Maximizing Timely Content Advertising in DTNs

Weixiong Rao*, Kai Zhao*, Yan Zhang†, Pan Hui‡ and Sasu Tarkoma*§
*Department of Computer Science, University of Helsinki, Finland, {firstname.lastname}@cs.helsinki.fi
†Simula Research Laboratory, Norway, yanzhang@simula.no
‡Deutsche Telekom Laboratories, Germany, pan.hui@telekom.de

***Abstract*— Many applications, such as product promotion advertisement and traffic congestion notification, benefit from the opportunistic content exchange in Delay Tolerant Networks (DTNs). An important requirement of such applications is timely delivery. However, the intermittent connectivity of DTNs may significantly delay content exchange and cannot guarantee timely delivery. The state-of-the-arts capture the mobility patterns or social properties of mobile devices. However, there is little optimization in terms of the delivered content. Without such optimization, the content demanded by a large number of subscribers could follow the same forwarding path as the content by only one subscriber. To address the challenge, in this paper, we separate content routing from content forwarding. For content routing, we leverage content properties to derive an optimal routing hop count for each content in order to maximize the number of nodes which receive demanded content. Next, for timely forwarding, we develop node utilities to capture interests and mobility patterns of mobile devices for the selection of content carriers. The distributed greedy relay scheme, *Ameba*, leverages the optimal routing hop count and developed utilities to timely relay content to the needed nodes as fast as possible. Illustrative results show that *Ameba* is able to achieve comparable delivery ratio as the Epidemic but with much lower overhead.**

I. Introduction

In Delay Tolerant Networks (DTNs), whenever mobile devices (PDAs, vehicles, mobile phones, etc.) encounter each other, they exchange content via short-range communications (e.g., Bluetooth or WiFi). Thus, many applications and services, such as product promotion advertisement and traffic congestion notification, benefit from the opportunistic content exchange. It is useful in the areas where wireless networks do not cover, wireless access is blocked, or cellular networks are congested [10]. Opportunistic communication thus helps expanding the network coverage, without building dedicated network infrastructures.

An important requirement associated with exchanged content is freshness. That is, besides delivering the content towards the needed users, we further expect the content to be *timely* delivered. For example, timely delivery of certain product promotion news can be critical, e.g., at least before the end day of the product promotion period. Otherwise, outdated content, though delivered, is meaningless to users.

However, opportunistic delivery fashion essentially conflicts the freshness requirement. Service providers expect to maximize the number of potential users who can *timely* receive content, and hence the items must be delivered in time. On the other hand, due to the intermittent connectivity of DTNs, the opportunistic exchange may experience certain *delay*, and cannot guarantee an expected delivery delay [4].

In this paper, we study the problem of maximizing the number of users who can timely receive content. In detail, we are inspired by the topic-based model [6] to offer personalized content subscription. This model is widely used in many real applications (e.g., RSS feeds, online games). Subscribers declare their interests by specifying topics inside subscription conditions (called filters). An advertisement (which precisely means a content item in this paper) is associated with a topic. An advertisement *matches* a filter (or a filter matches an advertisement), if and only if both the advertisement and filter contain the same topic.

The state-of-the-arts [7], [11], [15], [17] leveraged mobility patterns or social properties of mobile devices to select forwarding paths. However, there is little optimization in terms of the delivered content. Without such optimization, the content demanded by a large number of subscribers could follow the same forwarding path as the (same) content demanded by only one subscriber. It is because the previous work mainly leveraged on the properties of mobile devices, instead on the content properties, to optimize the overall content routing. Real applications show many unique content properties, such as correlation between content supplies and demands [16], the well-known Zipf distribution [2].

Based on these content properties, we propose a new content relay scheme, namely *Ameba*. Specially, differing from the above work, *Ameba* shares some similar idea to the content-centric networking [12], [13], [14]. It separates the end-to-end routing (from publishers to subscribers) from the point-to-point forwarding among encountered nodes. After reviewing related works (Section II), we defines the problem and highlights the solution (Section III), and make the following contributions to support the novel idea of separating the content routing from content forwarding:

- We develop an optimal strategy to design an optimal hop count for each *routed* content. The content with such routing hops is expected to reach the needed nodes (Section IV).
- By leveraging the developed utilities to capture the interests and mobility patterns of mobile devices (Section V), we design an adaptive algorithm to select the best carriers for timely *forwarding* (Section VI).
- We extensively evaluate the efficiency of the *Ameba* over two realistic data sets and verify that *Ameba* is able to achieve comparable delivery ratio as the Epidemic but with much lower overhead (Section VII).

§The last author is also affiliated with Department of Computer Science and Engineering, Aalto University, Finland

II. Related Work

In the context of DTN, [23] studied Epidemic routing by selecting all nodes in the network as relays. ProPHET [15] leveraged encounter history to predict the delivery possibility. [22] sprayed data to relays waiting for contacts with destinations. [1] leveraged a Markovian model to develop an efficient utility function for content dissemination. [3] used the path likelihoods to schedule packets transmitted to other peers and packets to be dropped. These works focused on capturing mobility patterns of mobile devices, and all designed for one-to-one routing. Recent works [7], [9], [11], [17] took advantage of social behaviors of mobile devices to identify the best information carrier. These works exploited social properties (e.g., centrality,) to support the one-to-one routing [11], [17], [3], and one-to-many routing [7], [9]. Differing from these works, we observe that content properties could optimize content delivery and design an optimal routing hop count for each routed content.

In terms of utilities for the selection of carriers, [7], [11], [9] mainly used the utilities to predict the delivery possibility of mobile devices. Differing from these works, the main purpose of our developed utilities is to capture the interests of mobile device (together with their mobility patterns). For the details to compute the utilities, [7] did not adopt an accumulative utility function. Though having the cumulative contact length as edge weights in social networks, [11] did not leverage such weights for data forwarding, and [9] used the same utilities for various content items. Instead, our proposed utilities incorporate the self-interest of a specific node and overall interests of the whole DTN, and develop a specific utility for each topic.

In content-centric networking [12], [13], a communication network allows a user to focus on the data he or she needs, rather than having to reference a specific, physical location where that data is to be retrieved from. Differing from such works with the vision for a clean slate design of future Internet, we leverage content properties to derive an optimal routing hop count. In addition, the previous works [19], [20] shared some similar idea to proactively assign more resources (such as replicas) for popular content, but were designed for P2P networks.

III. Problem Statement and Solution Overview

We expect DTNs as complementary network communication technologies [10]. They help content providers deliver the content (generated by the content providers) to the needed nodes and explore more potential customers. Suppose that the N opportunistically encountered devices are connected as a DTN. The roles of mobile devices can be *publishers*, *subscribers* or *carriers*. Publishers publish advertisements d_i of specific topics t_i. Subscribers register subscription filters (containing defined topics t_i) to expect the advertisements of t_i. In addition to publishers and subscribers, mobile devices can act as *carriers* to relay advertisements. When mobile devices encounter each other, the carried advertisements are then exchanged, and finally relayed from the publishers to the subscribers with the help of intermediate carriers.

