Preference-Aware Content Dissemination in Opportunistic Mobile Social Networks

Kate Ching-Ju Lin, Chun-Wei Chen
Research Center for IT Innovation
Academia Sinica, Taiwan

Cheng-Fu Chou
Dept. of Computer Science and Information Engineering
National Taiwan University, Taiwan

Abstract—As mobile devices have become more ubiquitous, mobile users increasingly expect to utilize proximity-based connectivity, e.g., WiFi and Bluetooth, to opportunistically share multimedia content based on their personal preferences. However, many previous studies investigate content dissemination protocols that distribute a single object to as many users in an opportunistic mobile social network as possible without considering user preference. In this paper, we propose PrefCast, a preference-aware content dissemination protocol that targets on maximally satisfying user preference for content objects. Due to non-persistent connectivity between users in a mobile social network, when a user meets neighboring users for a limited contact duration, it needs to efficiently disseminate a suitable set of objects that can bring possible future contacts a high utility (the quantitative metric of preference satisfaction). We formulate such a problem as a maximum-utility forwarding model, and propose an algorithm that enables each user to predict how much utility it can contribute to future contacts and solve its optimal forwarding schedule in a distributed manner. Our trace-based evaluation shows that PrefCast can produce a 18.5% and 25.2% higher average utility than the protocols that only consider contact frequency or preference of local contacts, respectively.

I. INTRODUCTION

Mobile Social Networks (MSNs) [1] [2] have become one of the emerging wireless communication techniques due to ubiquitous mobile devices, such as smartphones and pads, and their inherent proximity-based sharing capability, such as WiFi and Bluetooth. MSNs are originally developed to enable mobile users to opportunistically exchange information via social contacts even without permanent network connectivity between users [3]. Recently, MSNs are further integrated with 3G cellular networks to offload the traffic of multimedia content dissemination or advertisement broadcasting [4]. Instead of delivering content objects to each mobile user individually, a service provider can forward objects to part of users using 3G connection, and allow users to form an MSN and exploit opportunistic communications to carry-and-disseminate the objects, which consequently offloads cellular traffic. Therefore, there is an increasing demand for efficient content dissemination in a mobile social network.

With the growing popularity of personalization applications, such as Youtube or Pandora, clients prefer to access multimedia content objects based on their personal interests. For instance, some users might be interested more in sport clips, while some would like news clips more. Several unicast or multicast protocols designed for MSNs [5]–[13] however only target on speeding up content dissemination, without considering heterogeneous user preferences for various content objects. Even though those dissemination schemes can distribute objects quickly to as many users as possible, they might not be able to maximally satisfy user preferences. Therefore, the goal of this work is to design a preference-aware content dissemination system for MSNs.

Since connection between users in an MSN exists only during a very short contact period, an efficient content forwarding strategy that schedules when to forward which objects significantly determines the total utility (i.e., the quantitative metric of user satisfaction) that can be brought to all mobile users. Intuitively, a mobile user can forward objects merely according to the preference of his current contacts. It is however a sub-optimal strategy because, in an MSN, a user who receives the objects from the forwarder can further distribute those objects to his future contacts until the objects expire. Therefore, to maximize the utility contributed by each object forwarding, the forwarder should also consider the utility that could be gained by future contacts.

Consider the example shown in Fig. 1, where forwarder F owns objects $m_1$ and $m_2$ and meets neighboring users A and B. Each user has preferences (i.e., utility) $u_1$ and $u_2$ for objects $m_1$ and $m_2$, respectively, as shown in the figure. Assume that the contact duration is only long enough for forwarder F to disseminate one object. If the forwarder chooses to broadcast object $m_1$, it can provide its neighbors A and B the maximal utility, i.e., $u_1^A + u_1^B = 10 + 5 = 15$. However, such a forwarding strategy cannot achieve the maximal utility of the entire network because object $m_1$ is less popular than $m_2$. In particular, even though the current contacts A and B can gain a larger utility (called local utility for short) by retrieving $m_1$ from forwarder F, they cannot contribute a high global utility by sharing the object to their future contacts in groups $G_A$ and $G_B$ because object $m_1$ is less popular in

This work is partially supported by the National Science Council of R.O.C. under the contract No. NSC 100-2221-E-001-005-MY2.
$G_A$ and $G_B$. The objective of this paper is thus to develop a preference-aware content dissemination protocol that produces the maximal global utility to all mobile users in an MSN.

We propose PrefCast, a Preference-aware broadcast scheme that enables each forwarder to determine its forwarding strategy such that all users can cooperatively distribute objects and achieve the maximal global utility. To achieve the above goal, PrefCast has to address two main challenges:

(a) How does a node predict the utility it can contribute to future contacts (i.e., future utility contribution)? A forwarder can easily query the preference of its current contacts. However, to maximize the global utility, the forwarder also needs to know how much future utility can be generated if it broadcasts an object to its current contacts. Specifically, it needs to estimate how many future contacts can be encountered by its current contacts and what are the preferences of those future contacts. In addition, each user usually only has a limited buffer space, and therefore cannot cache all the objects overheard from other forwarders. To improve prediction accuracy of future utility contribution that can be brought by a user, we should also consider the probability that an object will be dropped by that user due to buffer overflow.

