EventMe: Location-Based Event Content Distribution through Human Centric Device-to-Device Communications

Fengrui Shi*, Zhijin Qin†*, and Julie A. McCann*
*Department of Computing, Imperial College London, UK
†School of Computing and Communications, Lancaster University, UK

***Abstract*—Location-based information dissemination has become increasingly popular in the recent years. Extensive research work has been done on the matching of interested parties to event information via publish/subscribe systems. However, the rich content types of such location-specific data, especially when the data are presented in multimedia form, requires efficient methods with low cost to transfer the content to the subscribers. In this paper, the potential of utilising human centric device-to-device (D2D) communications to disseminate location-based event content is investigated. The human centric D2D data dissemination process is formulated as a task assignment problem, which can be modelled as a Integer Quadratically Constrained Quadratic Programming (IQCQP) problem. Since the IQCQP problem is in general NP-hard, a sub-optimal polynomial framework named EventMe is proposed, which is able to compute a solution with guaranteed lower bounds on data distribution capacity in terms of throughput. Through extensive evaluation using several real world datasets, it has shown that EventMe is able to improve the network throughput by 100%-500% compared to baseline methods. A prototype is developed and shows that it is practical to implement EventMe on mobile devices by generating minimal control data overhead.**

I. Introduction

With the prevalence of social networks and mobile devices, location-based publish/subscribe techniques [1–3] have become popular for disseminating information about location-based events, such as concerts, shows, exhibitions and sales. The ubiquitous cameras and multimedia systems (e.g. Periscope [4]) enable publishers to produce rich content (e.g. images or short videos) to promote their events and attract more participants. However, the expensive and increasingly congested cellular network in the urban areas makes it both costly and slow for subscribers to obtain the multimedia data, whose size could range from several Megabytes to even Gigabytes. Therefore, techniques to assist location-based event content distribution (LECD) are more than desired.

Human centric device-to-device (D2D) communications [5, 6] can be considered as a promising solution, which utilises human power to proactively collect or deliver data to event subscribers through cheap communication channels such as WiFi. Human centric approaches have been widely used in areas such as mobile crowdsourcing/crowdsensing (MCS) and involve recruiting so-called 'workers' to complete data collection and sensing tasks [7, 8], which are two critical tasks in various communication networks [9, 10]. In these systems, the following three types of roles, including requester, worker and MCS platform, are usually assumed:

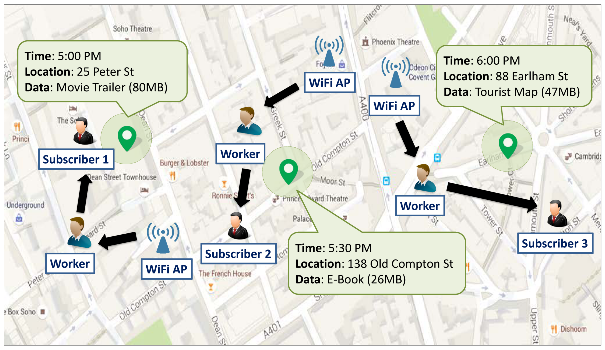


Fig. 1. An example of a human centric D2D-enabled LECD system.

- **Requesters** want to find some other people (workers) to help them complete tasks, e.g., collecting data, obtaining information. In order to do this, they would send requests including the details of the tasks to the MCS platform. In LECD systems, the requesters are therefore the subscribers who request location-based event content.
- **Workers** are those who would like to carry out tasks for others. They would indicate their interests to the tasks who information can be obtained from the MCS platform. In LECD systems, workers carry out data dissemination tasks for subscribers by adopting human centric D2D communications.
- **MCS platform** matches requesters' requests and workers' preferences and assigns tasks to workers based on criteria such as lower running cost or maximise benefits in terms of throughput or spatial coverage [7, 8]. In LECD systems, the platform assigns data distribution tasks to workers with the aim to maximise data distribution capacity of the system.