Due to the diverse advertisements and interests, publishers and subscribers define their favorite topics. Suppose there exist a total number of T topics. For a specific topic t_i $(1 \leq i \leq T)$, we assume that N_i nodes are interested in t_i (if a node registers a filter containing t_i, we say that the node is interested in t_i). We define the *demand rate* p_i of a topic t_i to be the ratio between N_i and N, i.e. $p_i = N_i/N$. Let M denote the total number of advertisements. We next define the *supply rate* q_i with respect to (w.r.t) a topic t_i, and thus $q_i \cdot M$ advertisements are associated with t_i.

Given the above notations, we define the following problem:

Problem 1 *Within a given time period P, we want to design an advertisement relay scheme to maximize the number of nodes which can receive the matching advertisements.*

In Problem 1, an advertisement matches a node if the node registers a filter having the same topics as the advertisement.

Given Problem 1, our basic idea is to separate content routing from content forwarding. For content routing, we leverage content properties to design an optimal hop count for each routed content. Next, for content forwarding, we develop the node utilities to capture interests and mobility patterns of mobile devices. Then the content forwarding leverages the utilities to select the best carriers for routed content.

In detail, in Section IV, for each advertisement d_i (with a topic t_i), *Ameba* develops an optimal number X_i of opportunistic encounters for d_i. Intuitively, the optimal number X_i can be treated as the overall routing hop count to route d_i from publishers to subscribers. That is, when *Ameba* creates only one copy for d_i, X_i is the number of hops (or TTL) to route the d_i towards the needed nodes. When *Ameba* creates multiple copies of some $d_{i'}$ $(\neq d_i)$, $X_{i'}$ is the *accumulative* number of hops to route all copies of the $d_{i'}$. Thus, *Ameba* adaptively allows either the unicast (e.g., for d_i) or multicast (e.g., for $d_{i'}$) to maximize timely content advertising. The key of *Ameba* is to leverage the content properties and proactively assign more hop count for popular advertisements which are highly demanded by a large number of subscribers. Thus, more nodes act as the carriers to route the popular advertisements, and subscribers timely receive the needed advertisements.

Note that the developed X_i in Section IV does not consider mobility patterns of DTN nodes. Thus, Section V designs the utilities to capture the mobility patterns and interests of mobile devices. Then, based on the utility of each node, *Ameba* selects the best carrier for each d_i to timely forward d_i towards the needed nodes. Specifically, we use a vector U^j to represent the utilities of a node n_j. Given T topics, U^j contains T elements. For a topic t_i $(1 \leq i \leq T)$, the element $u_i^j \in U^j$ represents how good the node n_j can successfully relay the advertisements with the topic t_i to the needed nodes. The vector U^j and the element u_i^j are named as the *vector utility* and an *element utility*, respectively. Based on the element utilities, *Ameba* can identify (i) the nodes that are interested in the topic t_i and (ii) how fast the encountered node can timely forward d_i to the needed nodes.

Symbol	Meaning
T	Total number of topics
M	Total number of advertisements
N	Total number of nodes in DTN
p_i	Demanding rate of a topic t_i $(1 \leq i \leq T)$
q_i	Supplying rate of a topic t_i $(1 \leq i \leq T)$
d_i	An advertisement associated with a topic t_i $(1 \leq i \leq T)$
$\mathcal{N}_i$	the set of nodes interested in t_i, with the cardinality N_i
H_i	Number of nodes which can successfully receive d_i
X_i	Number of required encounters to relay d_i to the needed nodes
H	Total number of nodes which receive matching advertisements
X	Total number of opportunistic encounters to relay advertisements

TABLE I

MEANINGS OF MAIN SYMBOLS USED

Given the above two techniques, in Section VI, *Ameba* selects the best carriers of each advertisement, such that an advertisement d_i and its copies experience X_i encounters as fast as possible. We prove that the problem of selecting the best carriers to relay advertisements is NP-hard. Then, we propose the algorithm details of *Ameba*.

IV. Optimal Advertisement Relay Strategy

We first highlight the general idea as follows. We define the event that two nodes encounter as an *encounter event*, no matter mobility pattern and social properties of mobile devices. For example, given three nodes n_j, $n_{j'}$ and $n_{j''}$, first n_j encounters $n_{j'}$ after 1 day, and then n_j encounters $n_{j''}$ after only 1 minute. The two occurred events are treated equally, because the occurrence of such events depends upon whether or not two nodes encounter, rather than the experienced period or encounter frequency.

Based on the encounter events, we derive a function $f(\cdot)$ between the needed encounters to relay an advertisement and the expectation of the number of nodes which can successfully receive the matching advertisements (Section IV-A). Next, considering the overall advertisements, we develop a strategy to ensure that each advertisement experiences an optimal number of encounters, such that we can maximize the total number of nodes which can successfully receive matching advertisements (Section IV-B).

Essentially, the techniques of this section are to exploit the properties of advertisements (i.e., the demanding and supplying rates), and do not consider the effect of mobility pattern of mobile devices. Such patterns are considered by the developed utilities for the selection of carriers (Section V). Table I summarizes the main symbols used in this section.

A. Probabilistic Advertisement Relay

Given an advertisement d_i with a topic t_i, we denote $\mathcal{N}_i$ to be the set of the nodes that expect the advertisement d_i (i.e., $N_i = p_i \cdot N$ is the cardinality of $\mathcal{N}_i$). For a specific encounter event among all encounter events, we compute the probability ρ_i that the encounter successfully relays an advertisement d_i to a node inside $\mathcal{N}_i$. Suppose a node carrying d_i is opportunistically encountering another node, which could be a member node inside $\mathcal{N}_i$ or not. Among the nodes in the DTN, N_i nodes are the members of $\mathcal{N}_i$. Thus, the average probability ρ_i is $\frac{N_i}{N}$, because (i) the defined encounter events depend only upon whether or not nodes encounter, instead of the mobility patterns and social properties of mobile devices and (ii) ρ_i now depends only upon the distribution of filters among the nodes of DTN.

Based on the above analysis, the probability ρ_i^1 of a specific encounter to relay d_i to the *first* new node inside $\mathcal{N}_i$ is $\frac{N_i}{N}$, because *any* node inside $\mathcal{N}_i$ could be the node to receive d_i.

Next, the probability ρ_i^2 that a specific encounter successfully relays d_i to the *second* new node in $\mathcal{N}_i$ is $\frac{N_i}{N} \cdot (1 - \frac{1}{N})$, where (i) $\frac{N_i}{N}$ is the probability of a specific encounter to successfully relay d_i to any node inside $\mathcal{N}_i$ and (ii) $(1 - \frac{1}{N})$ is the probability that among the nodes $\mathcal{N}_i$, any of those nodes (except the first new node) receives d_i. The reduction of $1/N_i$ is to eliminate the situation that d_i is duplicately forwarded to the first new node that has already received d_i.