(b) Given the estimate of the future utility contribution of each contact, how does a forwarder determine its optimal forwarding schedule? A forwarder might simultaneously meet multiple neighboring users, each of which has a different contact duration and can contribute a heterogeneous future utility. An object broadcasted in different time-slots might be overheard by different sets of neighboring users, and, thus, results in different total global utility contributions. Because the utility contribution of broadcasting the same object varies with time, a greedy scheme that broadcasts the object with the highest utility contribution might not guarantee the maximal global utility contribution for the entire network. Therefore, when a forwarder has multiple objects to broadcast, it needs to schedule the optimal forwarding sequence of its multiple objects with consideration of the contact durations and future utility contribution of all the current contacts.

The contribution of PrefCast is twofold. First, it transforms the forwarding scheduling problem to the maximum weight bipartite matching problem [14] for finding the maximum-utility forwarding sequence for each forwarder. Next, since the effectiveness of PrefCast depends to a large extent on accurate estimation of future utility contribution, we then propose a metric to predict the future utility contribution of each user. Our trace-based evaluation demonstrates that our preference-aware dissemination scheme, PrefCast, can produce a 18.5% and 25.2% higher total utility than the preference-oblivious and local-preference-based dissemination schemes, respectively, for all users in a mobile social network.

The remainder of this paper is organized as follows. Section II provides a review of related works on MSNs. Sections III and IV describe the PrefCast framework and the prediction metric, respectively. In Section V, we evaluate the performance of PrefCast using real trace data. Finally, Section VI contains some concluding remarks.

II. RELATED WORK

An MSN is a type of Delay Tolerant Networks (DTNs) [15], but considers an environment where users contact each other in their daily activity. Prior works on MSNs or DTNs can be classified into three categories: unicast, multicast, and content dissemination.

Several unicast routing schemes have been proposed to improve the end-to-end delivery ratio or transmission delay in a DTN. For example, Epidemic routing [16] floods messages to every contact until the messages reach the destination. To reduce the overhead of epidemic routing, some later works investigate the trade-off between the number of relays and the delivery probability. In PROPHET [5], each user predicts the contact probability of each node pair, and forwards messages to the node that has the largest contact probability with the destination. Spray and wait routing [17] operates similar to Epidemic routing, but restricts the number of forwardings for each object in order to reduce the forwarding overhead. Recently, social-based forwarding schemes [7] [8] consider various social network properties including centrality and communities, and forward data to the nodes playing the vital roles in a social network.

Unicast routing protocols have been extended to support multicast in [10]–[12]. The approach proposed in [10] attempts to maximize the message delivery ratio for multicast members that join or leave the network dynamically. In [11] [12], given a set of destinations, the authors extend the two-hop relay algorithms used in unicast to support multicast destinations.

Different from the above unicast and multicast protocols that have a set of specific destinations, our work targets on distributing messages to as many users as possible for content dissemination applications. The dissemination protocol proposed in [13] ensures that content delivered to the users is as “fresh” as possible subject to the limited downlink capacity of the service provider. Bo et al. [4] propose a framework to offload cellular traffic through opportunistic communications; however, this work focuses on selecting a suitable set of initial data sources, but does not discuss how to distribute messages in an opportunistic network. The user-centric scheme [18] attempts to select the minimum number of relays to forward an object to those users who might be interested in that objects. PrefCast differs from the above dissemination protocols in that it takes heterogeneous user preference into account, and enables each forwarder to determine its optimal object forwarding schedule that can produce the maximal total utility for all users in an MSN.

III. PrefCast FRAMEWORK

In this section, we first formally define the preference-aware content dissemination problem, and derive the maximum-utility forwarding model to optimally satisfy user preference. An optimal forwarding scheduling algorithm is then proposed to solve the above model. Finally, we analyze the performance
A. Problem Definition and Assumptions

We consider a mobile social network with a set of users \( V \) sharing a set of multimedia content objects \( M \). Suppose each user \( i \in V \) has a different preference \( u_{i,m} \) for each object \( m \in M \). User \( i \) can gain the utility \( u_{i,m} \) if it retrieves object \( m \) before the object \( m \) expires. Each user uses proximity-based communication techniques, such as WiFi or Bluetooth, to broadcast its objects to neighboring users within its transmission range. We define that a contact occurs when any two mobile users locate within the transmission range of each other. A user who receives any object becomes a forwarder, and is then able to help distribute that object to its future contacts. Each forwarder might have multiple objects to share with neighboring users, but can only broadcast a single object at a time. It therefore needs to determine the forwarding sequence of its objects. We refer to this problem as the forwarding scheduling problem, and derive it as a discrete-time model, where the length of each time-slot is set to the time required to broadcast an object. Without loss of generality, we assume that each transmission starts at the beginning of a time-slot and occupies the whole slot.

