pub/sub scheme discussed in this paper, because nodes in the network may have no idea what they are looking for, but just have interest in the contents which match some key words. For simplicity, we assume that the total number of channels in the network is C, and each content subscribed by nodes in the network can only be described by one channel. Furthermore, we assume that each node decides its subscribed channel (s) at the beginning of the network, and will not change it (them) in a long period.

In the pub/sub scheme, each node can be the subscriber or the publisher, or both the subscriber and the publisher. Each published content includes (d, c, T_d, T), while d is the sequence number of the content, c is the sequence number of the associated channel, T_d is the generation time of the content, and T is the Time-To-Live (TTL) of the content, which includes several units of time. Each published content also includes some key words to describe it. After publishing a content into the network, each node will store the published content to its buffer, so as to trade with others.

C. Assumptions

Without loss of generality, we assume that buffer spaces of nodes in the network are all B. Furthermore, contents generated in the network have the same volume capacity; thus when nodes exchange contents with each other, we can count the total content volume by counting the number of contents. Finally, during each contact, we assume that the contact duration between pair-wise nodes is long enough to complete the content exchange.

IV. Proposed Scheme *ConDis*

In this section, we first introduce the architecture of our proposed scheme, *ConDis*, and then provide the detailed computing method of the content utility, which includes the contact probability and the expected delay. Finally, we introduce the content exchange protocol when nodes are in contact.

A. The Architecture of ConDis

In this part, we give the architecture of our proposed scheme, *ConDis*. It includes the following four parts:

1) Subscribed Channel Manager: Each node's subscribed channel manager keeps the channel information subscribed by its one-hop neighbors, and merges it with its own subscribed channels to form a list of subscribed channels. Furthermore, each node needs to keep the pair-wise inter-contact time with all its one-hop neighbors.

2) Buffer State Collector: Each node's buffer state collector collects the content information stored at its one-hop neighbors and merges it with its own stored content information to form a list of its buffer state. Specifically, we assume that each content in the buffer is associated with a metadata, which includes its associated node, sequence number (d), associated channel (i.e., c), generation time (T_d), and Time-To-Live (TTL). Let $\mathcal{W}_i$ denote the set of metadata of contents stored at node i and its one-hop neighbors, and node i will update $\mathcal{W}_i$ when contents stored in its buffer change or it encounters new nodes.

3) Content Utility Estimator: Each content has an initial freshness value V when it is published by a certain node. Without loss of generality, we assume that the freshness value of a certain content for a certain subscribed node will decrease as time passes, while the corresponding freshness value for a certain unsubscribed node will be 0. Furthermore, when the TTL of a certain content is expired, the corresponding freshness value for a certain subscribed node will be 0. Therefore, the freshness value of a certain content d for a subscribed node i can be expressed as:

$$v_i(d) = \frac{R_d V}{T}, \tag{1}$$

where R_d is the remaining valid time of content d, which can be calculated by using the current time, the generation time T_d and TTL. Here, T is the TTL of content d.

Under the TFT scheme, in order to trade with others, nodes in the network need to store some contents in their buffer. However, since the importance of a certain content is different for different nodes, and the size of their buffer is limited, nodes have to choose some contents which are useful for them to store. Here, the usefulness of a certain content for a certain node depends on its neighbors. Specifically, taking the content d in channel c for node i as an example. As shown in Fig. 2, we denote $\mathcal{N}_i^1$ and $\mathcal{N}_i^2$ as node i's one-hop and two-hop neighbors, respectively, and divide the set of node i's one-hop neighbors into three kinds: the first kind are the nodes which subscribe to channel c and have not obtained content d, denoted as $\mathcal{N}_i^1(c)$; the second kind are the nodes which do not subscribe to channel c and have not obtained content d, denoted as $\mathcal{M}_i^1(c)$; and the third kind are the nodes which have already obtained content d, denoted as $\mathcal{L}_i^1(d)$. From the perspective of real-life trading, the first kind of nodes will absolutely choose to trade with node i, because they subscribe to channel c and have not obtained content d. Therefore, in the definition of content utility, we denote the value contributed by this part as the direct subscribed value. The second kind of nodes may also choose to trade with node i if many of their one-hop neighbors subscribe channel c. This is because these nodes can use content d to trade with their one-hop neighbors. Similarly, we denote the value contributed by this part as the indirect subscribed value. The third kind of nodes will absolutely not choose to trade with node i, because they have already obtained content d. As a result, in the utility definition of content d for node i, we only need to take into account the first kind and the second kind of node i's one-hop neighbors. Hence, the utility of a content d in channel c for

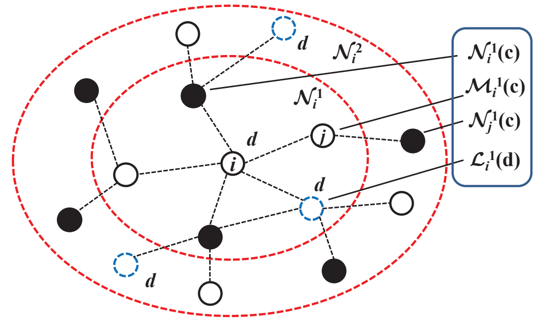


Fig. 2. Illustrating the utility definition of a certain content d in channel c for node i.

node i is defined as:

$$U_i(d) = wU_{di}(d) + (1-w)U_{indi}(d), \tag{2}$$

where $U_{di}(d)$ is the direct subscribed value, and $U_{indi}(d)$ is the indirect subscribed value; w in the range of $[0,1]$; here, w and $1-w$ represent the weight of the direct subscribed value, and the indirect subscribed value, respectively.

