Towards Maximizing Timely Content Delivery in Delay Tolerant Networks

Weixiong Rao, Kai Zhao, Yan Zhang, Pan Hui, and Sasu Tarkoma, Senior Member, IEEE

Abstract—Many applications, such as product promotion advertisement and traffic congestion notification, benefit from 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-art capture mobility patterns or social properties of mobile devices. Such solutions do not capture patterns of delivered content in order to optimize content delivery. 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, leading to traffic congestion and packet drop. To address the challenge, in this paper, we develop a solution framework, namely Ameba, for timely delivery. In detail, we first leverage content properties to derive an optimal routing hop count of each content to maximize the number of needed nodes. Next, we develop node utilities to capture interests, capacity and locations of mobile devices. Finally, the distributed forwarding scheme leverages the optimal routing hop count and node utilities to deliver content towards the needed nodes in a timely manner. Illustrative results verify that Ameba achieves comparable delivery ratio as Epidemic but with much lower overhead.

Index Terms—Information dissemination, delay tolerant networks, opportunistic communication, experiments

1 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). 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 [15]. Opportunistic communication thus helps to expand the network coverage, without building dedicated network infrastructures.

An important requirement associated with exchanged content is freshness. That is, besides successfully delivering the content to appropriate users, we further expect the content to be delivered in a timely fashion. For example, the timely delivery of certain product promotion advertisement is a critical requirement, 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 essentially conflicts with the freshness requirement. Service providers expect to maximize the number of potential users who can receive content in a timely manner, and hence the items must be delivered on 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 [6].

In this paper, we study the problem of maximizing the number of users who can receive content in a timely manner. In detail, we are inspired by the topic-based model [8] to offer personalized content subscription. This model is widely used in many applications (e.g., RSS feeds, online games) to decouple content producers and consumers. The consumers, namely 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 the advertisement and filter contain the same topic.

The state-of-the-art systems [10], [16], [21], [24] leveraged mobility patterns or social properties of mobile devices to optimize content delivery. However, such works do not capture the patterns of delivered content for the content delivery. Without such optimization, the content demanded by a large number of subscribers could follow the same forwarding path as the case with only one subscriber. This obviously leads to traffic congestion and packet drop on the forwarding path. Moreover, many real applications show that the number of subscribers frequently exhibits the well-known Zipf distribution [3], [22]. This further aggravates the above-mentioned issue.

To solve the above maximization problem, in this paper, we propose a solution framework, namely Ameba, by considering two subproblems and designing the associated techniques. First, Ameba considers a simple case by assuming that each node in a DTN has an equal probability to...
forward a given advertisement towards needed subscribers (i.e., ignoring the constraint of mobility pattern and limited capacity). In this way, Ameba develops a strategy to assign an optimal hop count for published content. To solve this subproblem, Ameba leverages the distribution of content and assigns a larger hop encounter for the highly popular content demanded by more subscribers. In this way, more nodes act as intermediate carriers of the popular content, and subscribers have more chance to receive the advertisement in a timely manner.

Next, Ameba solves the general maximization problem, where node capacity is heterogenous and mobility pattern is further considered. To this end, Ameba develops a metric, namely the forwarding utility, to identify (i) which nodes are interested in the advertisement and (ii) how fast the encountered node can forward the advertisement towards subscribers. Based on the developed utility, Ameba selects the best carriers to forward the advertisement, and adaptively creates the copies of an advertisement for timely delivery.

As a summary, we make the following contributions.

- We develop a forwarding strategy to design an optimal hop count for each content. The content with such a hop count is expected to reach the needed nodes.
- By the developed utilities to capture the interests, mobility patterns and resource capacity of mobile devices, we design an adaptive algorithm to select the best carriers for timely delivery.
- By analyzing the locations of mobile devices, we extend the utility function to optimize the selection of the best carrier, and further improve the forwarding algorithm.
- We extensively evaluate the efficiency of the Ameba over three realistic data sets and verify that Ameba is able to achieve a delivery ratio comparable with Epidemic but with much lower overhead.

The remainder of this paper is organized as follows. Section 2 first defines the problem and gives a solution overview. Section 3 derives optimal encounters for advertisements, and Section 4 develops the utilities. Section 5 next gives the distributed forwarding scheme Ameba, and Section 6 designs an enhanced Ameba. After that, Section 7 evaluates the Ameba solution, and Section 8 investigates related works. Finally Section 9 concludes the paper.

## 2 Problem Statement and Solution Overview

### 2.1 Problem Statement

We treat DTNs as complementary network communication technologies [15]. They help content providers deliver the content (generated by the content providers) to needed nodes and explore more potential customers. Suppose that $N$ opportunistically encountered devices are connected as a DTN. The roles of mobile devices can be publishers (sources), subscribers (destinations) or intermediate carriers. Publishers publish advertisements of specific topics. Subscribers register subscription filters (containing defined topics) to receive the needed advertisements. In addition to publishers and subscribers, mobile devices can act as carriers to relay advertisements. Mobile devices are typically equipped with short range interfaces (e.g., Bluetooth or Wi-Fi) to detect and communicate with each other. When mobile devices encounter each other, the advertisements are exchanged opportunistically, and relayed from the publishers to the subscribers with the help of the 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 $p_i = N_i/N$. 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 $P$ and a DTN consisting of $N$ mobile devices, we want to design an advertisement relay scheme to maximize the number of nodes which can receive the matching advertisements.*

The above problem implicitly defines two constraints. (i) The exchange of advertisements among mobile devices incurs resource consumption (e.g., energy). Due to the limited capacity (e.g., battery power), each node allows only a fixed number of exchanges with other nodes. Moreover, mobile nodes typically follow some movement pattern. A given period $P$ consequently indicates that $N$ mobile devices experience a certain number of encounters and an associated number of advertisement exchanges (because an exchange occurs if and only if two nodes encounter each other). (ii) A matching advertisement means that a subscriber defines a filter having the same topics as the advertisement. Thus, an advertisement matches a subscriber node if and only if the defined filter shares the same topic as the advertisement. The second constraint is related to the application requirement of the DTN. With such constraints, the objective of Problem 1 is to maximize the number of subscribers receiving matching advertisements.

### 2.2 Solution Overview

Neither broadcast nor unicast works efficiently for Problem 1. That is, broadcast may blindly forward advertisements to unwanted subscribers and incur high overhead (e.g., energy consumption). Next, unicast may not forward advertisements in a timely manner to subscribers, and the number of subscribers to receive matching advertisements is low.

The proposed Ameba essentially is a hybrid of multicast and unicast, and we highlight its basic idea as follows. After an advertisement is published by its publisher, the advertisement is forwarded from the publisher to needed subscribers. When a current carrier node of the advertisement encounters another node, Ameba adaptively decides to (i) relay the advertisement from the current node to the encountered node, or (ii) keep the advertisement still on the current node without any movement, or (iii) create a copy of the advertisement onto the encountered node, and the
current and encountered nodes both act as the carriers. Thus, if creating multiple copies of the advertisement in the DTN, *Ameba* forwards the advertisement in a multicast manner; and if only one copy is created, *Ameba* forwards the advertisement in a unicast manner. In a general case, *Ameba* adaptively creates the number of copies and allows multicast or (and) unicast to forward the advertisement and its copies towards the subscribers.

To enable the above idea, we divide Problem 1 into two subproblems (Problem 2 in Section 3 and Problem 3 in Section 5) and develop the following techniques:

(i) An optimal number of opportunistic encounters for each advertisement (Section 3): We first consider a simple case by assuming that each node in a DTN has the equal probability to forward a given advertisement towards the needed subscribers (i.e., we ignore the mobility pattern and resource capacity in each node). In this way, we focus on the needed hop count $X_i$ to forward the advertisement (together with its copies if any) from its publisher to needed subscribers. To derive the optimal $X_i$, based on the distribution of content supplies and demands, *Ameba* proactively assigns a larger $X_i$ for a popular advertisement demanded by more subscribers. Thus, more nodes act as intermediate carriers of the popular advertisement, and more subscribers have a chance to receive the needed advertisement.

(ii) A forwarding utility-based distributed forwarding algorithm: given the general case, i.e., by considering the heterogamous node capacity and node mobility pattern, we focus on how to find appropriate carrier nodes and how to forward an advertisement in a timely manner towards subscribers. To this end, we develop a metric, namely the forwarding utility (Section 4), to identify (i) which nodes are interested in the advertisement and (ii) how fast the encountered node can forward the advertisement towards subscribers. Next, based on the developed utility, *Ameba* selects the best carrier and adaptively creates the copies of an advertisement for timely delivery.