Similarly, the probability ρ_i^j that a specific encounter to successfully relay d_i to the j-th new node inside $\mathcal{N}_i$ is

$$\rho_i^j = \frac{N_i}{N} \cdot (1 - \frac{j-1}{N_i}) \tag{1}$$

In Equation (1), the item $(1 - \frac{j-1}{N_i})$ represents the probability that any of the nodes inside the set $\mathcal{N}_i$ (except the previous $(j-1)$ nodes) can receive the advertisement. Given the probability ρ_i^j, the trial that a specific encounter successfully relays d_i to the j-th new node inside $\mathcal{N}_i$ can be treated as a *geometric* probability distribution with the parameter ρ_i^j. Thus, the expected number of opportunistic encounters needed to successfully relay d_i to the j-th new node is

$$X_i^j = \frac{1}{\rho_i^j} = \frac{N}{N_i} \cdot \frac{N_i}{N_i - j + 1} = \frac{N}{N_i - j + 1} \tag{2}$$

Suppose our advertisement relay strategy will ensure that d_i experiences X_i opportunistic encounters towards the nodes in $\mathcal{N}_i$. We are interested in the number of nodes in $\mathcal{N}_i$ that can successfully receive d_i. Let H_i denote the number of nodes in $\mathcal{N}_i$ that can successfully receive d_i. Then, given the number X_i, we compute H_i as follows:

$$\sum_{j=1}^{H_i} X_i^j = \sum_{j=1}^{H_i} \frac{N}{N_i - j + 1} = X_i \tag{3}$$

For Equation (3), we make the following transformation:

$$\sum_{j=1}^{H_i} \frac{N}{N_i - j + 1} = N \cdot \left(\sum_{j=1}^{N_i} \frac{1}{j} - \sum_{j=1}^{N_i - H_i} \frac{1}{j} \right) \approx N \cdot \ln \frac{N_i}{N_i - H_i} \tag{4}$$

The above transformation (with $H_i < N_i$) utilizes the result that the summation $\sum_{j=1}^{N_i} \frac{1}{j}$, i.e., the harmonic number, is approximated by $\ln N_i$. Similar reasoning applies for $\sum_{j=1}^{N_i - H_i} \frac{1}{j}$ and it is approximated by $\ln(N_i - H_i)$. Consequently, we can derive the function $f(\cdot)$ between H_i and X_i as follows:

Theorem 1 *If our relay strategy ensures that an advertisement d_i associated with t_i can experience X_i opportunistic encounters, we have the function between H_i and X_i as follows.* $H_i = N \cdot p_i \cdot \left(1 - e^{-\frac{X_i}{N}}\right)$.

Proof: Based on Equations (2-4), we obtain $X_i = N \cdot \ln \frac{N_i}{N_i - H_i}$, and then validate the correctness of the theorem. ■

In case of $X_i = N \log N_i$, we have $H_i = N_i$.

B. Optimal Relay Strategy

In this subsection, considering the overall advertisements associated with T topics, we develop an optimal relay strategy to maximize the total number of nodes which can successfully receive matching advertisements.

By Theorem 1, if an advertisement d_i experiences X_i opportunistic encounters, H_i nodes can successfully receive d_i. We consider the effect of the supplying rate q_i as follows. Suppose that among all of the M distinct advertisements, the topic t_i is supplied with $q_i \cdot M$ advertisements. When such advertisements are independently published by publishers, by $q_i \cdot M \cdot X_i$ opportunistic encounters, $q_i \cdot M \cdot H_i$ nodes can successfully receive matching advertisements.

Given the total T topics, the number of nodes which receive the matching advertisements, denoted as H, is computed by:

$$H = \sum_{i=1}^{T} [q_i \cdot M \cdot H_i] = M \cdot N \cdot \sum_{i=1}^{T} \left[q_i \cdot p_i \cdot (1 - e^{-\frac{X_i}{N}}) \right] \tag{5}$$

On the other hand, mobile devices opportunistically encounter each other, and the occurrence of the content exchange depends on whether or not such devices encounter. Therefore, the overall opportunistic encounters to relay advertisements are decided by the mobility pattern of mobile devices (and other parameters like battery, etc.). Thus, within a given time period P, the total number of such opportunistic encounters is a given number X. Hence, we consider that the constraint that the total number of encounters, i.e., $\sum_{i=1}^{T} (q_i \cdot M \cdot X_i)$, is associated with at most X encounters.

Formally, given the optimization problem to maximize the H in Equation (5) with the constraint $\sum_{i=1}^{T} (q_i \cdot M \cdot X_i) = X$, we have the following theorem:

Theorem 2 *Given the constraint $\sum_{i=1}^{T} (q_i \cdot M \cdot X_i) = X$, the total number H of nodes receiving the matching advertisements is maximized when $X_i = \frac{X}{M} + N \ln p_i - N \sum_{i=1}^{T} (q_i \ln p_i)$.*

To prove Theorem 2, we leverage the Lagrange multiplier method to obtain the optimal X_i in terms of p_i and q_i. Due to the space limit, we skip the details of such proof. We refer interested readers to the technical report [21] for the details.

Based on Theorem 2, we have the following discussion.

- When p_i is larger, i.e., more nodes are interested in t_i, the item $\ln p_i$ is a smaller negative value, thus resulting in a larger X_i. In terms of physical interpretation, if d_i is demanded by more nodes, the relay strategy should ensure that more nodes act as the carriers to relay d_i, and the nodes carrying d_i encounter more nodes in $\mathcal{N}_i$.
- When X, M and N are given, X_i depends on the distribution of p_i and q_i. When the distribution of p_i and q_i is more skewed, $-N \sum_{i=1}^{T} (q_i \ln p_i)$ and X_i become smaller. This means that the skewed distribution functions of p_i and q_i help use smaller X_i to relay advertisements to the nodes $\mathcal{N}_i$. Many real applications indicate a skewed distribution of p_i and q_i [18], [6], which favors the optimization strategy.

The solution to the optimization problem in Theorem 2 needs to know the values of the parameters p_i, q_i, etc, which can be acquired by content providers as follows. First, since the content is generated by content providers, the providers can easily compute the parameters (e.g., q_i, etc.) w.r.t. the content. To compute the parameters (e.g., p_i, etc.) w.r.t. filters, subscribers can be required to register subscription filters to the registration server hosted by the content providers (That is possible because we expect that the DTN is as a complementary network communication technologies). Besides, the distributed solution, e.g., [11], helps approximate the parameters p_i and q_i. Finally, for the content with low p_i but required to be urgently delivered to the needed nodes (such as the alarm of urgent events), we assign the content with a high priority. Then, the content is not evicted by the buffer manager (which uses the least recent used (LRU) policy to evict expired items) and is timely relayed to the encountered nodes.