Specifically, we let \( M_f^\ast \) denote the set of objects stored by forwarder \( f \in V \) in a given time-slot \( \tau \), which is a subset of all objects, i.e., \( M_f^\ast \subseteq M \). Each object \( m \in M_f^\ast \) might be an original object owned by user \( f \) or a copy retrieved from other users, making \( f \) become a forwarder. We then collect the users that locate within the transmission range of forwarder \( f \) in time-slot \( \tau \) as a group \( V_f^\ast \). Since mobile users might join or leave a group arbitrarily, the members in forwarder \( f \)’s group \( V_f^\ast \) could change with time. This implies that broadcasting the same object \( m \in M_f^\ast \) in different time-slots, e.g., the current time-slot \( \tau \) or any future time-slot \( t > \tau \), could produce different utilities. Therefore, each forwarder must schedule the transmission sequence of its objects such that the total utility gained by all users can be maximized. In particular, for any forwarder \( f \), we define a binary variable \( x_{m,t} \) to indicate whether \( f \) decides to broadcast object \( m \) in time-slot \( t \). Namely, \( x_{m,t} = 1 \) if object \( m \) will be broadcast in time-slot \( t \), and \( x_{m,t} = 0 \) otherwise. Our goal is to enable each forwarder to find its optimal forwarding schedule \( x^\ast_{m,t} \), \( \forall m \in M_f^\ast, t \geq \tau \), denoted by the vector form \( x^\ast \) for short, to produce the maximal total utility for all users in an MSN. We formulate the above problem as the maximum-utility forwarding model in the next section, followed by the proposed solution to the model. Table I summarizes our definitions.

B. Maximum-Utility Forwarding Model

When forwarder \( f \) meets a group of neighboring users \( V_f^\ast \) in time-slot \( \tau \), it can broadcast any object \( m \in M_f^\ast \) to all its neighbors in \( V_f^\ast \). Each user \( i \in V_f^\ast \) who does not own object \( m \) can then obtain the utility \( u_{i,m} \). The forwarder \( f \) can therefore produce the local utility contribution \( \omega_{m,\tau} = \sum_{i \in V_f^\ast} u_{i,m} \).

Table I Notations used in the maximum-utility forwarding model

<table>
<thead>
<tr>
<th>Notations</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V )</td>
<td>the set of users in an MSN</td>
</tr>
<tr>
<td>( M )</td>
<td>the set of content objects</td>
</tr>
<tr>
<td>( V_f^\ast )</td>
<td>the set of users that have contact with forwarder ( f ) in time-slot ( \tau )</td>
</tr>
<tr>
<td>( M_f^\ast )</td>
<td>the set of objects owned by forwarder ( f ) in time-slot ( \tau )</td>
</tr>
<tr>
<td>( u_{i,m} )</td>
<td>preference (utility) of user ( i ) for object ( m )</td>
</tr>
<tr>
<td>( d_{i,m} )</td>
<td>the duration of contact between user ( i ) and ( m )</td>
</tr>
<tr>
<td>( T )</td>
<td>the set of feasible forwarding time-slots for a forwarder</td>
</tr>
<tr>
<td>( \omega_{m,\tau} )</td>
<td>total local utility contribution of broadcasting object ( m ) by forwarder ( f ) in time-slot ( \tau )</td>
</tr>
<tr>
<td>( U(i,m,\tau) )</td>
<td>the future utility contributed by ( i ) if it gets object ( m ) in time-slot ( \tau )</td>
</tr>
<tr>
<td>( \Omega(x), \Omega^* )</td>
<td>the total utility contributed by a forwarder schedule ( x ), and the maximal total utility ( \Omega^* = \max_x \Omega(x) )</td>
</tr>
</tbody>
</table>

Given a forwarding schedule \( x \), a forwarder can generate the following total global utility contribution for all time-slots \( t \geq \tau \): \( \Omega(x) = \sum_{m \in M_f^\ast, t \geq \tau} U(i,m,\tau) \).