### 2.3 Data Sets

Throughout this paper, we use three real-world data sets of human mobility traces to motivate and demonstrate the efficiency of our solution. (i) The Infocom '06 data set [7] contains opportunistic Bluetooth contacts between 98 iMotes, 78 of which were distributed to Infocom '06 participants and 20 of which were static. (ii) The MIT Reality trace [11] comprises 95 participants carrying GSM enabled cellphones over a period of nine months. (iii) In the UCSD data set [23], 274 WiFi-enabled PDAs were respectively used by 274 freshmen to log nearby Access Points (APs) for an 11-week period between Sept. 22, 2002 and Dec. 8, 2002 and a contact was recorded when two devices are associated to the same AP. Besides, we use the location information of phones (e.g., the GSM cellular tower in the MIT reality trace) for enhancement (Section 6).

### 3 Optimal Advertisement Relay Strategy

In this section, we first consider a simple case by assuming that each node in a DTN has an equal capacity to forward a given advertisement towards needed subscribers. Based on this assumption, we are interested in the optimal hop count to forward advertisements towards needed subscribers. In the next section, we will design a solution for a general case that the nodes are associated with heterogenous capacities.

#### 3.1 Overview

The general idea of our optimization is as follows. First, the exchanges of advertisements occur if and only if the associated mobile nodes carrying such advertisements encounter each other. Thus, the forwarding of advertisements from publishers towards needed subscribers depends upon the opportunistic encounters of mobile nodes. When the period $P$ is given, the mobile nodes in a DTN experience the total number $X$ of opportunistic encounters. Therefore, we treat the number $X$ as a constraint in this section to maximize the number of subscribers receiving advertisements of interest.

During the optimization, the key is to derive the function between the opportunistic encounters of mobile nodes used to exchange advertisements and the number of nodes (i.e., subscribers) to receive advertisements of interest. To this end, we first define the event that two nodes encounter as an *encounter event*, no matter what the mobility pattern and social properties of mobile devices. For example, given three nodes $n_i$, $n_j$, and $n_f$, the node $n_i$ first encounters $n_f$ after one day, and then encounters $n_f$ after only 1 minute. We treat the occurrence of the two events equally, because the occurrence of such events depends upon whether or not two nodes encounter, rather than the experienced period.

Based on the encounter events, we derive a function $f(\cdot)$ between the number of encounters to relay an advertisement and the number of nodes successfully receiving the advertisements of interest (Section 3.2). Next, considering the overall advertisements, we develop a strategy to assign an optimal number of encounters for each advertisement, such that we maximize the total number of nodes which can successfully receive matching advertisements (Section 3.3).

Essentially, this section exploits the properties of advertisements (i.e., the demanding and supplying rates), and do not consider the mobility pattern and capacity limit of DTN devices. We will utilize such a constraint and pattern to develop the node utilities for timely delivery (Section 4). Table 1 summarizes the main symbols used in this section.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>Total number of topics</td>
</tr>
<tr>
<td>$M$</td>
<td>Total number of advertisements</td>
</tr>
<tr>
<td>$N$</td>
<td>Total number of nodes</td>
</tr>
<tr>
<td>$p_i$</td>
<td>Demanding rate of a topic $t_i$ $(1 \leq i \leq T)$</td>
</tr>
<tr>
<td>$q_i$</td>
<td>Supplying rate of a topic $t_i$ $(1 \leq i \leq T)$</td>
</tr>
<tr>
<td>$d_i$</td>
<td>An advertisement associated with a topic $t_i$ $(1 \leq i \leq T)$</td>
</tr>
<tr>
<td>$N_i$</td>
<td>the set of nodes interested in $t_i$, with the cardinality $N_i$</td>
</tr>
<tr>
<td>$H_i$</td>
<td>Expected number of nodes which successfully receive $d_i$ to the needed nodes</td>
</tr>
<tr>
<td>$X_i$</td>
<td>Expected number of required encounters to relay $d_i$ to the needed nodes</td>
</tr>
<tr>
<td>$X$</td>
<td>Expected number of overall opportunistic encounters to relay advertisements</td>
</tr>
</tbody>
</table>

### 2.3 Data Sets

Throughout this paper, we use three real-world data sets of human mobility traces to motivate and demonstrate the efficiency of our solution. (i) The Infocom '06 data set [7] contains opportunistic Bluetooth contacts between 98 iMotes, 78 of which were distributed to Infocom '06 participants and 20 of which were static. (ii) The MIT Reality trace [11] comprises 95 participants carrying GSM enabled cellphones over a period of nine months. (iii) In the UCSD data set [23], 274 WiFi-enabled PDAs were respectively used by 274 freshmen to log nearby Access Points (APs) for an 11-week period between Sept. 22, 2002 and Dec. 8, 2002 and a contact was recorded when two devices are associated to the same AP. Besides, we use the location information of phones (e.g., the GSM cellular tower in the MIT reality trace) for enhancement (Section 6).

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>Total number of topics</td>
</tr>
<tr>
<td>$M$</td>
<td>Total number of advertisements</td>
</tr>
<tr>
<td>$N$</td>
<td>Total number of nodes</td>
</tr>
<tr>
<td>$p_i$</td>
<td>Demanding rate of a topic $t_i$ $(1 \leq i \leq T)$</td>
</tr>
<tr>
<td>$q_i$</td>
<td>Supplying rate of a topic $t_i$ $(1 \leq i \leq T)$</td>
</tr>
<tr>
<td>$d_i$</td>
<td>An advertisement associated with a topic $t_i$ $(1 \leq i \leq T)$</td>
</tr>
<tr>
<td>$N_i$</td>
<td>the set of nodes interested in $t_i$, with the cardinality $N_i$</td>
</tr>
<tr>
<td>$H_i$</td>
<td>Expected number of nodes which successfully receive $d_i$ to the needed nodes</td>
</tr>
<tr>
<td>$X_i$</td>
<td>Expected number of required encounters to relay $d_i$ to the needed nodes</td>
</tr>
<tr>
<td>$X$</td>
<td>Expected number of overall opportunistic encounters to relay advertisements</td>
</tr>
</tbody>
</table>
3.2 Probabilistic Advertisement Relay

Given a topic \( t_i \), we have \( N_i = p_i \cdot N \) subscribers interested in the advertisements with the topic \( t_i \), and denote \( N_i' \) to be the set of such subscribers. When all nodes encounter with each other, for a specific encounter event, we are interested in the probability \( \rho_i \) that such an encounter successfully relays an advertisement \( d_i \) to any a node inside \( N_i' \). Suppose a node carrying \( d_i \) is opportunistically encountering another node, which could be a member node inside \( N_i' \), or not. Among the nodes in the DTN, we have the number \( N_i \) of nodes that are inside the set \( N_i' \). Thus, the probability \( \rho_i \) is \( \frac{N_i}{N} \), because all encounter events are independent and further irrelevant to the mobility patterns and social properties of mobile devices, etc.

Based on the above discussion, the probability \( \rho_i^1 \) of the first new node inside \( N_i' \) to successfully receive \( d_i \) relayed from any node carrying \( d_i \) can be easily computed by \( \frac{N_i}{N} \). It is because any node inside \( N_i' \) could be the node to receive \( d_i \).

Next, the probability \( \rho_i^2 \) of the second new node in \( N_i' \) to receive \( d_i \) can be computed by \( \frac{N_i}{N} \cdot \frac{1}{N} \cdot (1 - \frac{N_i}{N}) \), where (i) \( \frac{N_i}{N} \) is the probability of any node inside \( N_i' \) to successfully receive \( d_i \), and (ii) \( (1 - \frac{N_i}{N}) \) is the probability that among the nodes \( N_i' \), the remaining nodes except the first new node receive \( d_i \). The reduction of \( 1/N_i \) avoids duplicately forwarding \( d_i \) to the first new node that has already received \( d_i \).