V. Relay Utilities

In this section, we give the details of the developed utilities, which capture the subscription interests and mobility pattern of mobile devices, for the selection of node carriers.

A. Data Structure

For a node n_j, we use a vector U^j (called *vector utility*) to represent n_j's utilities. To capture mobile devices' interests, U^j contains T elements (due to the existence of T topics), and each element is associated with a topic. The element $u_i^j \in U^j$ $(1 \leq i \leq T)$, called *element utility*, indicates the goodness of n_j to successfully relay an advertisement d_i (with a topic t_i) to the nodes in $\mathcal{N}_i$. A larger u_i^j indicates that n_j has more chance to successfully relay d_i to the nodes in $\mathcal{N}_i$.

The proposed utilities have two unique properties to capture the interests of mobile devices.

- Among all elements in U^j, the one with the largest value is associated with the topic t_i that node n_j itself is interested in. Thus, if the utility U^j is available to a third node n_k, by identifying the largest element inside U^j, n_k then identifies that n_j is interested in the topic associated with such an element, before n_k communicates with n_j. This property is useful for *Ameba* to timely relay d_i towards the needed nodes.
- Except the element with a largest value, the remaining elements in U^j accumulate the interests of the whole nodes in the DTN. That is, if a topic t_i is popularly registered in many nodes of DTN, the associated element is larger. In addition, the utility U^j incorporates the mobility pattern (e.g., encountering rate) of n_j.

In Figure 1, the leftmost column shows the utility vector of a node n_1. For the 4 topics $t_A ... t_D$, n_1 is assumed to be interested in t_A. Thus, the element of t_A has the largest value 0.614, indicating that n_1 is self interested in t_A. In addition, since t_B is popularly registered in 3 other nodes (n_2, n_3 and n_4), the associated element utility, 0.271, is also larger than the two remaining element utilities (i.e., 0.071 and 0.042).

In detail, for a specific topic t_i, the element utility u_i^j has considered the following cases. (i) The node n_j itself is

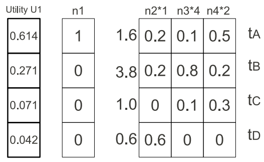


Fig. 1. An example to compute the utilities of n_1 for 4 topics $t_A ... t_D$

interested in t_i, and (ii) n_j, though not interested in t_i, can act as the carrier of d_i to successfully relay d_i towards $\mathcal{N}_i$. Therefore, we need the following information to compute u_i^j.

- For the interest of the node n_j itself, we use a 0/1 vector of T elements r_i^j to represent its own interest. The element r_i^j pertaining to t_i is equal to 1 (i.e., $r_i^j = 1$), iff the node n_j is registered with a filter defining t_i.
- The utilities U^k of the nodes (say n_k) that n_j previously encountered within a specific period P. U^k (resp. U^j) is available to n_k (resp. n_j) by exchanging the utilities when n_j and n_k encounter each other.
- The encountering rate $f_{j,k}$ of two mobile devices n_j and n_k. The value of $f_{j,k}$ captures the frequency that n_j and n_k ever encountered (note that $f_{j,k}$ can be adapted by considering the other aspects of n_j and n_k, such as the battery level, social properties and network bandwidth).

In Figure 1, the 2nd leftmost column indicates the 0/1 vector in terms of n_1's self-interest: the element pertaining to t_A is 1 because n_1 is interested in the topic t_A. The three columns from the 3rd leftmost column to the rightmost column in Figure 1 show the utilities of 3 previously encountered nodes n_2, n_3 and n_4, respectively. The items above the three columns show the encountering frequency (i.e., 1, 4, and 2) between n_1 and the three nodes inside a period. Following [7], the nodes that ever frequently encountered have high chance to encounter again in the future. Thus, $f_{j,k}$ helps predict the chance that n_j and n_k will encounter each other. More other techniques (e.g., by leveraging spatial and social properties) can be incorporated for further improvement.

Though the computation of the vector utility U^j does need all above items, the following subsection will show that it is unnecessary for n_j to maintain u_i^k and $f_{j,k}$ for all previously encountered nodes n_k. Thus, we avoid the expensive maintenance overhead, particularly when n_j encountered a large number of nodes.

B. Details to compute utilities

Suppose that n_j previously encountered K nodes n_k ($1 \leq k \leq K$) inside a window (e.g., a sliding window of 6 hours). To compute U^j, the *accumulation* technique captures the overall interests of the whole DTN, and the *normalization* technique captures the self-interests of n_j. In detail, the node n_j uses the following four steps to compute U^j.

(1) *Overall Accumulation*: For all of these K nodes n_k that n_j encounters, we compute the *overall utility* $U'(i)^j$ and its element $u'(i)^j$. For an element $u'(i)^j$ in terms of a topic t_i, we accumulate $u'(i)^j = \sum_{k=1}^{K} (f_{j,k} \cdot u_i^k)$ for all such nodes n_k. For example, in Figure 1, we compute $u'(i)^j$ for the topic t_A as 1.6 ($= 1 \times 0.2 + 4 \times 0.1 + 2 \times 0.5$). Similarly, we can compute the overall element utilities for other 3 topics t_B, t_C and t_D as 3.8, 1.0 and 0.5, respectively.

(2) *Overall Normalization*: We normalize the utility $U'(i)^j$ to $U(i)^j$ by computing the element $u(i)^j = u'(i)^j / \sum_{i=1}^{T} u'(i)^j$. Following the above example, for a topic t_A, its normalized overall utility is 1.6/(1.6+...+0.6)=1.6/7.0.

(3) *Self Accumulation*: Based on the normalized overall utility $U(i)^j$ and the self-interest r_j^i, we compute the utility U'^j by setting its element u'^j_i to $u'^j_i = \frac{r_i^j + u(i)^j}{\sum_{i=1}^{T} r_i^j + \sum_{i=1}^{T} u(i)^j} = \frac{r_i^j + u(i)^j}{\sum_{i=1}^{T} r_i^j + 1.0}$. For example for t_A, $u'^j_{t_A} = \frac{1+1.6/7.0}{1+1}$.

(4) *Self Normalization*: we finally normalize the utility U'^j to U^j by $u_i^j = u'^j_i / \sum_{i=1}^{T} u'^j_i$ (e.g., for t_A, $u_{t_A}^j = 0.614$).

During the above steps, step 1 *accumulates* the utilities of the previously encountered nodes n_k to capture the overall interests of the nodes n_k, such that n_j acts as the carrier to relay d_i from n_k towards the nodes $\mathcal{N}_i$ intermediately going through n_j. Next, step 3 accumulates the self-interests of n_j with the overall utilities. Thus, the accumulation ensures that U^j can measure the goodness to relay d_i towards n_j itself and other interested nodes.