Based on these settings, the ideal is to design a task assignment algorithm to maximise the data distribution capacity (throughput) of the system. However, the existing methods [7, 8] used for solving task assignment problems in mobile systems cannot be directly employed due to the following challenges pertaining to the D2D-enabled LECD systems:

- **Resource constraints.** The size of the data delivered by workers may be limited due to 1) the limited storage of mobile devices and 2) time constraints for the workers to get requested data and fulfil tasks.
- **Shared event data by subscribers.** Event data are usually popular and shared by many subscribers. Therefore,

given limited resources, delivering the same data to more subscribers would increase the system capacity.

- **Shared tasks by workers.** More than one workers would work on a same task, thus causing the potential problems of delivering unnecessary duplicated data.

These challenges would potentially cause duplicated efforts by the workers and under-utilised storage and communication capacity of the workers. In this paper, we model the task assignment problem in D2D-enabled LECD systems as an integer quadratically constrained quadratic programming (IQCQP) problem. Due to the complexity of the IQCQP model and the difficulties to scale, an efficient framework called *EventMe* is proposed. To the best of our knowledge, EventMe is the first framework which models and solves the task assignment problem for D2D-enabled LECD systems. The detailed contributions of this paper include:

- We formulated the task assignment problem in D2D-enabled LECD systems as an integer quadratically constrained quadratic programming (IQCQP) problem, with the purpose of maximising the content distribution capacity in terms of throughput.
- Since the IQCQP problem is NP-hard in general, we develop EventMe, which computes a slightly sub-optimal solution by applying three polynomial-time algorithms to specially designed data structures. We also show that it guarantees a 1/2 lower bound for the optimal data distribution capacity of the system.
- To evaluate the practical performance of EventMe, we perform extensive trace driven simulations using various datasets. In addition, we develop an Android-based prototype to study the practical issues of the framework through real-life experiments. The results demonstrate that EventMe significantly improves the content distribution capacity compared with the baseline methods.

The remainder of this paper is organised as follows: Section II presents the model of task assignment in D2D-enabled LECD systems. The proposed EventMe framework is discussed in Section III. Evaluation results are presented in Section IV, and we finally conclude this paper in Section V.

II. Task Assignment Modelling in Human Centric D2D-Enable LECD Systems

Fig.2 shows an example of multiple workers performing data distribution tasks for multiple subscribers. Workers could obtain data from WiFi Access Points (AP), the maximum amount of data they would deliver to each subscriber are decided by either subscribers or workers in advance. The amount of data obtained from WiFi is decided by workers, which could be influenced by both the time constraints due to human schedules or the storage capacity of workers' devices. Since a subscriber may meet multiple workers and vice versa, same data could be received from different workers, thus resulting in data duplication and storage waste. In order to solve these issues, we formulate the task assignment problem as follows.

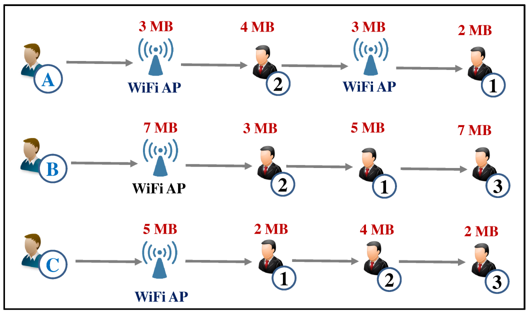


Fig. 2. An example of task completion sequence in a system with three subscribers (1, 2, 3) and three workers (A, B, C).

A. System Modelling and Problem Formulation

Table I summarises a list of essential notations used to present the model. $\mathcal{S}$ and $\mathcal{W}$ are the set of subscribers and workers. The event related data is divided into a set of data chunks $\mathcal{C}$, where each data chunk in $\mathcal{C}$ will be downloaded by the workers through WiFi APs, and transmitted to subscribers through D2D communications. $\mathbf{T}$ is used to regulate the temporal order of the tasks. $\mathbf{L}$ shows which data chunks a worker carries and $\mathbf{M}$ shows which data a worker delivers to a subscriber. $\mathbf{O}$ and $\mathbf{P}$ represent the constraints due to limited storage or subscribers' requests.