Similarly, the probability \( \rho_i^j \) of the jth new node inside \( N_i' \) to receive \( d_i \) is

\[
\rho_i^j = \frac{N_i}{N} \left( 1 - \frac{1}{N_i} \right). \quad (1)
\]

In Equation (1), the item \( (1 - \frac{1}{N_i}) \) represents the probability that any of the nodes inside the set \( N_i' \) (except the previous \( (j-1) \) nodes) can receive \( d_i \). Given the probability \( \rho_i^j \), the trial that the jth new node inside \( N_i' \), receives \( d_i \) can be treated as a geometric probability distribution with the parameter \( \rho_i^j \). Then, the expected number of opportunistic encounters needed to successfully relay \( d_i \) to the jth new node is

\[
X_i^j = \frac{1}{\rho_i^j} = \frac{N}{N_i} \cdot \frac{N_i}{N_i - j + 1} = \frac{N}{N_i - j + 1}. \quad (2)
\]

The above equation indicates that the expected number \( X_i^j \) of encounters are needed to ensure that the jth nodes inside \( N_i' \) will receive \( d_i \). Now, let us consider a reverse problem as follows. Given the \( X_i^j \) encounters, we are interested in how many nodes inside \( N_i' \), successfully receive \( d_i \). That is, we want to derive the expected number \( H_i \) of nodes inside \( N_i' \) that can successfully receive \( d_i \), as follows.

Based on Equation (2), the 1st, 2nd… jth new node respectively receives \( d_i \) when the expected number \( X_i^1, X_i^2, \ldots X_i^j \) of encounters occur. Thus, if all such \( j \) nodes receive \( d_i \), we need a total of \( (X_i^1 + X_i^2 + \cdots + X_i^j) \) encounters. Following the idea, given the number \( X_i \) of encounters, the expected number \( H_i \) of nodes to successfully receive \( d_i \) is computed by

\[
\sum_{j=1}^{H_i} X_i^j = \sum_{j=1}^{H_i} \frac{N}{N_i - j + 1} = X_i. \quad (3)
\]

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

\[
\sum_{j=1}^{H_i} \frac{N}{N_i - j + 1} = N \cdot \left( \sum_{j=1}^{H_i} \frac{1}{N_i - j + 1} \right) \approx N \cdot \ln \frac{N_i}{N_i - H_i}. \quad (4)
\]

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 \( \sum_{j=1}^{N_i-H_i} \frac{1}{j} \) which 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.

\[
H_i = N \cdot p_i \cdot (1 - e^{-\frac{X_i}{N}}). \quad (5)
\]

**Proof.** Based on Equations (2), (3), and (4), we obtain

\[
X_i = N \cdot \ln \frac{N_i}{N_i - H_i}
\]

and then validate the correctness of the theorem.

\[ \square \]

3.3 Optimal Relay Strategy

In this section, given the overall advertisements having \( T \) topics, we develop an overall strategy to assign an optimal number \( X_i \) of encounters for each advertisement. The optimization objective is to maximize the number of subscribers 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 \( d_i \). Next, among all \( M \) distinct advertisements, given the supplying rate \( q_i \cdot M \) advertisements are associated with the topic \( t_i \). Based on the assumption in this section, we know that any node has an equal chance to independently publish an advertisement. Thus, the number \( q_i \cdot M \cdot X_i \) of opportunistic encounters are needed, such that \( q_i \cdot M \cdot H_i \) nodes successfully receive needed advertisements.

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

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

Our objective is to maximize the above \( H \). On the other hand, given a given time period \( P \), we have the constraint of the totally \( X \) opportunistic encounters, i.e., \( \sum_{i=1}^{T} (q_i \cdot M \cdot X_i) = X \). Here, for the advertisements of each topic \( t_i \), we need the \( (q_i \cdot M \cdot X_i) \) number of encounters, and the total number of encounters is computed by \( \sum_{i=1}^{T} (q_i \cdot M \cdot X_i) \).

Given the above objective and the constraint, we define the following optimization problem:

**Problem 2.** Given the constraint \( \sum_{i=1}^{T} (q_i \cdot M \cdot X_i) = X \), we want to maximize the total number \( H \) of nodes receiving the matching advertisements.

To solve the above problem, we have the theorem:

**Theorem 2.** The total number \( H \) in Problem 2 is maximized when \( X_i = \frac{X}{M} + N \ln p_i - N \sum_{i=1}^{T} (q_i \ln p_i) \).
Proof. We use the Lagrange multiplier method to obtain the optimal $X_i$ in terms of $p_i$ and $q_i$. First, we find the Lagrange multiplier $\lambda$ that satisfies $\nabla X = \lambda \cdot \nabla c$, where $c = \sum_{i=1}^{T} (q_i \cdot X_i \cdot M) - X = 0$. Next, treating $p_i$, $q_i$, $X$, $M$ and $N$ as the constants,

$$\nabla X = M \cdot N \cdot \sum_{i=1}^{T} \left( p_i \cdot q_i \cdot e^{-\frac{X_i}{N}} \cdot \frac{1}{N} \right) \cdot \hat{u}_i,$$

where $\hat{u}_i$ is a unit vector. Next,

$$\lambda \cdot \nabla c = \sum_{i=1}^{T} (\lambda \cdot q_i \cdot M) \cdot \hat{u}_i.$$  \hfill (7)

Since $\nabla X = \lambda \cdot \nabla c$, then

$$\lambda \cdot q_i \cdot M = M \cdot N \cdot p_i \cdot q_i \cdot e^{-\frac{X_i}{N}} \cdot \frac{1}{N}. \hfill (8)$$

Solving for $X_i$ gives

$$X_i = -N \cdot \ln \frac{\lambda}{p_i}. \hfill (9)$$

Substituting the above equation to $c = \sum_{i=1}^{T} (q_i \cdot X_i \cdot M) - X = 0$ gives

$$-\ln \lambda = \frac{X}{MN} - \sum_{i=1}^{T} (q_i \cdot \ln p_i). \hfill (10)$$

We substitute the above result pertaining to $-\ln \lambda$ back to Equation (9) and arrive at an optimal $X_i$:

$$X_i = \frac{X}{M} + N \ln p_i - N \sum_{i=1}^{T} (q_i \cdot \ln p_i). \hfill (11)$$

Following the equation above, we arrive at Theorem 2. \end{proof}

Based on Theorem 2, we have the following discussion.

(i) 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 $X_i$. That is, if $d_i$ is demanded by more nodes, the relay strategy assigns a larger $X_i$, such that more nodes act as the carriers to relay $d_i$ and the nodes carrying $d_i$ encounter more nodes inside $N_i$. (ii) 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, $-N \sum_{i=1}^{T} (q_i \cdot \ln p_i)$ and $X_i$ become smaller. This means that the skewed distribution of $p_i$ and $q_i$ assigns a small $X_i$ to relay advertisements towards the nodes $N_i$. Many real applications indicate a skewed distribution of $p_i$ and $q_i$ [8], [27], which favors the optimization strategy.

3.4 Practical Relay Strategy

First, when the parameters $p_i$, $q_i$, etc., are known (we will give two approaches to acquire the parameters), we can solve $X_i$ with numeric values. After that, the classic rounding solutions (such as random rounding) can be applied to derive integer values. Now, our problem becomes how to acquire the parameters $p_i$, $q_i$, and we design the two following approaches.

First, as a centralized solution, we acquire the parameters from content providers. It is feasible since the content is generated by content providers and the providers can easily compute the parameters regarding the advertisement such as $M$ and $q_i$. Meanwhile, since the DTN is a complementary network communication technology, the mobile devices, as subscribers, typically register subscription filters to a registration server which is hosted by the content providers. Thus, the parameters regarding the filters and mobile devices, such as $p_i$ and $N$, are available to the content providers.

Second, in case that the centralized solution does not work, we follow [1] to estimate the parameters by $O(\log N)$ rounds of histogram exchanges. That is, each node first maintains a local histogram (e.g., an equi-width histogram) to capture the distribution of the parameters. Next, via a gossip style, each node $n_k$ exchanges its local histogram with every encountered node $n_j$. We call the process of every node exchanging and merging the histogram with an encountered node one round. After $O(\log N)$ rounds, the nodes of the DTN can capture the overall distribution, and the histogram of each node becomes stable. Based on the histogram, we accumulate the buckets in the histogram to approximate the parameters.