As shown in Fig. 2, if node $j \in \mathcal{N}_i^1(c)$ and j's other one-hop neighbors have already obtained a copy of content d, j can also obtain this content from them, which will decrease the direct subscribed value of content d for node i. Therefore, in the calculation of $U_{di}(d)$, we also take this situation into consideration. Moreover, in order to guarantee the freshness of content d for those subscribed nodes, we express the direct subscribed value $U_{di}(d)$ as:

$$U_{di}(d) = \sum_{j \in \mathcal{N}_i^1(c)} \frac{(R_d - ED_{ij})V}{T} \prod_{k \in \mathcal{L}_j^1(d)} [1 - Pr_{jk}(R_d)], \tag{3}$$

where $\mathcal{N}_i^1(c)$ is the set of node i's one-hop neighbors which subscribe to channel c and have not obtained content d, and ED_{ij} is the expected delay for transmitting content d from node i to node j; $\mathcal{L}_j^1(d)$ is the set of node j's one-hop neighbors which have already obtained content d; $Pr_{jk}(R_d)$ is the contact probability between nodes j and k in the remaining valid time R_d of content d.

Similarly, as shown in Fig. 2, taking node i's two-hop neighbors which subscribe to channel c into consideration, the indirect subscribed value $U_{indi}(d)$ can be expressed as:

$$U_{indi}(d) = \sum_{j \in \mathcal{M}_i^1(c)} \sum_{k \in \mathcal{N}_j^1(c)} \frac{(R_d - ED_{ik}^j)V}{T}, \tag{4}$$

where $\mathcal{M}_i^1(c)$ is the set of node i's one-hop neighbors which do not subscribe to channel c; $\mathcal{N}_j^1(c)$ is the set of node j's one-hop neighbors which subscribe to channel c and have not obtained content d; ED_{ik}^j is the expected delay for transmitting the content d from node i to k through j. It is worth noticing that, in the calculation of the indirect subscribed value, we do not take node k's other one-hop neighbors into consideration. This is because it is hard for node i to obtain the buffer state of node k's one-hop neighbors, or if obtained, the buffer state may be inaccurate.

When we substitute Eq. (3) together with Eq. (4) into Eq. (2), we can compute the content utility $U_i(d)$ of content d in channel c for node i as follows:

$$U_i(d) \; = \; w \sum_{j \in \mathcal{N}_i^1(c)} \frac{(R_d - ED_{ij})V}{T} \prod_{k \in \mathcal{L}_j^1(d)} [1 - Pr_{jk}(R_d)] + (1-w) \sum_{j \in \mathcal{M}_i^1(c)} \sum_{k \in \mathcal{N}_j^1(c)} \frac{(R_d - ED_{ik}^j)V}{T}. \tag{5}$$

In order to obtain the content utility $U_i(d)$ of content d in channel c for node i, we have to compute the contact probability $Pr_{jk}(R_d)$, the expected delay ED_{ij}, and ED_{ik}^j first. Therefore, in the next part, we will introduce the detailed computing method of $Pr_{jk}(R_d)$, ED_{ij} and ED_{ik}^j.

4) Buffer Manager: Since the importance of a certain content is different for different nodes, and the size of the buffer is limited, nodes in the network have to choose some useful contents to store. Therefore, in order to get as many subscribed contents as possible, and guarantee the freshness of these subscribed contents, the objective of nodes in the network is to maximize the expected content utility in their buffer, which can be expressed as:

$$Max \;\; U_i \; = \; \sum_{c=1}^{C} \Big(\sum_{d \in \theta(c)} U_i(d) - \sum_{d \in \phi(c)} U_i(d) \Big), \tag{6}$$

where U_i is the utility function of node i; C is the total number of channels; and $\theta(c)$ and $\phi(c)$ are the set of contents associated to channel c in their buffer after and before exchange, respectively.

The cache management of a node is mainly based on the content utility. When a new content is sent to the current node, it places the content into the buffer corresponding to its content utility. If the buffer size of the current node is full, the content with higher utility can seize the cache position that is occupied by the content with lower utility. Moreover, the expired contents will be deleted directly from the buffer, even if there is unoccupied space.

B. Computing Content Utility

In this part, we introduce how to compute the contact probability and the expected delay in the definition of content utility.