As presented below, an integer quadratically constrained quadratic program (IQCQP) can be used to model the data distribution task assignment problem in D2D-enable LECD systems. By solving the IQCQP problem, the data distribution capacity of the LECD systems could be maximised by avoiding the data duplication issues:

$$\underset{\mathbf{L},\ \mathbf{M}}{\textbf{maximize}} \sum_{k=1}^{K}\sum_{i=1}^{I}\sum_{j=1}^{J}\mathbf{M}_{i,j}^{k}\mathbf{L}_{i,j}^{k}, \tag{1a}$$

$$\textbf{subject to} \sum_{k=1}^{K}\mathbf{L}_{i,j}^{k} \le \mathbf{O}_{i,j}, \forall\ i,\ j, \tag{1b}$$

$$\sum_{k=1}^{K}\mathbf{M}_{i,j}^{k} \le \mathbf{P}_{i,j}, \forall\ i,\ j, \tag{1c}$$

$$\mathbf{L}_{i,j}^{k}(1-\mathbf{L}_{i',j}^{k}) = 0,\ \ \forall\ \mathbf{T}_{i,j} < \mathbf{T}_{i',j},\ i,\ j,\ k, \tag{1d}$$

$$\mathbf{M}_{i,j}^{k}\mathbf{M}_{i,j'}^{k} = 0,\ \ \forall\ i, j \neq j', k, \tag{1e}$$

$$\mathbf{L}_{i,j}^{k}, \mathbf{M}_{i,j}^{k} \in \{0,1\}, \forall\ i,\ j,\ k. \tag{1f}$$

The control variables in the problem are two 0-1 matrices $\mathbf{M}_{i,j}^{k}$ and $\mathbf{L}_{i,j}^{k}$ while $\mathbf{O}_{i,j}$, $\mathbf{P}_{i,j}$ and $\mathbf{T}_{i,j}$ are decided by subscribers and workers, which are known in advance. The details of each matrix is presented in Table I. We can see that the objective function (1a) aims to optimise the data delivery throughput by maximising the total number of unduplicated data chunks received by all subscribers. Constraint (1b) regulates the number of data chunks carried by an arbitrary worker $\mathcal{W}_j$ to be less than their desired amount. Constraint (1c) restricts the number of data chunks delivered from $\mathcal{W}_j$ to $\mathcal{S}_i$ to be less than their planned data size. Constraint (1d) means that in order to fully utilise either the WiFi or storage capacity, any

TABLE I
LIST OF NOTATIONS

$\mathcal{S}$	The set of all subscribers, $\mathcal{S} = \{\mathcal{S}_i\}$, where $i = [1, \ldots, I], I = \|\mathcal{S}\|$
$\mathcal{W}$	The set of all workers, $\mathcal{W} = \{\mathcal{W}_j\}$, where $j = [1, \ldots, J], J = \|\mathcal{W}\|$
$\mathcal{C}$	The set of all data chunks of the event content, $\mathcal{C} = \{\mathcal{C}_k\}$, where $k = [1, \ldots, K], K = \|\mathcal{C}\|$
T	A matrix $\in \mathbb{R}^{I \times J}$. $\mathbf{T}_{i,j}$ represents the time of performing data delivery task between the ith subscriber and the jth worker.
L	A 0-1 matrix $\in \mathbb{R}^{I \times J \times K}$. $\mathbf{L}_{i,j}^k = 1$ indicates that worker $\mathcal{W}_j$ has data chunk c_k when meeting subscriber $\mathcal{S}_i$ and vice versa.
M	A 0-1 matrix $\in \mathbb{R}^{I \times J \times K}$. $\mathbf{M}_{i,j}^k = 1$ indicates that worker $\mathcal{W}_j$ decides to deliver data chunk $\mathcal{C}_k$ to subscriber $\mathcal{S}_i$ and vice versa.
O	A matrix $\in \mathbb{R}^{I \times J}$. $\mathbf{O}_{i,j}$ represents the maximum data chunks worker $\mathcal{W}_j$ carries before performing tasks at $\mathbf{T}_{i,j}$.
P	A matrix $\in \mathbb{R}^{I \times J}$. $\mathbf{P}_{i,j}$ represents the planned amount of data to be delivered to subscriber s_i from worker $\mathcal{W}_j$.

given data chunk $\mathcal{C}_k$ can only be downloaded at most once by each $\mathcal{W}_j$ (i.e. without duplicate data). Similarly, constraint (1e) states that each subscriber $\mathcal{S}_i$ does not receive duplicated data chunks from the workers.