To evaluate the accuracy of the histogram sampling, we compute the histogram error $\varepsilon$ by $\varepsilon = \frac{\sum_{i=1}^{N} (S_i - T_i)}{\sum_{i=1}^{T_i}}$, where $S$ is the sampling histogram and $T$ is the histogram based on real data. Take node count estimation as an example. Fig. 1 plots the average accuracy of the histogram, when the number of exchanging rounds are varied. The error $\varepsilon$ (i.e., the $y$-axis in the figure) decreases rapidly when the number of rounds sampled per node increases. It is interesting to find that the accuracy of sampling algorithm in the MIT Reality data set is much higher than the two other traces. The reason is that in MIT Reality trace (the campus environment), the encounters event is more sparse and uniform compared to the Infocom '06 and UCSD data sets. As a result, the meeting node is less likely to encounter the same node again in an adjacent round, which will decrease the meaningless re-sampling of one single node in a short time. Despite the environment and the network size, the results of both traces show that our distributed approximate histogram is able to reduce the error to an acceptable ratio, say, 15 percent in $2 \cdot \log_2 N$ rounds.
4 Relay Utilities

In this section, we develop a forwarding utility for the general case where the interests, mobility pattern and capacity constraint of DTN nodes are considered. The utility measures the goodness of a node to relay advertisements to subscribers.

4.1 Data Structure

For a node \( n_t \), we use a vector \( U^j \) (called vector utility) to represent \( n_t \)'s utilities. \( U^j \) contains \( T \) elements (corresponding to the \( T \) topics), and the elements capture mobile devices' interests of the associated topics. Each element \( u^j_i \in U^j \) \( (1 \leq i \leq T) \), called element utility, is computed as a number inside the range \([0.0, 1.0]\). \( u^j_i \) measures the goodness of \( n_t \) to successfully relay an advertisement \( d_i \) (with a topic \( t_i \)) towards the nodes in \( N' \). A larger \( u^j_i \) indicates that \( n_t \) has more chance to relay \( d_i \) successfully to the nodes in \( N' \).

The proposed utility has 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 that node \( n_t \) itself is interested in. This property helps to identify \( n_t \)'s interest. Suppose that the utility \( U^j \) of \( n_t \) is available to a node \( n_k \) (which might not encounter the node \( n_t \)). By finding the largest element inside \( U^j \), \( n_k \) then identifies that \( n_t \) is self-interested in the associated topic. The key point is that even before \( n_k \) encounters \( n_t \), the interest of \( n_t \) is available to \( n_k \). This property is useful for Ameba to relay \( d_i \) in a timely manner towards the needed nodes.

- The elements inside \( U^j \) accumulate the overall interests of the whole nodes in the DTN. That is, if a topic \( t_i \) is popular among the DTN, the associated element \( u^j_i \) is larger. Thus, the element \( u^j_i \) directly captures the overall distribution of the demands of topic \( t_i \).

In addition, the utility \( U^j \) incorporates the interests, mobility pattern (e.g., encountering rate) and capacity constraint belonging to the node \( n_t \). Such a pattern is also helpful for the timely delivery.

In Fig. 2, the leftmost column shows an example utility vector of a node \( n_t \). Then, we can identify that \( n_t \) is (itself) interested in \( t_A \) (because the element of \( t_A \) has the largest value 0.614). In addition, since topic \( t_B \) is regularly registered in three other nodes \((n_2, n_3 \) and\( n_4))\), the associated element utility 0.271 is larger than the two remaining ones \((i.e., 0.071 \) and \( 0.042))\).

After describing the data structure of \( U^j \), we compute \( u^j_i \) based on the following information.

- For the self-interest of the node \( n_t \) itself, we use a 0/1 vector with \( T \) elements \( s^j_i \), e.g., the second column in Fig. 2, to represent the self-interest. The element \( s^j_i \) regarding \( t_i \) is equal to 1, if the node \( n_t \) is registered with a filter defining \( t_i \).

- The utilities \( U^k \) of the nodes (say \( n_k \)) that \( n_t \) previously encountered within a specific period \( P \). When \( n_t \) and \( n_k \) encounter each other, the utilities \( U^k \) (resp. \( U^j \)) are available to \( n_k \) (resp. \( n_t \)) by the opportunistic exchange.

- The encountering frequency rate \( f_{j,k} \) incorporates the mobility pattern and capacity constraint of the mobile nodes \( n_t \) and \( n_k \). It indicates the chance that the two nodes exchange advertisements.

In Fig. 2, the second leftmost column indicates \( n_j \)'s self-interest, and the three columns from the third leftmost column to the rightmost column show the three encountered nodes \((n_2, n_3 \) and \( n_4))\), frequency \((1, 4 \) and \( 2))\) and the associated element utilities, respectively. The encountering frequency is the number of encounters between \( n_t \) and the three nodes inside a period. Following [10], the nodes that ever frequently encountered are highly likely to encounter again in the future. Thus, \( f_{j,k} \) indicates the chance that \( n_t \) and \( n_k \) will encounter each other and then exchange advertisements.

Though the computation of \( U^j \) does need the above information, in the following section, we will show that it is unnecessary for \( n_t \) to maintain the detail of \( u^j_i \) and \( f_{j,k} \) for every previously encountered node \( n_k \). Thus, we avoid the expensive maintenance overhead (e.g., the space cost), particularly when \( n_t \) encountered a large number of nodes.

4.2 Steps to Compute Node Utilities

Suppose that \( n_t \) previously encountered \( K \) nodes \( n_k \) \( (1 \leq k \leq K) \) inside a time window (e.g., a sliding window of 6 hours, typically smaller than the given period \( P \)). The computation of \( U^j \) captures the overall interests in DTN by accumulation, and the self-interests of \( n_t \) by normalization. In detail, the node \( n_t \) uses the following four steps to compute \( U^j \).

1) Overall accumulation. Given the \( K \) nodes \( n_k \) that \( n_t \) encounters, we accumulate the overall utility \( U(i)^j \) by the sum operation. For an element \( u(i)^j \) of a topic \( t_i \), we compute \( u(i)^j = \sum_{k=1}^{K} f_{j,k} \cdot u^k_i \) of such \( K \) nodes \( n_k \). For example, for the topic \( t_A \) in Fig. 2, we compute \( u(i)^j = 1.6 = 1 \times 0.2 + 4 \times 0.1 + 2 \times 0.5 \).

Similarly, we compute the overall element utilities for the other three topics \( t_B, t_C \) and \( t_D \) as 3.8, 1.0 and 0.6, respectively.

2) Overall normalization. Based on the above accumulated utility \( U'(i)^j \), we next normalize it and each element \( u(i)^j \) is computed by \( u(i)^j = u'(i)^j / \sum_{i=1}^{T} u'(i)^j \). Following the above example, for a topic \( t_A \), its normalized overall utility is \( 1.6 / (1.6 + \cdots + 0.6) = 1.6 / 7.0 \).

3) Self accumulation. Based on the normalized overall utility \( U(i)^j \) and self-interest \( r^j_i \), we accumulate \( U(i)^j \)
and \( r^j_i \) to compute \( U^j_i \) by setting the element \( u^j_i = \frac{r^j_i + u(i)}{\sum_{i=1}^{n} r^j_i + \sum_{i=1}^{n} u(i)} = \frac{r^j_i + u(i)}{\sum_{i=1}^{n} r^j_i + 1.0} \). For example, given the topic \( t_{14} \), we have \( u^j_{t_4} = \frac{1.1 + 0.7}{1 + 1} \).

4) **Self normalization.** We finally normalize the utility \( U^j \) to \( U \) by \( u^j_i = u^j_i / \sum_{i=1}^{n} u^j_i \) (e.g., for \( t_{14}, u^j_{t_4} = 0.614 \)).

The above step 1 accumulates the utilities of the previously encountered nodes \( n_k \) to capture the overall interests of the nodes \( n_k \), such that \( n_j \) has a chance to act as a carrier to relay \( d_i \) from \( n_k \) towards the nodes \( N_i \) by intermediatingly going through \( n_j \). Next, step 3 accumulates the self-interests of \( n_j \) with the overall utilities. Thus, the accumulation thus helps to differentiate the self-interests of \( n_j \) and overall interests of the whole DTN.

In addition, given the self-interests with \( r^j_i = 1 \), the normalization of steps 2 and 4 ensures that among all elements in \( U^j \), the element \( u^j_i \) is associated with the largest value. This property helps to relay \( d_i \) towards needed nodes via the greedy forwarding scheme given in Section 5.

Finally, as mentioned before, it is unnecessary to maintain the detail of \( u^j_i \) and \( j_{n_k} \) for every previously encountered node \( n_k \). Instead, we only maintain the accumulated overall utility \( U^{'j} \) (see the above step 1). Whenever \( n_j \) encounters a node \( n_k \), the element \( u^{'j}_i \) of \( U^{'j} \) is incrementally updated by \( u^{'j}_i = u^{'j}_i + u^j_i \). After that, we still follow the above steps 2, 3, and 4 to renew the utility \( U \). Therefore, \( n_j \) only maintains the overall utility \( U^{'j} \), the self-interests \( r^{'j}_i \), and the final utility \( U^j \). Thus, without maintaining the utility \( U^k \) of every ever encountered node \( n_k \), the maintenance cost is trivial.