1) Contact Probability Prediction: As introduced in Section III, in this paper we also assume that the pair-wise inter-contact time in realistic traces follows an exponential distribution. The contact frequency λ_{ij} between nodes i and j is indicated by the contact rate, and can be computed by the following time average method:

$$\lambda_{ij} \; = \; \frac{n}{\Sigma_{l=1}^{n} T_{ij}^{l}}, \tag{7}$$

where $T_{ij}^1, \; T_{ij}^2, \; \ldots, \; T_{ij}^n$ are inter-contact time samples between nodes i and j.

Thus, the Probability Distribution Function (PDF) of the inter-contact time X_{ij} between nodes i and j can be expressed as:

$$f_{X_{ij}}(x) \; = \; \lambda_{ij} e^{-\lambda_{ij} x}. \tag{8}$$

Then, the contact probability between nodes i and j within the remaining valid time R_d of content d can be expressed as:

$$\begin{aligned} Pr_{ij}(R_d) \; &= \; Pr(X_{ij} \le R_d) \; = \; \int_0^{R_d} f_{X_{ij}}(x) dx \\ &= \; 1 \; - \; e^{-\lambda_{ij} R_d}. \end{aligned} \tag{9}$$

2) Expected Delay Prediction: Based on Eq. (8), the expected delay ED_{ij} for transmitting a certain content from node i to j can be calculated as:

$$ED_{ij} = E[X_{ij}] = \int_0^{\infty} x f_{X_{ij}}(x) dx = \frac{1}{\lambda_{ij}}. \tag{10}$$

After obtaining the expected delay ED_{ij} for transmitting a certain content from node i to node j, next we calculate the expected delay for transmitting a certain content from node i to node k through node j. Note that the total time to transfer a certain content from i to k through j is $X_{ik}^j = X_{ij} + X_{jk}$, while X_{ij} is the time needed for transmitting the content from i to j, and X_{jk} is the time needed for transmitting the content from j to k. Since the pair-wise inter-contact time follows an exponential distribution, we obtain that X_{ij} follows the exponential distribution with the parameter λ_{ij}. According to the memoryless property of the exponential distribution, X_{jk} also follows the exponential distribution, but with the parameter λ_{jk}. Therefore, based on Eq. (8), the PDF $f_{X_{ik}^j}(t)$ can be calculated as:

$$\begin{aligned} f_{X_{ik}^j}(x) \; &= \; f_{X_{ij}}(x) \otimes f_{X_{jk}}(x) \\ &= \; \lambda_{ij}\lambda_{jk} \int_0^x e^{-(\lambda_{ij}-\lambda_{jk})t} e^{-\lambda_{jk} t} dt \\ &= \; \frac{\lambda_{ij}\lambda_{jk}(e^{-\lambda_{ij}x} - e^{-\lambda_{jk}x})}{\lambda_{jk} - \lambda_{ij}}, \end{aligned} \tag{11}$$

where $\otimes$ is the convolution operator; $f_{X_{ij}}(x)$ is the PDF of the inter-contact time X_{ij} between nodes i and j, and $f_{X_{jk}}(x)$ is the PDF of the inter-contact time X_{jk} between nodes j and k.

Therefore, the expected delay ED_{ik}^j for transmitting a certain content from i to k through j can be expressed as:

$$\begin{aligned} ED_{ik}^j \; &= \; E[X_{ik}^j] \; = \; \int_0^{\infty} x f_{X_{ik}^j}(x) dx \\ &= \; \int_0^{\infty} \frac{x\lambda_{ij}\lambda_{jk}(e^{-\lambda_{ij}x} - e^{-\lambda_{jk}x})}{\lambda_{jk} - \lambda_{ij}} dx \\ &= \; \frac{\lambda_{ij} + \lambda_{jk}}{\lambda_{ij}\lambda_{jk}}. \end{aligned} \tag{12}$$

When we substitute Eq. (9) and Eq. (10), together with Eq. (12), into Eq. (5), we can compute the content utility $U_i(d)$ of content d in channel c for node i as follows:

$$\begin{aligned} U_i(d) \; &= \; w \sum_{j \in \mathcal{N}_i^1(c)} \frac{(R_d - \frac{1}{\lambda_{ij}})V}{T} \prod_{k \in \mathcal{L}_j^1(d)} e^{-\lambda_{jk} R_d} \\ &+ (1-w) \sum_{j \in \mathcal{M}_i^1(c)} \sum_{k \in \mathcal{N}_j^1(c)} \frac{(R_d - \frac{\lambda_{ij}+\lambda_{jk}}{\lambda_{ij}\lambda_{jk}})V}{T}. \end{aligned} \tag{13}$$