In addition, given the self-interests with $r_i^j = 1$, the *normalization* of steps 2 and 4 ensures that among all elements in U^j, the element u_i^j is associated with the largest value. This property helps relay the advertisement d_i towards those nodes in $\mathcal{N}_i$ via the greedy policy in the proposed *Ameba* scheme.

As mentioned previously, it is unnecessary to maintain the u_i^k and $f_{j,k}$ for all previously encountered nodes n_k. Instead, we only maintain the accumulated overall utility $U'(i)^j$ (see the above step 1). When n_j encounters a new node n_k, the element $u'(i)^j$ of $U'(i)^j$ is incrementally updated by $u'(i)^j = u'(i)^j + u_i^k$. After that, we still follow the above steps 2-4 to renew the utility U^j. Therefore, the node n_j only needs to maintain the overall utility $U'(i)^j$, the self-interests r_i^j, and the final utility U^j, and the associated maintenance cost is low.

VI. Distributed Greedy Relay Scheme

Given the assigned opportunistic encounters X_i (given by Theorem 2 of Section IV), we develop a distributed relay algorithm, *Ameba*, which leverages the developed utilities to select the best nodes as the carries of d_i.

A. Two Basic Solutions

Before presenting the proposed relay algorithm, we first give two basic solutions. Since Theorem 2 assigns X_i encounters for the advertisement d_i, we intuitively consider that the advertisement d_i (and its copies, if have) is associated with an overall TTL equal to X_i.

Suppose that the initial publisher node n_1 starts the relay of d_i. The first basic solution, namely *broadcast*, is shown as follows. Whenever the initial node n_1 encounters a node n_2, d_i is copied to the node n_2. If n_2 registers a filter f matching d_i, then d_i is successfully disseminated to the needed node. After that, both n_1 and n_2 remain the copies of d_i and act as the carriers of d_i. Meanwhile, such two copies of d_2 are associated with a new TTL, which is equal to $(X_i - 1)/2$. If n_1 and n_2 encounter more other nodes, d_i is similarly copied and the associated TTLs are renewed. The relay of d_i (and its copy) is stopped when the current TTL is equal to 0.

Second, based on the *unicast* solution, when n_1 encounters n_2, d_i is relay from n_1 to n_2 if the element utility u_i^2 is larger than u_i^1; meanwhile, n_1 does not remain a copy of d_i. Then, n_2 acts as the unique carrier of d_i with a new TTL $(X_i - 1)$. When n_2 encounters a third node n_3, if $u_i^3 > u_i^2$, then d_i is relayed to n_3 associated with the new TTL $(X_i - 2)$. As before, the unicast is finished when the current TTL is equal to 0. Different from the broadcast scheme, the unicast scheme has only one copy of d_i during the whole relay process.

The above broadcast and unicast schemes suffer from disadvantages. First, the broadcast incurs too many copies of advertisements. Due to a limited buffer size in each node, too many copies of advertisements incur the overflow issue and evict useful advertisements (we use the least recent used (LRU) policy to evict the advertisements). On the other hand, though avoiding the overflow issue, the unicast incurs a high latency before d_i is successfully relayed to the expected nodes. The high latency may be even larger than the given time period P, and the advertisement becomes expired.

In this section, the *Ameba* scheme combines the benefits of the unicast and broadcast schemes, and relay (either broacast or unicast) advertisements towards the needed nodes. Depending on the utilities of encountered nodes, the *Ameba* leverages the developed utilities to select the best carriers, and adaptively adjusts the number of advertisement copies to timely deliver the advertisement to the needed nodes as fast as possible.

B. Problem Statement

We model the DTN as an opportunistic graph G. Each node n_j $(1 \leq j \leq N)$ in DTN is represented by a corresponding vertex in G. If the node n_j opportunistically encounters a node $n_k(\neq n_j)$, there is an edge between n_j and n_k. In the graph G, each node n_j is associated with a utility vector U^j, and each element utility u_i^j in the vector U^j represents the utility of n_j to relay the advertisement d_i. The weight associated with the edge between n_j and n_k depends upon the element utility u_i^j and the encountering frequency $f_{j,k}$ between two nodes n_j and n_k. Larger values of u_i^j and $f_{j,k}$ indicate higher chance of n_j to successfully relay d_i from n_j to n_k (and vice versa). Thus, we can designate that the edge between n_j and n_k with a weight $(f_{j,k} \times u_i^j)$. We formulate the following problem.

Problem 2 *Given an opportunistic graph G, we want to find X_i connected edges starting from n_j, such that the sum of all such edges are maximized.*

The optimization problem to maximize the sum of X_i connected edges of filters is harder than its decision problem. If a solution of the optimization problem is known, we can find the vertices which are the endpoints of such edges. Then, by simply counting the number of nodes that appear in $\mathcal{N}_i$, we can compare the number of such nodes with the given number H_i. We can prove that Problem 2 is NP-hard by reducing it from the classic set cover problem. Therefore, we give a heuristic algorithm as follows.

C. A Distributed Heuristic Solution

Since no node in DTN knows the global opportunistic graph G, we can not directly apply the classical greedy algorithm of the set cover problem. We instead propose a distributed greedy-based heuristic algorithm, *Ameba*, to select the best nodes in DTN to timely relay the advertisement d_i. We suppose that the relay of d_i starts from an initial node n_j.

Algorithm 1: Relay_Selection

input : Node n_j, carrying d_i, is opportunistically encountering node n_k

1 n_j (resp. n_k) updates U^j (resp. U^k) by exchanging utilities with n_k (resp. n_j);
2 **if** *n_k is interested in d_i* **then** d_i is forwarded to n_k;
3 **if** *the element utility u_i^k is larger than the element utility u_i^j* **then**
4   n_k keeps a copy of d_i;
5   **if** $u_i^k > \mu_i^j$ `// /*`μ_i^j `is the largest element utility of the topic` t_i `among all nodes that` n_j `ever encountered*/`
6   **then** n_j removes d_i;
7 **else** n_k does not keep a copy of d_i;

As shown in Algorithm 1, the two encountering nodes n_j and n_k exchange and update their utilities U^j and U^k (Section V has already given the details to compute the utilities). Next, when the node n_j is interested in d_i, then d_i is copied from t_i to n_k. After that, n_k receives the matching advertisement d_i.

In line 3, if the element utility u_i^k of n_k in terms of the topic t_i is larger than the element utility u_i^j of n_j in terms of t_i, then n_k keeps a copy of d_i in the buffer (line 4). Otherwise, n_j will not keep a copy of d_i (line 7), if one of the following cases occurs: (i) n_j is interested in t_i but n_k not, indicating that $u_i^k < u_i^j$ must hold, (ii) neither n_j nor n_k is interested in t_i, but $u_i^k < u_i^j$ holds, and (iii) both n_j and n_k are interested in t_i, but $u_i^k < u_i^j$ holds. It means that even if line 2 is executed, n_k still removes d_i (line 7). This makes sense because n_j is better than n_k to act as the carrier of d_i (e.g., n_k is interested in t_i but none of its previously encountered nodes is interested in t_i. Thus it is meaningless for n_k to act as the carrier).