## 5 Distributed Forwarding Scheme

Based on the assigned opportunistic encounters \( X_i \) (given by Section 3) and the developed forwarding utilities (given by Section 4), we design a distributed forwarding algorithm *Ameba* in this section.

### 5.1 Alternative Solutions

Given the hop count \( X_i \) assigned for an advertisement \( d_i \), we first adapt two classic schemes, broadcast and unicast, to our problem. That is, an advertisement \( d_i \) (and its copies, if any) is associated with an accumulated TTL equal to \( X_i \). The relay of \( d_i \) starts from a publisher node \( n_1 \).

(i) **Broadcast.** Whenever the publisher node \( n_1 \) encounters a node \( n_{2}, d_i \) is copied to the node \( n_2 \). Then both \( n_1 \) and \( n_2 \) maintain the copies of \( d_i \), each of which is associated with the updated TTL \((X_i - 1)/2\), and independently act as the carriers of \( d_i \). 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.

(ii) **Unicast.** When \( n_1 \) encounters \( n_2, \) if \( u^j_i > u^{'j}_i \), we relay \( d_i \) from \( n_1 \) to \( n_2 \). The node \( n_2 \) then acts as a new carrier of \( d_i \), associated with the renewed TTL \((X_i - 1)\). The unicast is finished when the current TTL is equal to 0. Different from the broadcast scheme, the unicast scheme maintains only one copy of \( d_i \) during the whole relay process.

During the above schemes, the advertisement \( d_i \) after each exchange is updated with a new TTL counter (e.g., \((X_i - 1)/2\) in broadcast and \((X_i - 1)\) in unicast). The update of the TTL is to consider that each exchange of \( d_i \) consumes the resource (e.g., energy) of the two encountered nodes. When the TTL is updated to zero, *Ameba* stops the exchanges of \( d_i \) and no resource is then consumed. Given the constraints of Problem 1, the updated TTL thus helps to reduce the resource consumption of DTN nodes.

The above adapted schemes suffer from disadvantages. First, broadcast creates too many copies of advertisements. Such copies overly consume the limited resources of mobile nodes, and some subscribers consequently cannot receive the needed advertisements. On the other hand, unicast incurs high latency before \( d_i \) is successfully relayed to subscribers. The latency may be larger than the given time period \( P \), and the advertisements expire.

### 5.2 Problem Statement

*Ameba* in essence combines the benefits of the adapted unicast and broadcast schemes, and delivers the advertisement to the needed nodes as fast as possible.

Formally, we model a 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 \( n_k \) (\( k \neq j \)), there is an edge between \( n_j \) and \( n_k \). In the graph \( G \), each node \( n_j \) is associated with a utility vector \( U^j \). The weight \( w_{jk} \) of the edge between \( n_j \) and \( n_k \) depends upon the element utility \( u^j_i \) and the encountering frequency \( f_{jk} \) between two nodes \( n_j \) and \( n_k \). (see the detail of \( f_{jk} \) in Section 4.1). Larger values of \( u^j_i \) and \( f_{jk} \) indicate a higher chance of \( n_j \) to successfully relay \( d_i \) from \( n_j \) to \( n_k \). Thus, we can designate the edge between \( n_j \) and \( n_k \) with a weight \( w_{jk} = f_{jk} * u^j_i \). Next, we define the following problem.

**Problem 3.** Given a graph \( G \) and an advertisement \( d_i \), we want to find the number \( X_i \) of edges starting from a publisher \( n_j \), such that the sum of all such edge weights is maximized.

In the following sections, we will first prove that Problem 3 is NP-hard by reducing it from the set cover problem (Section 5.3), and then propose a heuristic algorithm (Section 5.4).

Note that the above problem considers only one advertisement \( d_i \), and can be intuitively treated as a special case of Problem 1. When we consider the general case with multiple advertisements, Problem 1 is NP-hard, unless \( P = NP \).

### 5.3 NP-Hard Problem

In order to show the NP-completeness of Problem 3, for any subset of \( n_j \)’s neighbors, we can easily check in polynomial time whether they are registered with the filters satisfied with \( d_i \). Next, we prove that the NP-complete Set Cover problem [14] can be reduced to the special case of Problem 3 in polynomial time. In such a special case, each edge is associated with an equal weight, and the number \( N \) of nodes are interested in the advertisement, where \( N \) is the total number of nodes in \( G \).

Based on the graph \( G \), we first build a new graph \( G' \) as follows. For any node \( n_j \in G \), we denote \( R(n_j) \) to be \( n_j \)’s neighbors, and create a copy of \( n_j \), denoted as \( n'j \). Then for each neighbor node \( n_k \in R(n_j) \), there is a corresponding
In Algorithm 1, the two encountering nodes \( n_j \) and \( n_k \) exchange and update their utilities \( U \) and \( U^k \) (Section 4 has given the details to compute the utilities). If the node \( n_j \) is interested in \( d_i \), the advertisement \( d_i \) is copied from \( t_i \) to \( n_j \), and \( n_j \) then receives the matching advertisement \( d_i \).

Next in line 3, if \( u_i^k \) (i.e., the element utility of node \( n_k \) in terms of topic \( t_i \)) is larger than \( u_i^j \), then \( n_j \) keeps a copy of \( d_i \) in the buffer (line 4). Otherwise, node \( n_k \) will remove \( d_i \) (line 7). The removal still occurs if one of the following cases occurs: (i) \( n_j \) is interested in \( t_i \), but \( n_k \) is not, (ii) neither \( n_j \) nor \( n_k \) is interested in \( t_i \), or (iii) both \( n_j \) and \( n_k \) are interested in \( t_i \). It means that \( n_k \) will remove \( d_i \) even if line 2 is executed (line 7). This makes sense because \( n_j \) is better than \( n_k \) to act as the carrier of \( d_i \) (due to \( u_i^k < u_i^j \)). For example, \( n_j \) is interested in \( t_i \), and none of its previously encountered nodes is interested in \( t_i \). Given this case, it is meaningless for \( n_k \) to still retain a copy of \( d_i \), and act as the carrier of \( d_i \).

After that, in lines 5 and 6, there is another option whether or not \( n_j \) itself still keeps a copy of \( d_i \) in the local buffer. In case that \( u_i^k \) is larger than \( u_i^j \) (where \( u_i^j \) is the largest utility of the topic \( t_i \) among all nodes that \( n_j \) ever encountered), node \( n_j \) deletes \( d_i \) from the local buffer and will not act as a carrier of \( d_i \). During the encounter between \( n_j \) and \( n_k \), \( u_i^j \) is updated if \( u_i^k \) is larger than the current \( u_i^j \).

In addition, Algorithm 1 updates the TTL of \( d_i \) as follows. Suppose that the current TTL of \( d_i \) is \( X_i \) (\( X_i \) is initially equal to \( X_0 \)). If both \( n_j \) and \( n_k \) keep a copy of \( d_i \) in their local buffers, the new TTL associated with the copy is equally set by \((X_i - 1)/2\). Otherwise, if either \( n_j \) or \( n_k \) keeps a copy of \( d_i \), the new TTL associated with this copy is \((X_i - 1)\). Finally, Ameba terminates the relay of \( d_i \) if the associated TTL is zero.

Based on Algorithm 1, those nodes with larger utilities \( u_i^j \) are selected as the carriers of \( d_i \). As a result, the advertisement \( d_i \) is relayed in a greedy fashion towards the nodes with larger element utilities (for example, those nodes interested in \( d_i \)). The greedy fashion helps forwarding \( d_i \) towards the needed nodes with high delivery ratio and significantly low overhead. That is, following the broadcast scheme, Algorithm 1 creates multiple copies of \( d_i \), and thus increases the delivery ratio; follows the unicast scheme, Algorithm 1 limits the number of such copies in lines 3, 4, 5, and 6, and avoids creating too many copies of advertisements and to select the best nodes in DTN for timely delivery. The relay of \( d_i \) starts from an initial publisher node \( n_j \). Next, considering the constraint that each node is associated with a limited resource (e.g., energy) inside the period \( P \), we assume that each node maintains a fixed number of advertisements in the local buffer.
copies. Our experiment verifies that Algorithm 1 achieves high delivery ratio and low overhead.