We leave presenting how each user predicts its future utility contribution \( U(i,m,\tau) \) in Section IV, and first address how a forwarder \( f \) schedules the forwarding sequence of all its objects \( m \in M_f^\ast \) according to the global utility contribution \( \omega_{m,\tau} \) for all \( t \geq \tau \). Intuitively, for each time-slot \( t \), forwarder \( f \) can greedily maximize the global utility by broadcasting the object \( m_g \) with the maximal global utility contribution, i.e., \( m_g = \arg\max_{m \in M_f^\ast} \omega_{m,\tau} \). We let \( \hat{x} \) denote the forwarding schedule selected by such a greedy solution, and hence \( \hat{x}_{m_g,t} = 1 \). The forwarder \( f \) can then remove object \( m_g \) from
achieve the maximal total contribution, predict the contact durations, multiple successive time-slots after time-slot \(\tau\). Such a maximum-utility forwarding problem for forwarder \(f\) in time-slot \(\tau\) can be transformed to the maximum weight bipartite matching (MWBM) problem, and the system can only gain the future utility contribution \(U(i, m, t)\) if user \(i\) receives the object \(m\) before it leaves the group, i.e., \(t < \tau + d_{fji}\). Therefore, the global utility of broadcasting object \(m\), i.e., \(\omega^\theta_{m,t}\), varies with time-slot \(t\), as shown in Eq. (3a). The objective is hence to find the optimal forwarding schedule \(x^*_{m,t}\) for all \(m \in M_f^\tau\) and \(t \in \mathcal{T}\), such that forwarder \(f\) can contribute the maximal total global utility \(\Omega^* = \Omega(x^*)\).

C. Optimal Forwarding Scheduling Algorithm

The above maximum-utility forwarding model can actually be transformed to the maximum weight bipartite matching (MWBM) problem [19]. To represent our maximum-utility forwarding problem as the MWBM problem, we create a bipartite graph \(G = (M_f^\tau \cup \mathcal{T}, E)\), where the vertex set is the union of the sets of objects and available time-slots and the edge set \(E = \{(m, \tau) : m \in M_f^\tau, \tau \in \mathcal{T}\}\), as shown in Fig. 2. Each edge \(e = (m, \tau) \in E\) is associated with a weight, which is set to the global utility \(\omega^\theta_{m,t}\) that \(f\) can contribute by broadcasting object \(m\) in time-slot \(\tau\), i.e., setting \(x_{m,t} = 1\).

Note that our model restricts that each object \(m \in M_f^\tau\) can be assigned at most a single time-slot and, in addition, each time-slot \(\tau \in \mathcal{T}\) can be allocated to at most a single object. Therefore, any feasible solution of the forwarding schedule \(x\) is a matching \(\bar{M}\) in the bipartite graph \(G\), where \(\bar{M}\) is a subset of \(E\) such that no two edges in \(\bar{M}\) share an endpoint, as the red lines shown in Fig. 2. In particular, for any \(m \in M_f^\tau\) and \(\tau \in \mathcal{T}\), the forwarding schedule \(x_{m,t} = 1\) if edge \((m, \tau)\) is included in the matching \(\bar{M}\), and \(x_{m,t} = 0\), otherwise. Then, we get that the total weight of edges in \(\bar{M}\) exactly equals the total global utility contribution \(\Omega(x) = \sum_{m \in M_f^\tau, \tau \in \mathcal{T}} x_{m,t} \omega^\theta_{m,t}\). As a result, finding the maximal total global utility \(\Omega^*\) is

![](image.png)
equivalent to solving the maximum weight matching in the bipartite graph \( G \). Therefore, we can apply the well-known polynomial-time algorithm, called the Hungarian algorithm [14], to solve the maximum weight bipartite matching, i.e., the optimal forwarding schedule \( x^* \).

D. How Well does Greedy Approximate the Optimal Solution?

The last section shows that the forwarder can apply the Hungarian algorithm to find the optimal forwarding schedule \( x^* \) that produces the maximal utility \( \Omega^* \). However, since members in group \( V_f^* \) could gradually leave the group after the current time-slot \( t \) and, also, the forwarder cannot predict who will join the group after \( t \), the global utility function \( \omega_{g,m,t} \) for any object \( m \) is hence a monotonic decreasing function over \( t \), i.e., \( \omega_{g,m,t} \leq \omega_{g,m,t}' \) if \( t' > t \). With the monotonic decreasing property of \( \omega_{g,m,t} \), the greedy forwarding schedule \( \tilde{x} \) can actually produce a total utility \( \tilde{\Omega} = \Omega(\tilde{x}) \) that approximates the maximal utility \( \Omega^* \). Before computing the performance gap between \( \Omega^* \) and \( \tilde{\Omega} \), we first derive the following Lemma.

**Lemma 1**: The transmission time-slots of the objects assigned by the greedy forwarding schedule \( \tilde{x} \) could be reordered to achieve a higher utility.