In lines 5-6, there is another option whether or not n_j still needs to keep a copy of d_i in the buffer and acts as the carrier of d_i. If u_i^k is larger than μ_i^j, i.e., the largest utility of the topic t_i among all nodes that n_j ever encountered, then n_j removes d_i from its buffer and will not act as the carrier of d_i. Otherwise, n_j still keeps d_i in its buffer and acts as the carrier of d_i. During the encounter between n_j and n_k, μ_i^j is updated if u_i^k is larger than the current μ_i^j.

In addition, Algorithm 1 updates the TTL of d_i as follows. Suppose that the current TTL associated with the advertisement d_i is X_i' (X_i' is initially equal to X_i). If both n_j and n_k keep the copies of d_i in their local buffers, the new TTLs associated with both copies are set by $(X_i' - 1)/2$. Otherwise, if either n_j or n_k keeps a single copy of d_i, the new TTL associated with such a copy is $(X_i' - 1)$. The relay of a copy of d_i is terminated if the associated TTL is zero.

Based on Algorithm 1, those nodes having larger utilities u_i^j are selected as the carriers of d_i, and d_i is greedily relayed to the nodes with larger element utilities. The greedy policy helps relay d_i to the nodes in $\mathcal{N}_i$ as fast as possible. In addition, Algorithm 1 follows the idea of broadcast to allow multiple copies of d_i, and thus increases the delivery ratio. Algorithm 1 also follows the idea of unicast to limit the number of such copies in lines 3-6, and avoids creating too many copies. Our experiment shows that Algorithm 1 efficiently delivers d_i towards the needed nodes with high delivery ratio and significantly low overhead.

VII. Evaluation

A. Experimental Settings

We compare the *Ameba* scheme with the Epidemic scheme [23] (i.e., the flooding-based approach), ProPHET [15] (i.e., the probabilistic approach), and Bubble Rap (i.e., social aware approach). Note that for Bubble Rap, when a content item is needed by multiple subscribes and such subscribers are located at multiple communities, we then have to forward the item to all such communities. In addition, we also compare the *Ameba* scheme with the broadcast and unicast schemes having the optimal encounters, both of which are given in Section VI-A.

We use two real-world datasets of human mobility traces to run our simulation. The Infocom06 dataset [5] contains opportunistic Bluetooth contacts between 98 iMotes, 78 of which were distributed to Infocom06 participants and 20 of which were static. The MIT Reality trace [8] comprises 95 participants carrying GSM enabled cell-phones over a period of 9 months. We consider, as in [5], that two phones are in contact when they share the same GSM base station.

Next, following the previous works [6], we use the Zipf distribution to generate filters and content items for a set of given topics. To select subscribers, we ensure that each node registers a filter and thus the number of subscribers (and the number of filters) is equal to the node count. Next, we randomly choose the publishers from all nodes.

Table II shows the parameters used in the experiments. Taking the Zipf parameter α as an example, 0.95 is the default value and the interval $[0.0, 1.2]$ is the allowable range. In addition, we are interested whether or not the skew of filter distribution correlates the skew of content popularity. The *correlation* means that a topic, which is highly demanded by subscribers, simultaneously popularly appears in content items. Otherwise, the skew of both distributions is *anti-correlated*. By default, we set up the correlated topics to generate filters and content (with the Zipf distribution).

Parameter	Infocom06	MIT Reality
num. of topics	10	10
num. of filters	78	95
num. of content items	78×1.5	95×1.5
buf. size per node	30: [5-100]	40:[5-100]
running period P	5: [1-840] mins	21:[1-90] days
Zipf parameter α	0.95: [0.0-1.2]	0.95: [0.0-1.2]

TABLE II
Parameters for Digg Experiments

Finally, we use the following metrics to measure the performance of our simulations. (i) Delivery ratio: the average ratio of the number of delivered destinations to the total number of destinations. (ii) Average delay: the average delay for all the delivered destinations to receive the data. (iii) Average cost: the average number of content transmissions (including transmissions for duplicated copies) used to deliver a data item. Thus, the average cost measures the average *overhead* to deliver the data item. The duplicated copies are shown as follows. Since the topic-based model is essentially a many-to-many communication manner (one topic is associated with multiple content items and multiple subscribers), a content item, after arriving at a subscriber node, still needs to be forwarded to the remaining subscribers. As a result, a content item could duplicatedly reaches the same relay nodes. It is particularly true when existing content items are evicted due to the limited buffer size (we use a LRU eviction policy).

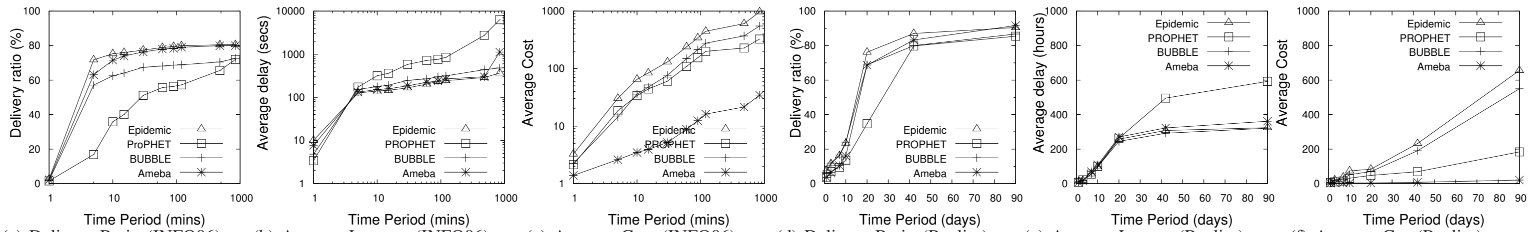
B. Effect of Time Period

First, in Figure 2, we study the effect of the allowable time period P over the Infocom06 and MIT reality traces. As shown in Figure 2 (a), among the four schemes, the *Ameba* scheme achieves comparable delivery ratio as the Epidemic scheme, the Bubble Rap scheme has a lower delivery ratio than *Ameba*, and the ProPHET scheme has the least delivery ratio. In particular, a larger time period P, varying from 1 minute to 840 minutes, leads to a larger delivery ratio. This is due to the sparse DTN property and the low encountering rate of node contacts. A larger P indicates that more nodes have chance to relay content items towards the needed nodes, and thus the delivery ratios of all four schemes increase accordingly.

Second, Figure 2 (b) plots the average delay. Result indicates that the *Ameba* scheme uses a comparable low delay as the Epidemic scheme to deliver content items towards needed nodes. It is because the developed optimization strategy can help optimize the delivery of highly demanded content by using more nodes as carriers, and then content items are delivered as fast as possible.