6 LOCATION-ASSISTED AMEBA ENHANCEMENT

In this section, we propose an Ameba enhancement. The basic idea is to leverage the location information of mobile devices and to improve the developed forwarding utility and distributed relay algorithm. Based on the observations of three mobile device trace files (Section 6.1), we then give the details of the enhancement (Section 6.2).

6.1 Trace-Based Study

Based on the three DTN trace files, we examine the population density of different areas (in short area density) in DTNs and exploit it for advertisement relay. The location information could be passively gathered by GPS, WiFi or other positioning solutions [9], [26].

The area density is defined as follows. When a node $n_i$ (e.g., a mobile phone carried by a user) visits an area $j$, we say one visit $v_{ij}$ occurs. The visit was logged by iMote in the Infocom '06 data set or by cellular tower in the MIT Reality data set, or logged by the APs in the UCSD data set. Based on the logged visits, we calculate the area density $p_j$ of an area $j$ by $p_j = \frac{\sum_{i=1}^{N} v_{ij}}{\sum_{i=1}^{N} \sum_{j=1}^{R} v_{ij}}$, where $N$ is the total number of nodes and $R$ is the total number of areas. By the definition $p_j$, an area $j$ becomes more dense, (i.e., a higher $p_j$), when more users visit the area $j$.

To compute $p_j$, we divide a map into areas. For the Infocom '06 data set, we divide the map (i.e., the Infocom '06 conference site) into nine areas in equal sizes. For the MIT Reality data set, based on the extracted time-stamped tower transition logs in the form of [nodeID, date, AreaID.cellID], we get 213 locations (i.e., AreaIDs). In the UCSD data set, we use the 287 APs to identify the associated location based on the time-stamped log sequence ([Time, UserID, AP]). In addition, we set the observation period to three days in the Infocom '06 data set, three weeks in the MIT Reality data set, and three weeks in the UCSD data set for fairness with the MIT reality trace. Based on the divided areas, we compute the area density of each area and plot the cumulative distribution (CDF) of the Infocom '06, Reality and UCSD data sets in Fig. 4.

Fig. 4 indicates that the visits of participants are highly clustered in some specific areas. In Infocom '06 data set, the top three hot areas among the total nine areas account for nearly 65 percent of total visits; while in the MIT reality data set, the top two areas (1 percent) account for 70 percent of the visits, and the CDF Figure indicates that the top eight areas (3 percent) account for 65.4 percent of total visits. These numbers indicate that the area density distribution is very heterogeneous with a very few hot areas and most rarely visited areas. The intuition behind this is as follows. In the Infocom '06 conference setting, the hot areas are reception on the first floor and demo room on the second floor. In the Reality data set, although it does not give the actual locations of all divided areas, it provides the actual locations of two hot areas, i.e., the MIT Media lab and Sloan business school. In the UCSD data set, these hot area APs were located at the student housing facility [23].

Next Fig. 5 plots the hourly visits of hot areas. The label ‘evening-1’ in the Infocom '06 figure means the evening of the first day from 18:00 to 24:00, ‘midnight-1’ means the midnight of the first day from 00:00 to 06:00. The label ‘Mon-1’ in both MIT Reality and UCSD figures represents Monday of the first week from 00:00 to 23:59. We define the hot areas to be those areas having the associated area density in a time period (e.g., ‘evening-1’ or ‘Mon-1’) at least 65 percent. From this figure, we find that these hot areas do not consistently follow stable curves over time. In the Infocom '06 conference settings, people tend to visit hot areas during evening and midnight. This is mainly because participants of the conference are more likely to check in or ask for help at nights. In the MIT reality data set, we observe that the students tend to visit their work place during weekdays. In the UCSD data set, the majority of first-year students stay at their student housing facility (hot area) during nights and weekends. They visited other locations such as lecture halls where a large number of undergraduate courses are taught during weekday noon.

Table 2 summarizes the average density of a hot area in peak time and normal time. The peak time of hot areas means the evening and midnight (from 18:00-00:00) in the Infocom '06 data set, weekdays (from Monday to Friday) in the MIT reality data set, and the nights (evening and midnight) in weekdays and all weekends (from Saturday to Sunday) in the UCSD data set. The rest periods are the normal time.

As a summary of the above observation, we draw the conclusion: (i) the participant visits exhibit strong spatial properties (that is, the participants frequently visit in a small number of hot areas, and rarely visit the remaining areas), and (ii) the participant visits also exhibit strong temporal properties (e.g., the majority of participant visits are clustered during some specific periods).

6.2 Ameba Enhancement

In this section, we leverage the above spatial and temporal properties of participant visits to improve Ameba.

| Table 2: Average Hot Area Density over Peak Time and Normal Time |
|-----------------|-------------|-------------|-------------|
| **Dataset**     | Infocom06   | Reality     | UCSD        |
| **Peak time**   | 0.84        | 0.74        | 0.66        |
| **Normal time** | 0.62        | 0.55        | 0.57        |
First, we improve the node utility (Section 4) by defining an encounter event: Two nodes encounter each other in the same area if their arrival time at such an area are within a specific period. For illustration we use one minute in the three data sets. The interval is chosen to be shorter than the device discovery interval in order to avoid the synchronous device discovery periods [25].

We use $f_{j,k}$ to predict the chance that two mobile devices $n_j$ and $n_k$ will encounter each other. We compute $f_{j,k}$ as follows: Depending upon the similarity of the locations that $n_j$ and $n_k$ have respectively visited, we compute $f_{j,k} = 1 - \sum_{i=1}^{C} |V_{ji} - V_{ki}|$, where $C$ is the total number of areas, and $V_{ji}$ indicates the percentage of user $u_i$’s visits at the area $i$.

The above $f_{j,k}$ captures the possibility that two nodes visit the same place in the future. Note that in Section 4, the parameter $f_{j,k}$ incorporates the mobility pattern and capacity constraint of $n_j$ and $n_k$. The normalized $f_{j,k}$ (the detail of the normalization refers to Section 4.2) implicitly indicates the chance that $n_j$ and $n_k$ will encounter each other in the future. The above $f_{j,k}$ then improves the $f_{j,k}$ in Section 4 by the spatial and temporal properties.

To clearly illustrate the intuition of $f_{j,k}$, we give an example. We consider $C = 5$ areas. A user $u_i$ visits the areas with 10, 10, 10, 20 and 50 times respectively, and another user $u_k$ visits the areas with 25, 0, 0, 20 and 5 times respectively. We then have $f_{j,k} = 1 - \frac{0.1 - 0.0 + 0.1 - 0.0 + 0.2 - 0.4 + 0.5 - 0.1}{10} = 0.4$. Based on this example, if $n_j$ and $n_k$ visit the same areas more frequently, we have higher $f_{j,k}$.

When the above $f_{j,k}$ is ready, we can follow Section 4.2 to compute the utility, and improve Algorithm 1.

### Algorithm 2: Enhanced Ameba

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

1. $n_j$ (resp. $n_k$) updates $U^i$ (resp. $U^j$) by exchanging utilities with $n_k$ (resp. $n_j$).
2. If $n_k$ is interested in $d_i$, or $V_{ji} > \lambda$ during peak time then
3. $d_i$ is forwarded to $n_k$. 
4. If the element utility $u_i$ is larger than the element utility $u_j$ then
5. $n_j$ keeps a copy of $d_i$;
6. if $u_i > u_j // u_j$ is the largest element utility of the topic $t_i$ among all nodes that $n_j$ ever encountered.
7. $n_j$ removes $d_i$;
8. else $n_j$ does not keep a copy of $d_i$;

The above algorithm only updates line 2 of the previous Algorithm 1 by adding another condition $V_{ji} > \lambda$ during peak time. Here $V_{ji}$ is the sum of the visit percentage $V_{jk}$ of user $u_i$’s visits at hot areas during the peak time. When $V_{jk}$ is higher, then $u_k$ visits the hot areas more frequently. The intuition of $V_{jk} > \lambda$ is as follows. When $u_k$ more frequently visits the hot areas (due to a higher $V_{jk}$), the node $u_k$, though not interested in $d_i$, could help to forward $d_i$ to the nodes that are interested in $d_i$. Such forwarding is useful because many users frequently encounter each other at the hot areas during peak time. By setting $\lambda=0.95$, our experimental results indicate the advantage of the enhanced Ameba over the original scheme (i.e., Algorithm 1 in Section 5).