**Proof**: Suppose the greedy algorithm decides to broadcast two objects \( m \) and \( m' \) in time-slots \( t \) and \( t' \), i.e., \( \tilde{x}_{m,t} = \tilde{x}_{m',t'} = 1 \), where \( t < t' \). Then, broadcasting two objects \( m \) and \( m' \) can contribute the utility \( \tilde{\Omega} = \omega_{m,t} + \omega_{m',t} \). Say if we swap the order of these two transmissions, i.e., sending object \( m' \) in \( t' \) and sending object \( m \) in \( t \), the utility then becomes \( \Omega' = \omega_{m',t'} + \omega_{m,t} \). Because it is possible that utility degradation of object \( m' \) between time-slots \( t' \) and \( t \) is larger than that of object \( m \), i.e., \( \omega_{g,m,t'} - \omega_{g,m,t'}' > \omega_{g,m,t} - \omega_{g,m,t} \), in this case, the swapped order can produce a higher utility for two objects than the greedy solution, i.e., \( \Omega' = \omega_{m',t'} + \omega_{m,t} > \omega_{m,t} + \omega_{m',t'} \). Hence, we get that it is possible to reorder the sequence of object transmissions in the greedy schedule \( \tilde{x} \) to achieve a higher utility.

Based on the above Lemma, if the number of available objects is no larger than the number of all available time-slots, i.e., \( |M| \leq |\mathcal{T}| \), both the greedy and optimal solutions must broadcast all the objects, but just arrange them in different transmission orders. Let \( t_m^* \) and \( t_m \) denote the time-slots assigned by the optimal and greedy solutions, respectively, to object \( m \), i.e., \( x_{m,t_m^*} = 1 \) and \( x_{m,t_m} = 1 \). The gap between the total utility of the optimal solution, \( \Omega^* \), and the utility of the greedy solution, \( \tilde{\Omega} \), then equals \( \Omega(x^*) - \Omega(\tilde{x}) = \sum_{m \in M} (\omega_{m,t_m^*} - \omega_{g,m,t_m}) \). On the contrary, when \( |M| > |\mathcal{T}| \), the performance gap between \( \Omega^* \) and \( \tilde{\Omega} \) could become larger because some objects selected by the optimal schedule might not be included in the greedy schedule. However, we observe that this situation rarely happens.

That is, the gap between \( \Omega^* \) and \( \tilde{\Omega} \) mainly comes from utility degradation of any object \( m \) between the assigned time-slots \( t_m^* \) and \( t_m \) if \( m \) is selected by both the greedy and optimal schedules. Recall that such utility degradation over time is because some users might leave the group earlier and can only help contribute the utility if they get the object before leaving the group. Thus, if all users join the group \( V_f^* \) with a similar duration, based on Eq. (3a), \( \omega^*_{m,t} \) then does not change much over time. In this case, the greedy solution can approximate the optimal solution well. By contrast, the performance gap increases if more users leave in different time-slots.

IV. METRIC ESTIMATION

Since there are no specific destinations in content dissemination applications, our goal is to propagate objects to as many users who are interested in them as possible. In contrast to existing MSN unicast routing protocols that predict the probability of delivering a message to the destinations, our model needs each user \( i \) to predict the future utility contribution \( U(i,m,t) \) that it can contribute by forwarding object \( m \) to all its contacts during \( [t,T^\text{max}_m] \), where \( T^\text{max}_m \) is the expiration time of object \( m \). Note that user \( i \) can only contribute object \( m \) to any contact \( j \) if \( j \) does not have object \( m \) when they meet after time-slot \( t \). Intuitively, we have \( U(i,m,t) \geq U(i,m,t') \) for any time-slots \( t < t' \) if \( i \) does not drop \( m \) after \( t \). This is because the number of users that have not downloaded object \( m \) decreases with time. Therefore, the value of \( U(i,m,t) \) can be determined by three factors: 1) the probability that forwarder \( i \) can meet other contacts who have not owned object \( m \) after time-slot \( t \); 2) the utility of \( i \)'s contacts on object \( m \); and 3) the probability that user \( i \) drops object \( m \) due to its limited buffer space.

To compute \( U(i,m,t) \), we start by deriving the probability that a contact \( j \in \mathcal{V} \) has not received object \( m \) in time-slot \( t' \) from any contacts \( k \in \mathcal{V} \), denoted by \( \Phi_{j,m}(t') \). Based on the observation in [12] [13], we can model the contact process of each node pair as a homogeneous Poisson process. Let the random variable \( N_{jk}(y) \) denote the cumulative number of contacts between \( j \) and \( k \) in continuous-time \( y \) with the mean contact frequency \( \lambda_{jk} \). However, since we formulate the problem as a discrete-time model, we redefine the random variable \( N_{jk,y}(y) \) as \( N_{jk}^d(t) \), which equals the cumulative number of contacts between \( j \) and \( k \) until the \( t^{th} \) time-slot, and set \( N_{jk}^d(t)=N_{jk}(T_{tx}(t)), \) where \( T_{tx} \) is the length of a time-slot. Without loss of generality, we let any contact in time interval \( [T_{tx}(t-1), T_{tx}t] \) start transmitting in time \( T_{tx}t \), i.e., the beginning of time-slot \( t \). Then, we can compute the probability \( q_{jk} \) that \( j \) and \( k \) will not meet in a time-slot \( t \) by

\[
q_{jk} = P(N_{jk}(T_{tx}t) - N_{jk}(T_{tx}(t-1)) = 0) = e^{-\lambda_{jk}T_{tx}}. \tag{4}
\]