C. The Content Exchange Protocol

Based on the above models, assumptions, and definitions, our proposed scheme, *ConDis*, is outlined below. Taking two nodes i and j as an example. When i meets j, node i needs to decide whether or not to exchange contents in its buffer with the latter. If j has some useful contents in its buffer for i, for example, some contents which are subscribed by i, or some contents which can increase the content utility in i's buffer, then i will choose to exchange contents with j. From the perspective of real life trading, node i will give priority to obtaining its subscribed contents with higher freshness value, and then aims to obtain contents which can increase its own expected content utility in its buffer. Then, our proposed scheme *ConDis* works as follows, in five steps:

1) When i meets j, node i first sends a control message to j, which includes the subscribed channel list (including that of itself and its neighbors), the contact frequency of its one-hop neighbors, and the set $\mathcal{W}_i$ (which includes the metadata of contents stored in its own buffer and its one-hop neighbors). Node j also sends a similar control message to i.
2) When i receives the control message from j, it first updates the past stored control messages with the new received control message from j. Then, it creates a set S_i to denote the set of contents that are stored at j from $\mathcal{W}_j$, and a set L_i to denote the set of contents that are available at j but not at i, i.e., $L_i = S_i - (S_i \cap S_j)$. Based on the stored control messages, node i calculates the freshness value of contents in L_i which meet its interest, and the content utility of all contents in L_i according to Eq. (13). Similarly, node j does so in a similar way.
3) Node i checks if there are any contents in L_i matching its interest. Let $L_i^{'}$ denote the set of such contents. Then node i adds them into the candidate request list R_i in the decreasing order of the freshness value with high priority. After determining the subscribed contents, i next determines the contents not matching its interest. Let $L_i^{''}$ denote the set of contents in L_i that does not include contents matching its interest. Node i then adds them into the candidate request list R_i in the decreasing order of the content utility with low priority. Accordingly, node j does so in a similar way, and obtains the candidate request list R_j.
4) After determining the candidate request list of each other, nodes i and j will trade with each other to obtain subscribed contents, and contents which can increase the total content utility in their buffer from each other. In the decreasing order of priority and content utility, nodes i and j will trade contents in the candidate request list of each other, one by one. After obtaining a new content, they will store it in their buffer according to its content utility. They will finish the trading until one side does not have contents, which can increase the total content utility in the buffer for the other side.
5) After finishing the trading with each other, nodes i will update L_i, and send a new control message to j. Node j does so in a similar way.

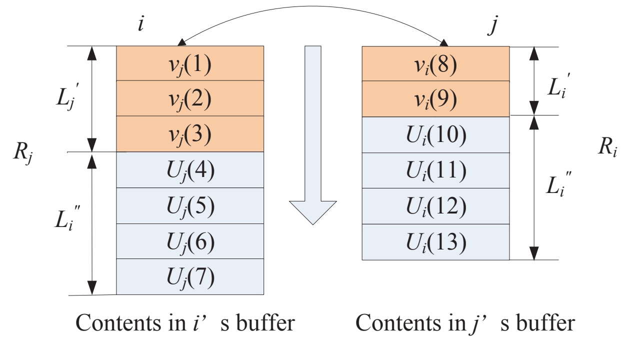


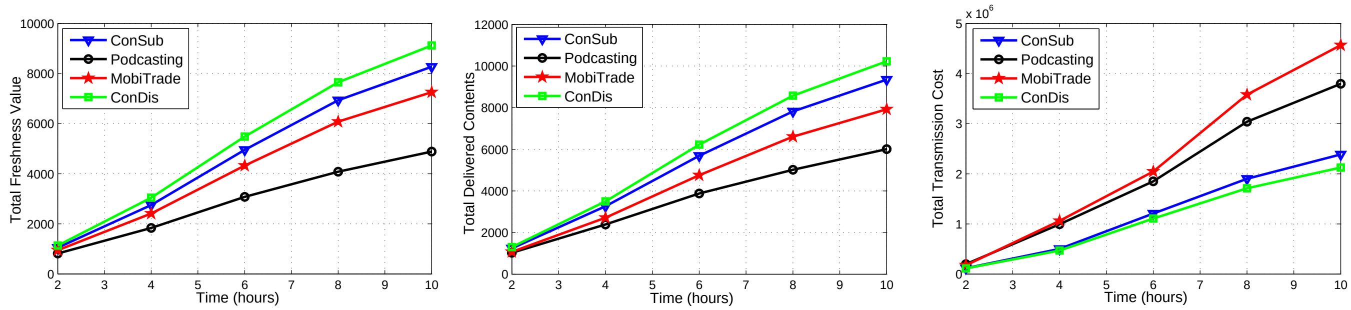
Fig. 3. The content exchange process between nodes i and j in *ConDis*.