### 7 Evaluation

#### 7.1 Experimental Settings

We compare Ameba with the Epidemic scheme [29] (i.e., the flooding-based approach), ProPHET [21] (i.e., the probabilistic approach), and Bubble Rap (i.e., the social aware approach). Note that for Bubble Rap, when a content item is needed by multiple subscribers and such subscribers are located at multiple communities, we then have to forward the item to all such communities. In addition, we compare Ameba with the adapted broadcast and unicast schemes with the optimal encounters, both of which are introduced in Section 5.

Next, we use the three data sets in Section 2.3 (i.e., Infocom ’06, MIT reality and UCSD traces) to simulate the mobility pattern of DTN nodes. Next, by the Zipf distribution, we generate filters and content items for a set of given topics [8]. In terms of 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 publishers among the DTN nodes.

Table 3 shows the parameters used in the experiments. Taking the Zipf parameter $\alpha$ as an example, we use 0.95 as the default value and the interval [0.0, 1.2] as the allowable range. For example, with the default $\alpha=0.95$, we generate filters and advertisements by the Zipf distribution as the following example results: the number of generated filters over the 1st topic... and the 10th one is respectively 4, 22, 3, 3, 6, 3, 14, 15 and 2 (totally 78 filters equal to the node count) corresponding to the demanding rate $p_1 = 4/78, \ldots, p_{10} = 2/78$, and the number of advertisements is respectively 5, 33, 6, 3, 8, 9, 9, 21, 22 and 3 (totally $78*1.5=117$ advertisements) corresponding to the supply rate $q_1 = 5/117, \ldots, q_{10} = 3/117$.

In addition, we are interested whether or not filter distribution correlates to content popularity. The correlation means that a topic, if highly demanded by subscribers, simultaneously popularly appears in content items. Otherwise, both distributions are anti-correlated. By default, we set up the correlated topics to generate filters and content (with the Zipf distribution).

### Table 3 Parameters Used for Experiments

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Infocom06</th>
<th>MIT Reality</th>
<th>UCSD Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>num. of topics</td>
<td>10</td>
<td>10</td>
<td>30</td>
</tr>
<tr>
<td>num. of filters</td>
<td>78</td>
<td>95</td>
<td>274</td>
</tr>
<tr>
<td>num. of content items</td>
<td>$78 \times 1.5$</td>
<td>$95 \times 1.5$</td>
<td>$274 \times 1.5$</td>
</tr>
<tr>
<td>buf. size per node</td>
<td>[30: [5-100]]</td>
<td>[40: [5-100]]</td>
<td>[100: [20-150]]</td>
</tr>
<tr>
<td>running period $P$</td>
<td>[5: [1-840]] mins</td>
<td>[21: [1-90]] days</td>
<td>[1: [1-5]] days</td>
</tr>
<tr>
<td>Zipf parameter $\alpha$</td>
<td>0.95: [0.0-1.2]</td>
<td>0.95: [0.0-1.2]</td>
<td>0.95: [0.0-1.2]</td>
</tr>
</tbody>
</table>
During the experiment, we follow the above parameters to repeat the experiments by 10 times (with random sources and destinations) and measure the average of the following metrics.

- **Delivery ratio.** The average ratio of the number of successfully delivered destinations to the total number of destinations.
- **Average delay.** The average delay for all the delivered destinations to receive the data.
- **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 duplicately reach 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).

### 7.2 Effect of Time Period

First, in Figs. 6, 7 and 8, we study the effect of the allowable time period $P$ over the three traces. In Fig. 6a, among the four schemes, Ameba achieves comparable delivery ratio as Epidemic, Bubble Rap has a lower delivery ratio than Ameba, and ProPHET has the least delivery ratio. Next, a larger $P$ indicates that more nodes have a chance to relay content items towards the needed nodes, and thus the delivery ratios of all four schemes increase accordingly. Note that in Fig. 6a, the delivery ratio of Epidemic becomes saturated around 80 percent, due to the following reason: Given the fixed size of the buffer in each node, the node drops out those overflowed advertisements caused by the flooding messages in Epidemic. When we increase the default buffer size to 100, our experiment shows that Epidemic can achieve nearly 100 percent delivery ratio. This result will be consistently verified by Section 7.3.

Second, Fig. 6b plots the average delay, indicating that Ameba uses a low delay comparable to Epidemic to deliver content items towards needed nodes. It is because the developed optimization strategy can optimize the delivery of highly demanded content by using more nodes as carriers, and then content items are delivered in a timely manner.

Next, for the average cost in Fig. 6c, Ameba uses significantly low cost. It is caused by Ameba’s optimization policy (to assign more relay encounters for popular items) and forwarding algorithm (to adaptively create item copies). Instead, Bubble Rap forwards an item, if needed by more communities, to all such communities, leading to high cost close to Epidemic. Note that the average cost in this figure is even larger than the number of devices (98 imotes for the Infocom ’06 data set), because duplicate items reach the same nodes (caused by the many-to-many pub/sub model and LRU eviction policy for the limited buffer size). A similar situation occurs for other metrics, e.g., the average cost.

For the MIT reality and UCSD university traces given by Figs. 7 and 8, the nodes in the two traces use a longer time period than the nodes in the Infocom ’06 trace, before the same delivery ratio (for example, 80 percent) is achieved. It is because the encountering rates of the pairwise nodes in the two traces are much lower than those in the Infocom ’06 trace. Nevertheless, Ameba consistently uses the least average cost.

### 7.3 Effect of Buffer Size

In this experiment, we vary the buffer size $S$ per node, and study the performance of four schemes in Figs. 9, 10 and 11. In general, a larger $S$ helps achieve higher delivery ratios for all four schemes. Among the four schemes, Epidemic obviously benefits most from a larger $S$ by creating the largest number of copies. Similar to Epidemic and Bubble Rap, Ameba allows more copies of popular content items, and ProPHET benefits least from a larger $S$.

A larger $S$ does not necessarily lead to a smaller average delay and cost. In Fig. 9c, only when $S$ is sufficiently large, the average cost of Epidemic becomes smaller. This is because the average cost depends on the number of delivered advertisements and the number of relays used. Though
a larger $S$ does increase the number of delivered items, it also leads to a larger number of relays. Only when $S$ is sufficiently large, the number of delivered items becomes large enough to ensure smaller average cost. The trend consistently appears in all three traces. *Ameba* limits the copies of content and a larger $S$ consistently leads to smaller cost.

### 7.4 Effect of Zipf Parameter

Figs. 12, 13 and 14 study the effect of the Zipf parameter $\alpha$. A larger $\alpha$ achieves higher delivery ratio, lower average delay and cost for *Ameba*. The results in the figures are consistent with our analysis in Section 3, which favors the skewed distribution of $p_i$ and $q_i$. For the MIT reality and UCSD traces in Figs. 13 and 14, they exhibit a similar growth of delivery ratio, but with less average cost when $\alpha$ becomes larger.

In addition, the results of Epidemic in Figs. 12a, 13a and 14a show that the skewed distribution has little effect on the delivery ratio. It is because Epidemic blindly broadcasts content items and does not reduce the overall delivery ratio. However, due to the default correlated distribution between $p_i$ and $q_i$, Epidemic still ensures that a content item encounters the needed nodes earlier. Thus, the average delay and cost are reduced for a large $\alpha$. A similar situation occurs for ProPHET and Bubble Rap, though the reduction trend of average cost and delay is relatively smooth compared with Epidemic.

### 7.5 Comparison of Three Schemes

Figs. 15, 16 and 17 compare *Ameba* with the adapted broadcast and unicast schemes given in Section 5.1.

First in Fig. 15, by combining the benefits of both unicast and broadcast, *Ameba* outperforms the two adapted schemes in terms of all used metrics. Second, for all three schemes, the anti-correlated topics lead to relatively less delivery ratio and higher average delay, but with larger average cost compared with the results of correlated topics. For example the average cost of *Ameba* with anti-correlated topics is increased by 35.66 percent when compared with the one with correlated topics. Finally, we note that the unicast and broadcast in this figure use less delivery ratio than ProPHET and Epidemic, respectively. It is because the optimal encounters given by the developed optimization strategy can help the two adapted schemes achieve better delivery of the popular advertisements, which are highly demanded by the majority of mobile devices.

Similarly for MIT reality and UCSD traces, Figs. 16 and 17 show a similar trend as Fig. 15, and *Ameba* uses the least cost to achieve the comparable delivery ratio.