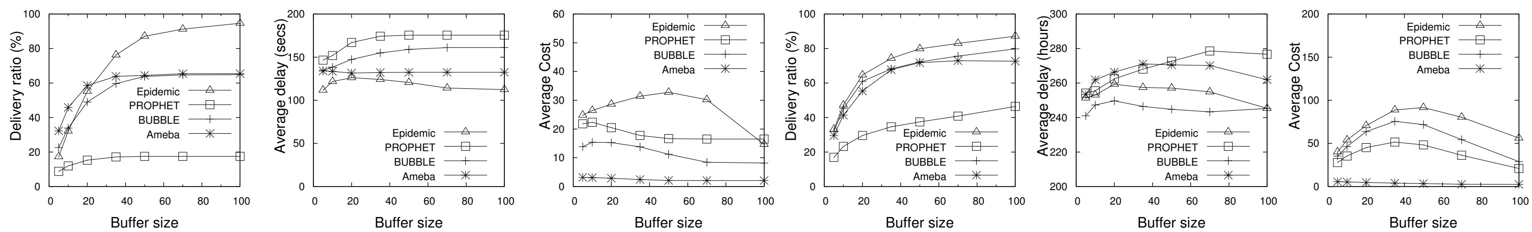
Next for the average cost, Figure 2 (c) shows that the *Ameba* scheme uses a significantly low cost. For example, when P = 840 minutes, the average cost of the *Ameba* scheme is 3.50% of the Epidemic scheme, 5.22% of the Bubble Rap scheme, and 10.3% of the ProPHET scheme, respectively. These results indicate the high efficiency of the *Ameba* scheme. The low cost of *Ameba* is mainly because given the skewed popularity distribution α = 0.95, *Ameba* proactively assigns more optimal encounters for popular items, but meanwhile assigns fewer encounters for unpopular items. Furthermore, the relay algorithm of *Ameba* limits the copies of content items and thus limits the content transmissions. For Bubble Rap, since a content item might be located at multiple communities, the item has to be forwarded to all such communities, leading to the cost of Bubble Rap close to the Epidemic approach. In addition, the average cost shown in this figure is even larger than the number of devices (98 imotes for Infocom06 data set). It is because duplicate content items reach the same relay nodes (caused by the many-to-many topic model and LRU eviction policy for the limited buffer size). Similar situation occurs for other results for the average cost.

When we compare Figures 2 (a-c) with Figures 2 (d-e), the nodes in the MIT reality trace have to use a longer time period than the nodes in the Infocom06 trace, until the almost same delivery ratio (for example, 80%) is achieved. It is



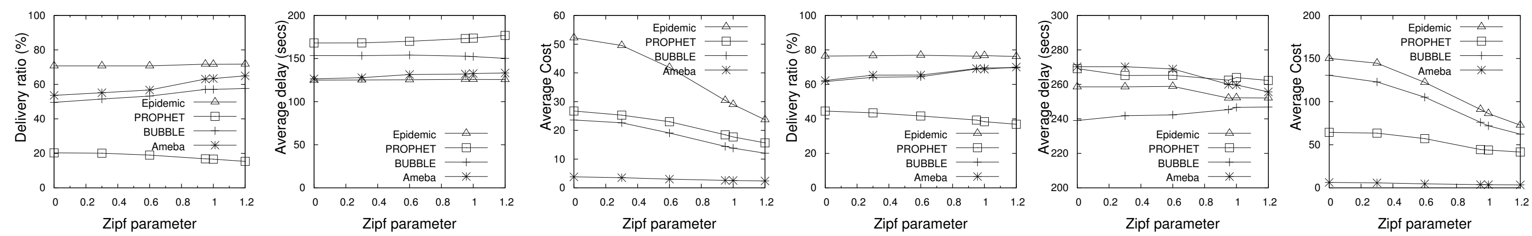
(a) Delivery Ratio (INFO06) (b) Average Latency (INFO06) (c) Average Cost (INFO06) (d) Delivery Ratio (Reality) (e) Average Latency(Reality) (f) Average Cost(Reality)

Fig. 2. Effect of the time period P on the Infocom06 and MIT Reality data traces



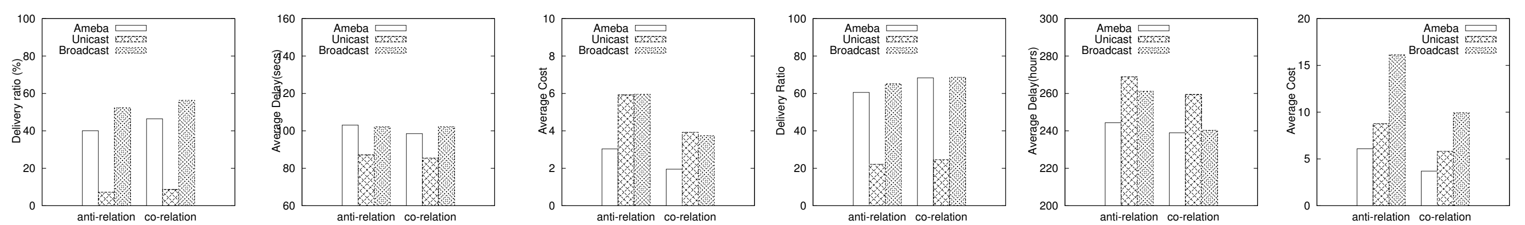
(a) Delivery Ratio (INFO06) (b) Average Latency (INFO06) (c) Average Cost (INFO06) (d) Delivery Ratio (Reality) (e) Average Latency(Reality) (f) Average Cost(Reality)

Fig. 3. Effect of the buffer size S on the Infocom06 and MIT Reality data traces



(a) Delivery Ratio (INFO06) (b) Average Latency (INFO06) (c) Average Cost (INFO06) (d) Delivery Ratio (Reality) (e) Average Latency(Reality) (f) Average Cost(Reality)

Fig. 4. Effect of the Zipf Parameter on the Infocom06 and MIT Reality data traces



(a) Delivery Ratio (INFO06) (b) Average Latency (INFO06) (c) Average Cost (INFO06) (d) Delivery Ratio (Reality) (e) Average Latency(Reality) (f) Average Cost(Reality)

Fig. 5. Comparison of Three Schemes

because the encountering rates of the pairwise nodes in the MIT reality trace are much lower than those in the Infocom06 trace. Nevertheless, consistent with Figure 2 (c), the *Ameba* scheme in Figure 2 (e) uses the least average cost.

C. Effect of Buffer Size

In this experiment, we vary the buffer size S per node (given the size S, when more content items come at the node, the old items are evicted by the FIFO manner), and study the performance of four schemes in Figure 3. In general, a larger S helps achieve higher delivery ratios for all four schemes. Among the four schemes, the Epidemic scheme obviously benefits most from a larger S, because the Epidemic scheme creates the largest number of copies of content items to be propagated across nodes. Similar to the Epidemic and Bubble Rap schemes, the *Ameba* scheme allows more copies of popular content items, and benefits from a larger S. Instead, the ProPHET scheme benefits least from a larger S.