Fig. 3 gives an example about the content exchange process between nodes i and j in *ConDis*. As introduced above, R_i and R_j represent the candidate request list of node i and node j, respectively. For simplicity, we use small numbers: 1, 2, ..., 13, to represent the sequence number of contents stored in nodes i and js' buffers. Here, $v_j(1)$ denotes the freshness value of content 1 for subscribed node j, and $v_j(1) \geq v_j(2) \geq v_j(3)$. Similarly, $v_i(8)$ denotes the freshness value of content 8 for subscribed node i, and $v_i(8) \geq v_i(9)$. $U_j(4)$ denotes the utility of content 4 for node j, and $U_j(4) \geq U_j(5) \geq U_j(6) \geq U_j(7)$. Similarly, $U_i(10)$ denotes the utility of content 10 for node i, and $U_i(10) \geq U_i(11) \geq U_i(12) \geq U_i(13)$. According to the content exchange protocol introduced above, nodes i and j start the trading from the top of each other's candidate request list, which means they will give priority to obtain subscribed contents which have higher freshness value. It is worth noticing that if $|L_i^{'}| = |L_j^{'}|$, nodes i and j will finish the trading of their subscribed contents. However, if $|L_i^{'}| \neq |L_j^{'}|$, in order to obtain other subscribed contents, one side which does not have further subscribed contents for the other side needs to provide contents, which can increase the total content utility in the buffer for the other side.

V. Performance Evaluation

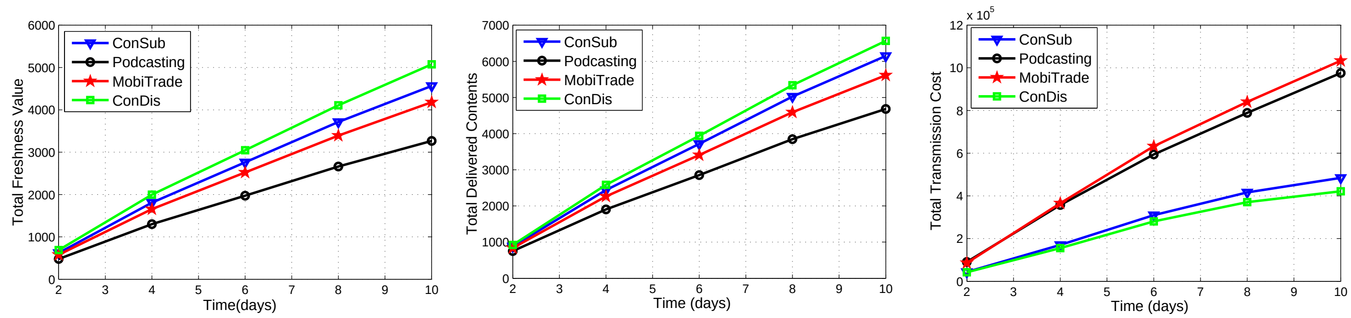
In this section, we evaluate the performance of our proposed scheme, *ConDis*, in selfish OppNets. Specifically, we compare it with the following three existing schemes:

1) **Podcasting** [18]: nodes receive all contents which their neighbors subscribe, and randomly discard contents when their buffer is full.
2) **MobiTrade** [9]: each node defines a buffer quota for each channel based on the past reward of the channel, and adaptively manages its buffer according to the buffer quota.
3) **ConSub** [8]: each node exchanges contents according to the content utility, which is determined by the contact probability and the cooperation level between the current node and its one-hop neighbors subscribing to the associated channel. During each contact, the objective of nodes in the network is to maximize the utility of the content inventory stored in their buffer.



(a) Total Freshness Value (b) Total Delivered Contents (c) Total Transmission Cost

Fig. 4. Performance comparison of *ConDis* with other existing schemes in the *Infocom 06* trace when w is 0.9



(a) Total Freshness Value (b) Total Delivered Contents (c) Total Transmission Cost

Fig. 5. Performance comparison of *ConDis* with other existing schemes in the *MIT Reality* trace when w is 0.9

TABLE I
BASIC STATISTICS OF THE TRACES

Trace	*Infocom 06*	*MIT Reality*
Device	iMote	Smart Phone
Network type	Bluetooth	Bluetooth
No. of internal contacts	182,951	114,046
Duration (days)	3	246
Granularity (seconds)	120	300
No. of devices	78	97
Contact frequency/pair/day	6.7	0.024

We use two experimental traces, *Infocom 06* [27] and *MIT Reality* [28], collected from realistic environments to evaluate the performance of the above schemes. Nodes in these two traces carry bluetooth-enabled portable devices, which record contacts by periodically detecting their nearby peers. The traces cover various types of corporate environments, and have various experimental durations. The detail of these two traces is summarized in Table I.