### 7.6 Study of Enhancement *Ameba*

In Fig. 18, we use the Infocom '06 trace to compare the delivery ratio, average delay and average cost of the original *Ameba* (Algorithm 1 in Section 5) and the enhanced *Ameba* (Algorithm 2 in Section 6). For convenience, we call Algorithms 1 and 2 “mobility-based” and “location-adaptivity-based” algorithms, respectively. In the figure, the location-adaptivity based enhancement outperforms the mobility-based scheme, with higher delivery ratios, similar delays and overheads. For example, in Figs. 18a, 19a and 20a, the location-adaptivity based enhancement achieves 27, 10 and 22 percent improvement in terms of delivery ratio. This is mainly because the enhancement leverages the spatial and temporal properties to capture the movement pattern of users more actually. Meanwhile, forwarding the content to those users who frequently visit hot areas during peak time improves the possibility to deliver the content to its destinations.
For the two university traces given by Figs. 19 and 20, the associated delivery ratio and average delay gradually grow when the allowable time period $P$ becomes larger. We can find that the growth trend of all three metrics in Fig. 18 differs from the ones in Figs. 19 and 20. The main reason is that the mobility pattern of those participants in the Infocom ‘06 conference is strongly decided by the conference schedule, and the university traces indicate the relatively long-term mobility pattern.

**8 RELATED WORK**

In the context of DTN, [29] studied Epidemic routing by selecting all nodes in the network as relays. ProPHET [21] leveraged encounter history to predict the delivery possibility. Spyropoulos et al. [28] sprayed data to relays waiting for contacts with destinations. Boldrini et al. [2] leveraged a Markovian model to develop an efficient utility function for content dissemination, and [17] investigated optimal rate allocation schemes to maximize the data dissemination speed. Burgess et al. [4] 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 [10], [12], [16], [24] 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 [4], [16], [24], and one-to-many routing [10], [12]. The journal version [13] of [12] further improves the model of [12] to analyze nodes’ capabilities for forwarding data to multiple destinations in case of the multiple-data multicast (MDM).

We mainly compared [12], [13] with our work as follows. (i) In terms of the scenario, [12], [13] consider that carrier nodes know the destination nodes of the data source, and therefore focus on maintaining a social forwarding path from the data source to destinations. Significantly differing from the scenario in [12], [13], the pub/sub model adopted in our work is widely used to decouple data sources and destinations, and it is impossible to maintain forwarding paths from data sources to destinations before content items have already reached the needed destinations. (ii) In terms of the studied problem, the MDM problem in [12], [13] assumes that the content items are all from one data source (i.e., the one-to-many manner) and instead we generally consider the many-to-many manner problem, where there exist an arbitrary number of data sources and the content needed by a subscriber could come from multiple data sources. Given the many-to-many manner, the maintenance of forwarding paths from a data source to destinations (i.e., a forwarding tree rooted at the source) leads to redundancy. For example [19] proposed a mesh overlay to avoid the inefficiency of maintaining an overlay tree for every source in Internet-scale data dissemination. (iii) In terms of the forwarding model, [12], [13] use the multicast manner, and we adaptively allow the unicast and (or) multicast, depending on the created content copies. Finally (iv) in terms of utilities for the selection of carriers, [12], [13] mainly use the utilities to predict the delivery possibility of mobile devices. The main purpose of our developed utilities is to capture the interests of mobile devices (together with mobility patterns). Given the above difference, it is infeasible for our work to maintain forwarding paths from data sources to destinations in our studied scenario and problem. Nevertheless, as shown in Section 4, the social properties in [12], [13] can be incorporated into the computation of forwarding utility in this paper.

In terms of the location-aware dissemination in DTNs, [5] analyzed the mobile users’ movement on the geographic axis and they observed that mobile users usually visit several regular locations rather than moving randomly. Based on users’ movement, they designed a mobility trajectory of the superuser to broadcast data to other normal users actively. Our paper instead analyzes the user movement between different locations on the time axis. We find that there are hot areas where the users frequently visit, and the user visit patterns change over time, exhibiting the spatial

![Fig. 16. Comparison: Reality (a) Delivery Ratio, (b) Avg. Latency, and (c) Avg. Cost.](image1)

![Fig. 17. Comparison: UCSD (a) Delivery Ratio (b) Avg. Latency (c) Avg. Cost.](image2)

![Fig. 18. Enhancement: Infocom '06 (a) Delivery Ratio, (b) Avg Latency, and (c) Avg Cost.](image3)

![Fig. 19. Enhancement: Reality (a) Delivery Ratio, (b) Avg. Latency, and (c) Avg. Cost.](image4)

![Fig. 20. Enhancement: UCSD (a) Delivery Ratio, (b) Avg. Latency, and (c) Avg. Cost.](image5)
and temporal properties. In addition, our algorithms avoid the single point failure problem caused by one single superuser [5], and instead adaptively select the relay nodes depending on the associated utilities.

Finally differing from our work to advertise content in DTNs, many previous works in the data mining area have widely studied the spread of influence maximization problem in online social networks. For example, [18] proposed the greedy algorithm on the problem of selecting nodes maximizing influence in a social network, and designed an approximation algorithm within a constant factor from the optimal. Leskovec et al. [20] generalized the work of [18] to determine those k blogs that a reader should read in order to detect the quick break of an important story.

9 Conclusion
To advertise content towards needed nodes over a DTN in a timely manner, Ameba carefully adjusts the number of encounters and the number of content copies for advertised content, develop forwarding utilities to capture interests, mobility patterns, capacity constraint and visit locations of mobile devices with low maintenance cost, and design distributed relay algorithms to select the best nodes as the carriers. Via extensive experiments, our evaluation demonstrates that the proposed Ameba scheme is able to achieve high delivery ratio and significantly low overhead.

Acknowledgments
This work was partially supported by Academy of Finland (Grant No. 139144). Part of the research was also conducted in the Internet of Things program of DIGILE (Finnish Strategic Centre for Science, Technology and Innovation in the field of ICT and digital business). Part of this work was done when Weixiong Rao was at the Department of Computer Science, University of Helsink, Finland. K. Zhao is the corresponding author.

References
Weixiong Rao received the BSc and MSc degrees from North (Beijing) Jiaotong University and Shanghai Jiaotong University respectively, and the PhD degree from the Chinese University of Hong Kong in 2009. After that, he was with the Hong Kong University of Science and Technology (2010), University of Helsinki (2011-2012), and the University of Cambridge Computer Laboratory Systems Research Group (2013). He is currently with the School of Software Engineering, Tongji University, China. His research interests include networked data system and energy-aware mobile computing.

Kai Zhao received the BS degree from Shandong University in 2009 and the MS degree from the University of Helsinki in 2011, and is currently working toward the PhD degree in the University of Helsinki. His research interests include human mobility, Internet of things, and mobile social networks.

Pan Hui received the BEng and MPhil degrees both from the Department of Electrical and Electronic Engineering, University of Hong Kong, and the PhD degree from Computer Laboratory, University of Cambridge. He is currently a faculty member of the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology, where he directs the HKUST-DT System and Media Lab. He is also a distinguished scientist of Telekom Innovation Laboratories (T-labs) Germany, and an adjunct professor of social computing and networking at Aalto University, Finland. Before returning to Hong Kong, he has spent several years in T-labs and Intel Research Cambridge. He has published more than 100 research papers and has some granted and pending European patents. He has founded and chaired several IEEE/ACM conferences/workshops, and was on the organizing and technical program committee of numerous international conferences and workshops including the IEEE Infocom, ICNP, SECON, MASS, Globecom, WCNC, ITC, ICWSM, and WWW. He is an associate editor for both the IEEE Transactions on Mobile Computing and the IEEE Transactions on Cloud Computing.

Sasu Tarkoma received the MSc and PhD degrees in computer science from the Department of Computer Science, University of Helsinki. He is a full professor in the Department of Computer Science, University of Helsinki, and the head of the networking and services specialization line. He has managed and participated in national and international research projects at the University of Helsinki, Aalto University, and the Helsinki Institute for Information Technology (HIIT). He was in the IT industry as a consultant and chief system architect as well as the principal researcher and laboratory expert at Nokia Research Center. His interests include mobile computing, Internet technologies, and middleware. He is a senior member of the IEEE.

Yan Zhang received the PhD degree from Nanyang Technological University, Singapore. He is with Simula Research Laboratory, Norway, and is an adjunct associate professor at the University of Oslo, Norway. He is an associate editor or guest editor of a number of international journals. He is an organizing committee chair for many international conferences. His research interests include resource, mobility, spectrum, energy, and data management in communication networks.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.