Thus, we can set the contact probability between \( j \) and \( k \) in each time-slot at \( (1-q_{jk}) \). We observe that user \( j \) will not be able to download object \( m \) from user \( k \) before time-slot \( t \) if and only if one of two following events occurs: 1) \( j \) and \( k \) have never met before time-slot \( t \), i.e., \( N_{jk}^d(t-1)=0 \); or 2) the last contact between \( j \) and \( k \) was in time-slot \( z < t \), but \( k \) did not have object \( m \) in time-slot \( z \). Therefore, the probability \( \Phi_{j,k,m}(t) \) that \( j \) cannot download object \( m \) from \( k \) before time-slot \( t \) can be computed by

\[
\Phi_{j,k,m}(t) = q_{jk}^{t-1} + \sum_{z=1}^{t-1} (1-q_{jk}) q_{jk}^{t-1-z} \Phi_{k,m}(z). \tag{5}
\]
Then, the probability $P_{j,m}(t)$ that user $j$ has not downloaded object $m$ until time-slot $t$ equals the probability that $j$ cannot download object $m$ from any user $k \in \mathcal{V}$, $j \neq k$, before $t$.

$$P_{j,m}(t) = \prod_{j \neq k} \Phi_{j,k,m}(t)$$

(6)

Because Eq. (6) includes the probability $q_{jk}$ of no contact between $j$ and $k$, user $j$ has a lower probability $P_{j,m}(t)$ if it can meet other users $k \in \mathcal{V}$ more frequently, i.e., a lower $q_{jk}$. Moreover, the contact $k$ will only be helpful if it carries object $m$ when it meets $j$. Consequently, $j$ is more likely to have a lower probability $P_{j,m}(t)$ and cache object $m$ in time-slot $t$ if its contact $k$ has a higher probability of receiving object $m$ before they meet, i.e., with a lower $P_{k,m}(z)$ for all $z < t$.

Since only the data sources have object $m$ in the initial time-slot 1, we have $P_{k,m}(1) = 0$ if $k$ is a data source of object $m$ and $P_{k,m}(1) = 1$ otherwise. Given the initial value of $P_{k,m}(1)$, we can compute the value of $P_{k,m}(z)$ for each time-slot $z = 2, 3, \ldots, t-1$ iteratively for all users $k \in \mathcal{V}$. Note that, even though the computation of $P_{k,m}(t)$ in Eq. (6) involves the value of $P_{k,m}(z)$ for all users $k \in \mathcal{V} \setminus \{j\}$, $j$ does not really need to solve $P_{k,m}(z)$ for all users. Instead, $j$ only needs to compute $P_{k,m}(z)$ for its historical contacts because, if $j$ and $k$ have never met, the contact probability $(1 - q_{jk})$ must be 0. To reduce the complexity, we let each user $j \in \mathcal{V}$ compute its $P_{j,m}(t)$ and exchange the information about $P_{j,m}(t)$ with other users $k \in \mathcal{V}$ when they meet. Hence, the computational complexity can be shared by all users.

We can compute the cumulative probability that $i$ can forward object $m$ to contact $j$ between $t$ and $T_{m}^{\text{max}}$ as follows.

$$U(i, m, t) = \sum_{z=t}^{T_{m}^{\text{max}}-1} (1-q_{ij})^{T_{m}^{\text{max}}-1-z} \Phi_{j,m}(z)$$

(7)

However, the object $m$ might be dropped by user $i$ in any time-slot $z$ due to buffer overflow with a probability $d_{i,m}(z)$. The cumulative probability needs to be rewritten as

$$U(i, m, t) = \sum_{z=t}^{T_{m}^{\text{max}}-1} (1-q_{ij})^{T_{m}^{\text{max}}-1-z} \Phi_{j,m}(z)(1 - d_{i,m}(z))$$

(8)

The expected utility that $i$ can contribute to contact $j$ between $t$ and $T_{m}^{\text{max}}$ for object $m$ then equals

$$\kappa_{i,j,m}(t) = U_{i,j,m} \sum_{z=t}^{T_{m}^{\text{max}}-1} (1-q_{ij})^{T_{m}^{\text{max}}-1-z} \Phi_{j,m}(z)(1 - d_{i,m}(z)).$$

Therefore, the expected total utility that all future contacts can gain by downloading object $m$ from user $i$ during $[t, T_{m}^{\text{max}}]$ can be predicted by

$$U(i, m, t) = \sum_{n=0}^{\infty} (\sum_{F \cap F_{n} \neq \emptyset} \prod_{j \in F} \kappa_{i,j,m}(t) \prod_{j \in F} (1-\kappa_{i,j,m}(t)))$$

(9)