A. Simulation Setup

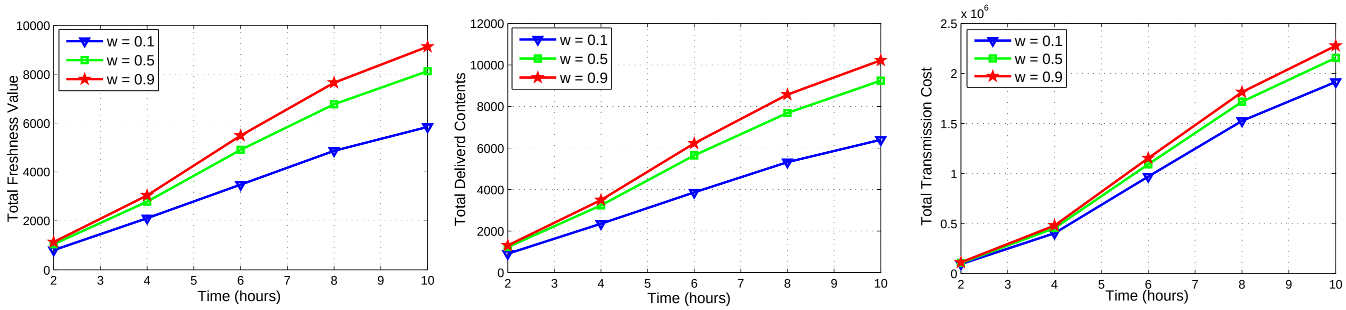
During the whole experiment, we consider that each node generates contents that match one of the channels, and the content generation rate follows a uniform distribution, while 1 content/hour means that nodes generate 1 content per hour. Since the contact frequency in the *Infocom 06* trace is much larger than that in the *MIT Reality* trace, we set the content generation rate as 1 content/hour, and the TTL as 2 hours in the *Infocom 06* trace; while in the *MIT Reality* trace, we set the content generation rate as 1 content/day, and the TTL as 1 day. Furthermore, there are 5 channels in the network, and each node only expresses interest, randomly, in one channel. The size of all contents equals to $40K$, each experiment has the same TTL, and, the buffer size of nodes in the network $B = 1000K$. Finally, the initial freshness value of all contents in the network $V = 1$, and when the TTL of a certain content is expired, the corresponding freshness value will be 0. In our simulation studies, we focus on the following three performance metrics for performance evaluation:

1) **Total freshness value**: the total freshness value of contents successfully delivered for channels subscribed by nodes in the network, which reflects the effectiveness of the scheme.
2) **Total delivered contents**: the total number of contents successfully delivered for channels subscribed by nodes in the network, which reflects the effectiveness of the scheme.
3) **Total transmission Cost**: the total number of contents exchanged by nodes in the network, which reflects the energy consumption of the scheme.

B. Performance Comparison

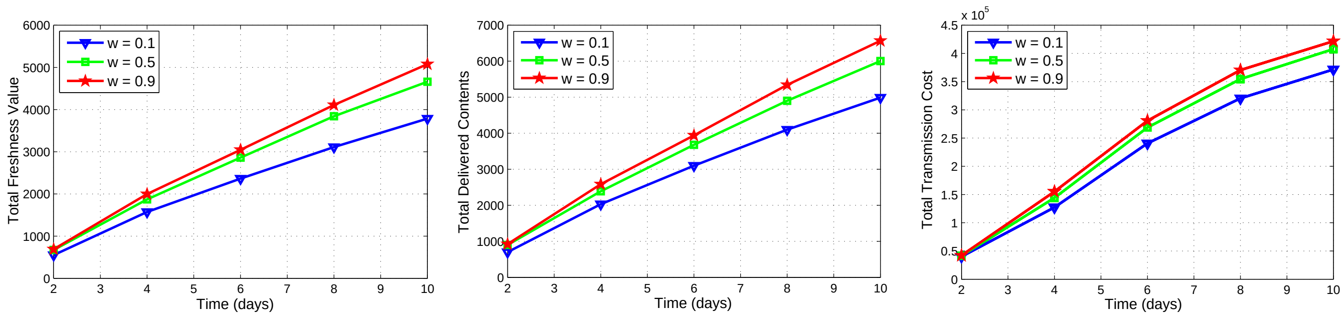
In this part, we compare the performance of our proposed scheme, *ConDis*, with three other existing schemes using the *Infocom 06* trace and the *MIT Reality* trace. Here, our aim is to evaluate the performance of our proposed scheme, *ConDis*, in different traces.

Fig. 4 shows the performance comparison of our proposed scheme, *ConDis*, with other three existing schemes in the



(a) Total Freshness Value (b) Total Delivered Contents (c) Total Transmission Cost

Fig. 6. Performance of *ConDis* with different w in the *Infocom 06* trace



(a) Total Freshness Value (b) Total Delivered Contents (c) Total Transmission Cost

Fig. 7. Performance of *ConDis* with different w in the *MIT Reality* trace

Infocom 06 trace when w is 0.9. It can be found that as the simulation time increases, our proposed scheme *ConDis* outperforms other three existing schemes in terms of total freshness value, total delivered contents, and total transmission cost in the *Infocom 06* trace. Moreover, the longer the simulation time is, the better our proposed scheme *ConDis* performs. This is because our proposed scheme, *ConDis*, takes both the direct subscribed value and the indirect subscribed value into consideration from the perspective of real life trading. Furthermore, *ConDis* also takes into account the freshness of contents, and designs a novel content exchange protocol in the content exchange process when nodes are in contact. It can be also found that Podcasting performs worst in the *Infocom 06* trace. The main reason is that nodes in Podcasting receive all contents to which their one-hop neighbors subscribe, and do not manage their buffer. Although Mobitrade and Consub take the buffer management into consideration, they both rely on the past transaction between nodes to define the utility of contents, which may be inaccurate for predicting the future trading value of contents. Moreover, they do not take the freshness of contents into consideration. Therefore, our proposed scheme, *ConDis*, can achieve a better performance than Podcasting, MobiTrade, and Consub in terms of total freshness value, total delivered contents, and total transmission cost in the *Infocom 06* trace.