A larger S does not necessarily lead to a smaller average delay and cost. In Figure 3 (c), only after S is sufficient large, the average cost of the Epidemic scheme becomes smaller. This is because the average cost depends on both the number of delivered advertisements to the needed nodes and the number of used relays. Though a larger S does increase the number of delivered items, it also leads to a larger number of relays. After P is sufficient large, the number of delivered items becomes enough large to ensure that the average cost is smaller. Such trend consistently appears in both Figure 3 (c) and Figure 3 (f). The *Ameba* scheme limits the copies of content and a larger S consistently leads to smaller cost.

D. Effect of Zipf Parameter

Figure 4 studies the effect of the Zipf parameter α. By varying α from 0.0 to 1.2, we can find that a larger α helps achieve higher delivery ratio, lower average delay and cost for the *Ameba* scheme. For example, in Figure 4 (a), the delivery ratio of the *Ameba* scheme increases from 26.8% to 49.2% with around 183% growth, and the average cost decreases from 1.815 to 1.259 with around 31% reduction. These results are consistent with our analysis in Section IV, which favors the skewed distribution of p_i and q_i. In addition, for the MIT reality trace, Figures 4 (d-f) indicate the similar growth of the delivery ratio, and the reduction of the average delay and cost.

For the Epidemic scheme, Figures 4 (a) and (d) indicate that the skewed distribution has low effect on the delivery ratio. It is because the Epidemic scheme blindly broadcasts content items towards encountering nodes and does not reduce the overall delivery ratio. However, due to the default correlated distribution of p_i and q_i, the Epidemic scheme ensures that an content item easier encounters the needed nodes. Thus, the average delay and cost are reduced with a larger α. Similar situation occurs for the ProPHET and Bubble Rap approaches, though the reduction trend of average cost and delay is relatively low compared with the Epidemic scheme.

E. Comparison of Three Schemes

Finally, we compare the *Ameba* scheme with the broadcast and unicast schemes given in Section VI-A.

Figures 5 (a-c) plot the results of the Infocom06 trace. It is observed that anti-correlated topics lead to less delivery ratio and larger average delay and cost than correlated topics. For example, for the *Ameba* scheme, the average cost with anti-correlated topics is decreased by 35.66%. This is because, if a topic is popular among mobile devices, the developed optimization policy proactively relay the advertisements of such topics with a larger number of opportunistic encounters. Then, such opportunistic encounters benefit a larger number of content items (due to the correlated topics). It obviously leads to a higher delivery ratio and less average delay and cost.

Among the three schemes, the *Ameba* scheme achieves the least average cost and comparable delivery ratio and average delay as the broadcast scheme. It is because the *Ameba* scheme combines the befits of both unicast and broadcast schemes. In addition, the unicast and broadcast schemes in this figure use less delivery ratio than the ProPHET and Epidemic schemes, respectively. It is because the optimal encounters given by the developed optimization strategy can improve the delivery of the popular advertisements, which are highly demanded by the majority of mobile devices.

For the MIT reality data set, Figures 5 (d-f) show the similar trend as Figures 5 (a-c). The *Ameba* scheme uses the least cost to achieve the comparable delivery ratio.

VIII. Conclusion

To timely deliver content over a DTN, *Ameba* carefully adjusts the number of encounters and content copies for advertised content, computes the utilities with low maintenance cost to capture interests and mobility patterns of mobile devices, and develops a distributed relay algorithm to select the best nodes as the carriers. Our experiments demonstrate that the proposed *Ameba* scheme is able to achieve high delivery ratio and significantly low overhead.

Acknowledgment: This work is partially supported by Academy of Finland (Grant No. 139144), and we would like to thank the anonymous SECON 2012 reviewers for their valuable comments. Part of the research was also conducted in the Internet of Things program of Tivit (Finnish Strategic Centre for Science, Technology and Innovation in the field of ICT), funded by Tekes.

References

[1] C. Boldrini, M. Conti, and A. Passarella. Modelling data dissemination in opportunistic networks. In *Proceedings of the third ACM workshop on Challenged networks*, CHANTS '08, New York, NY, USA, 2008. ACM.

[2] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web caching and zipf-like distributions: Evidence and implications. In *INFOCOM*, pages 126–134, 1999.

[3] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *INFOCOM*, 2006.

[4] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Trans. Mob. Comput.*, 6(6):606–620, 2007.

[5] A. Chaintreau, A. Mtibaa, L. Massoulié, and C. Diot. The diameter of opportunistic mobile networks. In *CoNEXT*, page 12, 2007.

[6] G. Chockler, R. Melamed, Y. Tock, and R. Vitenberg. Constructing scalable overlays for pub-sub with many topics. In *PODC*, 2007.

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

[8] N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, 2006.

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

[10] B. Han, P. Hui, V. A. Kumar, M. V. Marathe, J. Shao, and A. Srinivasan. Mobile data offloading through opportunistic communications and social participation. *IEEE Trans. Mob. Comput.*, 99(PrePrints), 2011.

[11] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. In *MobiHoc*, pages 241–250, 2008.

[12] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. Braynard. Networking named content. In *CoNEXT*, pages 1–12, 2009.

[13] P. Jokela, A. Zahemszky, C. E. Rothenberg, S. Arianfar, and P. Nikander. Lipsin: line speed publish/subscribe inter-networking. In *SIGCOMM*, pages 195–206, 2009.

[14] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica. A data-oriented (and beyond) network architecture. In *SIGCOMM*, pages 181–192, 2007.

[15] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. *Mobile Computing and Communications Review*, 7(3):19–20, 2003.

[16] H. Liu, V. Ramasubramanian, and E. G. Sirer. Client behavior and feed characteristics of rss, a publish-subscribe system for web micronews. In *Internet Measurment Conference*, pages 29–34, 2005.

[17] M. Motani, V. Srinivasan, and P. Nuggehalli. Peoplenet: engineering a wireless virtual social network. In *MOBICOM*, 2005.

[18] W. Rao, L. Chen, and A. W. Fu. On efficient content matching in distributed pub/sub systems. In *INFOCOM*, 2009.

[19] W. Rao, L. Chen, A. W.-C. Fu, and Y. Bu. Optimal proactive caching in peer-to-peer network: analysis and application. In *CIKM*, pages 663–672, 2007.

[20] W. Rao, L. Chen, A. W.-C. Fu, and G. Wang. Optimal resource placement in structured peer-to-peer networks. *IEEE Trans. Parallel Distrib. Syst.*, 21(7):1011–1026, 2010.

[21] W. Rao, K. Zhao, and S. Tarkoma˙ Maximizing timely content advertising in delay tolerant networks. In *Technical Report of the Computer Science Department, University of Helsinki*, 2011.

[22] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *WDTN*, 2005.

[23] A. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. In *Duke Technical Report CS-2000-06*, July 2000.