In Eq. (8), the dropping probability $d_{i,m}(z)$ depends on the dropping policy of each user. We consider the policy that each user $i$ drops the object $m$ if it is least interested in, i.e., with the lowest utility $u_{i,m}$, when its buffer is overflowed. Specifically, each user sorts its objects by utility in ascending order, and drop the first object when it does not have enough buffer space to cache a new object. Assume that user $i$ ranks object $m$ as the $k^{\text{th}}$ object of interest among all its objects. The user will drop object $m$ if it receives $k$ objects with a utility higher than object $m$ after its buffer becomes full. Say that, in the current time-slot $\tau$, user $i$ has a residual buffer space that can cache additional $\beta$ objects. The dropping probability $d_{i,m}(z)$ can then be computed by the following equation.

$$d_{i,m}(z) = \sum_{t=\tau}^{z} \frac{P(R_{i}(t-\tau, 0) = \beta)P(R_{i}(z-t, u_{i,m}) \geq k)}{\beta!}$$

(11)

where $R_{i}(t, u)$ denotes the number of objects with a utility higher than $u$ received by user $i$ during a period of $t$ time-slots. Thus, $P(R_{i}(t-\tau, 0) = \beta)$ indicates the probability that user $i$ receives any $\beta$ objects from $\tau$ to $t$ and leads to a full buffer, while $P(R_{i}(z-t, u_{i,m}) \geq k)$ is the probability that user $i$ receives more than $k$ objects with a utility higher than $u_{i,m}$ during time-slots $[t, z]$, as a result dropping object $m$. Again, based on [12] [13], we model the contact process of each node pair as a homogeneous Poisson process, and estimate the above probability as follows.

$$P(R_{i}(t-\tau, 0) = \beta) = \frac{(\lambda_{u}(t-\tau))^{\beta} e^{-\lambda_{u}(t-\tau)}}{\beta!}$$

(12)

$$P(R_{i}(z-t, u) \geq k) = 1 - \sum_{b=0}^{k-1} \frac{(\lambda_{u}(z-t))^{b} e^{-\lambda_{u}(z-t)}}{b!}$$

(13)

where $\lambda_{u}$ is the frequency of meeting a user who can contribute an object with the utility higher than $u$. In other words, $\lambda_{u}$ denotes the frequency of meeting a user who can contribute any object. However, since it is difficult to predict whether a contact can share an object with the utility higher than $u$, we approximate the value of $\lambda_{u}$ by the contact frequency multiplied the normalized utility, i.e., $\lambda_{u} = \lambda_{0} * u / \sum_{m \in M_{\tau}^{+}} u_{i,m}$, where $M_{\tau}^{+}$ is the set of objects owned by user $i$ in $\tau$.

V. PERFORMANCE EVALUATION

We then evaluate the performance of PrefCast using three real traces: NUS [20], INFOCOM06 [21], and MIT Reality [22]. The NUS trace contains the schedules of the 4,885 classes and 22,341 students for 77 class hours. We randomly select 500 users from the trace in our trace-based simulations. We assume that two students can contact each other if and only if they are in the same classroom, which follows the same method in [20]. We randomly select some students in each class to be absent or leave early, and generate contact patterns outside the classrooms based on the survey and observations made in [20]. The INFOCOM06 trace includes 78 users who attend the same conference for a few days, while the MIT Reality trace includes 97 users who work in the same building.
Except the above large-scale and small-scale traces, we also evaluate the performance through the synthetic trace generated based on SLAW, the state-of-the-art human mobility model proposed in [23]. Each synthetic trace includes 150 users.

We use log-based user profiles collected from Last.fm [24], a database that tracks listening habits of music. We collect profiles for 8,000 users. For each user, the data set records the 100 songs it had listened to the most. We categorize the songs of each user based on their artists, and let the number of songs owned by a user with the same artist represent its preference (i.e., utility) for any songs of that artist. For example, if a user has 5 songs of Justin Bieber, that user has a preference 5 for Justin Bieber. That is, it can gain the utility 5 if it downloads any song of Justin Bieber that is not cached in its buffer. We let each user associates with a randomly-selected listening profile, and compute its utility for each object based on the artist of that object. We consider a scenario where each user can use the broadcast technique, such as WiFi multicast, to distribute its objects at the transmission rate 1 Mb/s. For each trace-based simulation, we randomly select 10 users as the initial data sources, each of which helps disseminate a distinct 30MB object. Each user has a buffer that can store 100 objects.