Fig. 5 shows the performance comparison of our proposed scheme, *ConDis*, with other three existing schemes in the *MIT Reality* trace when w is 0.9. It can be found that, similar to the results in Fig. 4, *ConDis* also outperforms other schemes in terms of total freshness value, total delivered contents, and total transmission cost in the *MIT Reality* trace. Moreover, the longer the simulation time is, the better *ConDis* performs. Podcasting also performs worst in the *MIT Reality* trace, and Consub also performs better than Mobitrade in the *MIT Reality* trace. It is worth noticing that the simulation time in the *MIT Reality* trace is much longer than that in the the *Infocom 06* trace. This is because the experimental duration in the *MIT Reality* trace is much longer than that in the *Infocom 06* trace, and the contacts in the *MIT Reality* trace are much sparser than those in the *Infocom 06* trace.

To summarize, as the simulation time increases, our proposed scheme *ConDis* not only outperforms other schemes in terms of total freshness value, total delivered contents and total transmission cost in the *Infocom 06* trace, but also in the *MIT Reality* trace, which demonstrates the effectiveness of our proposed scheme. Moreover, the longer the simulation time is, the better our proposed scheme *ConDis* performs.

C. Impact of w

In this part, we evaluate the performance of our proposed scheme, *ConDis*, with different values of w (w = 0.1, 0.5 and 0.9) using the *Infocom 06* trace and the *MIT Reality* trace, aiming to check the impact of the changing metrics w on the performance of *ConDis* in different traces.

Figs. 6 and 7 show the performance of our proposed scheme, *ConDis*, with different values of w (w = 0.1, 0.5 and 0.9) in the *Infocom 06* trace and the *MIT Reality* trace, respectively. It can be found that, as w increases from 0.1 to 0.9, *ConDis* not only performs better in terms of total freshness value, total delivered contents, and total transmission cost in the *Infocom 06* trace, but also in the *MIT Reality* trace. The main reason is that w changes the balance between the direct subscribed value and the indirect subscribed value, which are determined by nodes' one-hop neighbors and two-hop neighbors, respectively. As w increases, nodes in the network will give priority to storing contents subscribed directly by their one-hop neighbors, which means nodes can obtain more contents matching their interest from the trading with their one-hop neighbors. Moreover, in order to calculate the indirect subscribed value of a certain one-hop neighbor, the current node needs to predict its two-hop neighbors' information based on its past stored control messages, which may be inaccurate. Therefore, our proposed scheme, *ConDis*, performs better when w is larger.

In summary, w has a significant impact on the performance of our proposed scheme, *ConDis*. Since it may be inaccurate for a certain node to predict its two-hop neighbors's information based on its past stored control messages, it is better for nodes in the network to store contents subscribed directly by their one-hop neighbors. Therefore, in order to increase the performance of *ConDis*, nodes in the network should give priority to storing contents subscribed directly by their one-hop neighbors.

VI. Conclusion

In this paper, we have investigated the pub/sub content dissemination in OppNets. Considering the selfish behavior of nodes and the freshness of contents, we propose an incentive-driven and freshness-aware pub/sub content dissemination scheme, called *ConDis*, for selfish OppNets. In *ConDis*, we choose the TFT scheme as the incentive scheme to deal with selfish behaviors of nodes in the network. Moreover, we also propose a novel content exchange protocol when nodes are in contact. Specifically, during each contact, the exchange order is decided by the content utility, and the objective of nodes in the network is to maximize the utility of the content inventory stored in their buffer. Extensive realistic trace-driven simulations are conducted to evaluate the performance of our proposed scheme, *ConDis*, and the results show that *ConDis* is superior to other existing schemes in terms of total freshness value, total delivered contents, and total transmission cost.

VII. Acknowledgement

This research was supported in part by NSF grants ECCS 1231461, ECCS 1128209, CNS 1138963, CNS 1065444, and CCF 1028167.

References

[1] J. Fan, J. Chen, Y. Du, W. Gao, J. Wu, and Y. Sun. Geo-community-based broadcasting for data dissemination in mobile social networks. *IEEE Transactions on Parallel and Distributed Systems*, 24(4):734–743, 2013.