A. Performance of PrefCast

Our trace-based simulations compare the following schemes: 1) Epidemic routing [16], 2) PROPHET [5], 3) Local Utility, 4) PrefCast (Greedy), and 5) PrefCast (Optimal). In Epidemic routing, each forwarder randomly selects an object to broadcast in each time-slot. PROPHET focuses on finding the forwarder that has the highest probability of delivering a message to the destination using unicast. However, since we consider a content dissemination environment, we modify PROPHET to Interest-based PROPHOE (or I-PROPHET), which allows a forwarder to find an object that can be distributed to the most number of users. Specifically, we let each user only have a binary utility for each object, i.e., an interested or non-interested object, and compute its probability of having a contact with other users who have not owned that object. The forwarder then computes the summation of such contact probability for all group members who are interested in an object as the weight of that object, and broadcasts the object with the highest weight. In Local Utility, each forwarder broadcasts the object with the highest local utility \( \omega^l \) in each time-slot. Finally, we evaluate our proposed PrefCast that considers the global utility \( \omega^g \) with respect to optimal forwarding scheduling and greedy forwarding scheduling.

Fig. 3 plots the average utility obtained by each user over time. The figures show that PrefCast can produce a 18.5% and 25.2% higher average utility than I-PROPHET and Local Utility, respectively, until the deadline. The improvement is mainly because that, in PrefCast, each forwarder considers the future utility contribution of its members. Specifically, the utility metric \( U(i, m, t) \) of a group member \( i \) used in PrefCast not only considers heterogeneous user preference, but, more importantly, predicts how many users can gain utility by obtaining object \( m \) from member \( i \). Therefore, a forwarder can select the object that can benefit all users in the system, instead of an object that only interests local neighboring users. The figures also show that, in most of the traces, I-PROPHET performs better than Local Utility. This also explains that the number of future feasible contacts, i.e., the estimated number of contacts who have not owned the considering object, has a greater impact on utility contribution than local preference.

In most of traces, the greedy forwarding schedule can perform close to the optimal forwarding schedule because of the monotonic decreasing utility function, as discussed...
in Section III. The gap between the greedy and optimal schedules is highly related to the number of objects owned by a forwarder and the duration of a group, i.e., the number of decision variables $x_{m,t}$ in the bipartite matching problem. Then, because users in the NUS and SLAW traces have a longer contact duration, the performance gap in NUS and SLAW therefore increases slightly.

We then vary the number of distinct objects from 10 to 40. Fig. 4 plots the average utility of all users until the end of the simulations. PrefCast can always produce a higher utility than the other schemes. The improvement increases with the number of distinct objects because heterogeneity in object utility becomes more obvious when there are more objects in the system. In this case, it becomes more important to select appropriate objects to broadcast in a limited contact duration.

**B. Sensitivity Analysis**

We next discuss how PrefCast performs in different environments with respect to the object size, the number of users, the transmission range, and the buffer size. Fig. 5 demonstrates the impact of the file size on the utility contribution using the INFOCOM06 trace. The figure shows that the advantage of our PrefCast increases when the file size grows up. This is because a larger file size, e.g., a music file or a video file, requires a longer transmission time. As a result, a forwarder needs to broadcast a set of objects that can benefit the network the most in a limited contact duration.

Fig. 6 shows the impact of the number of users on the utility contribution using SLAW. We choose SLAW in this simulation because we can adjust the number of users in a synthetic trace. When there are more users that can help forward objects, each user can quickly get the objects that it is interested in. Therefore, the figure shows that the utility improvement is limited when there are more users in the network.

Fig. 7 shows the impact of the transmission range of each mobile device on the utility contribution using SLAW. Again, we choose SLAW in this simulation because the real traces only specify the contact event without the explicit geometric location information. However, the SLAM mobility model assigns each user a geometric location. Note that more users can overhear the transmission of an object from a forwarder
if the forwarder has a longer transmission range. Thus, when there are more members in a forwarder’s group, the forwarder generates a significant local utility, which dominates the benefit of future contributions. This is the reason that the performance gap between PrefCast and Local Utility decreases when the transmission range increases.

Finally, Fig. 8 compares the performance of PrefCast when the dropping probability is considered, as in Eq. (8), or not, as in Eq. (7). The figure shows that the performance gap between PrefCast and Local Utility is small when the buffer size is too small or too large. When the buffer size is too large, each user can almost cache all objects it downloads from other forwarders and does not need to drop any objects. On the contrary, when the buffer can only include a few number of objects, each user can only cache a few objects and, in general, acquire a low total utility. The impact of considering the dropping probability is therefore only obvious when the buffer is large enough but cannot cache all the objects.

VI. CONCLUSION

In this paper, we have proposed a preference-aware content dissemination protocol called PrefCast, which distributes content objects to maximally satisfying user preferences in a mobile social network. We formulate the problem as a maximum-utility forwarding scheduling model that allows each forwarder to determine its optimal forwarding strategy with consideration of future utility contributions, and mathematically analyze the performance gap between the optimal and greedy forwarding schedules. In addition, we propose a metric to predict the future utility contribution that a user can make. Such an estimate enables a forwarder to solve its optimal forwarding schedule in a distributed manner. The trace-based evaluation shows that PrefCast can provide a higher total utility than the preference-oblivious and local-preference-based dissemination schemes in a mobile social network.

REFERENCES