[2] J. Fan, J. Chen, Y. Du, P. Wang, and Y. Sun. Delque: A socially-aware delegation query scheme in delay tolerant networks. *IEEE Transactions on Vehicular Technology*, 60(5):2181–2193, 2011.

[3] J. Wu and Y. Wang. Social feature-based multi-path routing in delay tolerant networks. In *Proceedings of IEEE INFOCOM*, 2012.

[4] H. Zhou, J. Chen, H. Zhao, W. Gao, and P. Cheng. On exploiting contact patterns for data forwarding in duty-cycle opportunistic mobile networks. *IEEE Transactions on Vehicular Technology, DOI:10.1109/TVT.2013.2267236*, 2013.

[5] H. Zhou, H. Zhao, and J. Chen. Energy saving and network connectivity tradeoff in opportunistic mobile networks. In *Proceedings of IEEE Globecom*, 2012.

[6] V. Lenders, G. Karlsson, and M. May. Wireless ad hoc podcasting. In *Proceedings of IEEE SECON*, 2007.

[7] F. Li and J. Wu. MOPS: Providing content-based service in disruption-tolerant networks. In *Proceedings of IEEE ICDCS*, 2009.

[8] H. Zhou, J. Chen, J. Fan, Y. Du, and S. K. Das. Consub: incentive-based content subscribing in selfish opportunistic mobile networks. *IEEE Journal on Selected Areas in Communications, DOI:10.1109/JSAC.2013.SUP.0513058*, 2013.

[9] A. Krifa, C. Barakat, and T. Spyropoulos. Mobitrade: trading content in disruption tolerant networks. In *Proceedings of ACM CHANTS*, pages 31–36, 2011.

[10] R. Mahajan, M. Rodrig, and D. Wetherall J. Zahorjan. Sustaining cooperation in multi-hop wireless networks. In *Proceedings of USENIX NSDI*, pages 231–244, 2005.

[11] S. Zhong, J. Chen, and Y. R. Yang. Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. In *Proceedings of IEEE INFOCOM*, volume 3, pages 1987–1997, 2003.

[12] F. Milan, J. J. Jaramillo, and R. Srikant. Achieving cooperation in multihop wireless networks of selfish nodes. In *Proceeding of the workshop on Game theory for communications and networks*, 2006.

[13] J. J. Jaramillo and R. Srikant. Darwin: distributed and adaptive reputation mechanism for wireless ad-hoc networks. In *Proceedings of ACM MobiCom*, pages 87–98, 2007.

[14] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. R. Rao. Cooperation in wireless ad hoc networks. In *Proceedings of IEEE INFOCOM*, volume 2, pages 808–817, 2003.

[15] C. Boldrini, M. Conti, and A. Passarella. Contentplace: social-aware data dissemination in opportunistic networks. In *Proceedings of ACM MSWiM*, pages 203–210, 2008.

[16] G. Sollazzo, M. Musolesi, and C. Mascolo. Taco-dtn: a time-aware content-based dissemination system for delay tolerant networks. In *Proceedings of ACM MobiOpp*, pages 83–90, 2007.

[17] M. Motani, V. Srinivasan, and P. S. Nuggehalli. Peoplenet: engineering a wireless virtual social network. In *Proceedings of ACM MobiCom*, pages 243–257, 2005.

[18] M. May, V. Lenders, G. Karlsson, and C. Wacha. Wireless opportunistic podcasting: implementation and design tradeoffs. In *Proceedings of ACM CHANTS*, pages 75–82, 2007.

[19] P. Costa, C. Mascolo, M. Musolesi, and G. P. Picco. Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 26(5):748–760, 2008.

[20] M. McPherson, L. Smith-Lovin, and J.M. Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27:415–444, 2001.

[21] W. Gao, G. Cao, M. Srivatsa, and A. Iyengar. Distributed maintenance of cache freshness in opportunistic mobile networks. In *Proceedings of IEEE ICDCS*, 2012.

[22] U. Shevade, H. H. Song, L. Qiu, and Y. Zhang. Incentive-aware routing in dtns. In *Proceedings of IEEE ICNP*, pages 238–247, 2008.

[23] T. Ning, Z. Yang, X. Xie, and H. Wu. Incentive-aware data dissemination in delay-tolerant mobile networks. In *Proceedings of IEEE SECON*, 2011.

[24] K. Srinivasan, S. Rajkumar, and P. Ramanathan. Incentive schemes for data collaboration in disruption tolerant networks. In *Proceedings of IEEE GLOBECOM*, 2010.

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

[26] V. Conan, J. Leguay, and T. Friedman. Characterizing pairwise inter-contact patterns in delay tolerant networks. In *Proceedings of ACM SenSys*, pages 321–334, 2007.

[27] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. Crawdad data set cambridge/haggle (v. 2009-05-29), 2009.

[28] N. Eagle, A. Pentland, and D. Lazer. Inferring social network structure using mobile phone data. In *Proceedings of the National Academy of Sciences*, pages 15274–15278, 2009.