B. Achieving Lower Bounds with A Simplified Problem

It can be verified that problem (1a)-(1f) is an integer quadratically constrained quadratic program (IQCQP) [11] (the formulated problem in this paper is a special case of Mixed Integer QCQP since all control variables x are integers here). Since computing the optimal solution of the IQCQP problem is NP-hard in general, we aim to develop a slightly suboptimal but efficient practical solution to the problem (1a)-(1f) in polynomial time with explicit performance bounds, in order to have a scalable system. Therefore, the following constraints are added to the original IQCQP model given by (1a)-(1f).

$$\mathbf{L}_{i,j}^k \mathbf{M}_{i,j}^k (1 - \mathbf{L}_{i',j}^k \mathbf{M}_{i',j}^k) = 0, \tag{2a}$$

$$\text{where} \sum_{k=1}^{K} \mathbf{L}_{i,j}^k \mathbf{M}_{i,j}^k < \sum_{k=1}^{K} \mathbf{L}_{i',j}^k \mathbf{M}_{i',j}^k \forall i, j, k. \tag{2b}$$

Constraint (2a) states that if the number of data chunks ($\sum_{k=1}^{K} \mathbf{L}_{i,j}^k \mathbf{M}_{i,j}^k$) received by $\mathcal{S}_i$ from $\mathcal{W}_j$ is less than the data received by $\mathcal{S}_{i'}$ from $\mathcal{W}_j$ (as in (2b)), the data chunks received by $\mathcal{S}_i$ must be also received by $\mathcal{S}_{i'}$ (2a). For example, in Fig. 2, if subscriber $\mathcal{S}_1$ receives 3 data chunks $L_{1,j}$ from $\mathcal{W}_j$ and $\mathcal{S}_2$ receives 2 chunks $L_{2,j}$ from $\mathcal{W}_j$ as well, $\mathcal{L}_{2,j} \subset \mathcal{L}_{1,j}$. By adding the constraint given in (2a)-(2b), dimensionality of the problem is reduced from three to two. Constraint (2a) can eliminate the influence of the workers' decisions on the subscribers. In the original problem, the download and delivery decisions by one worker affect both the other workers and all subscribers, which produces an optimisation problem over three dimensions. As a result, the problem can be solved using a polynomial algorithm with $\mathcal{O}(I(J + I^2))$ as shown in section III[1].

III. EVENTME FRAMEWORK: ALGORITHMS WITH GUARANTEED LOWER BOUNDS AND POLYNOMIAL COMPLEXITY

In order to solve the formulated problem in the above, we propose an efficient framework called EventMe in this section. We also prove its correctness, the lower bound to the optimal data distribution capacity and computation complexity.

A. Overview

The workflow of EventMe is illustrated in Fig. 3(a), in which three algorithms 1) greedy allocation, 2) bipartite graph generation and 3) weighted maximum matching are involved:

1) Subscribers submit requests and workers choose tasks. After gathering workers' preference, the coordinator computes an initial Task Assignment (**TA**) structure.
2) **TA** is then updated using a greedy allocation algorithm as shown in Algorithm 1.
3) A Diff Matrix (**DM**) is obtained from the updated **TA** to generate Row Index Sets ($\mathcal{RIS}$), which includes Positive Row Index Sets ($\mathcal{RIS}^+$) and Negative Row Index Sets ($\mathcal{RIS}^-$). Here, $\mathcal{RIS}$ is used to check the possibility to make further improvement to **TA**.
4) Algorithm 2 is then used to generate a weighted bipartite graph *BG*, in which $\mathcal{V}^+$ and $\mathcal{V}^+$ are its two parts representing the sets in $\mathcal{RIS}^+$ and $\mathcal{RIS}^-$ respectively. The weight $w_{i,j}$ of the edge $e_{i,j}$ connecting $\mathcal{V}_{\rangle}^{+}$ and $\mathcal{V}_{|}^{-}$ is computed by $|RIS_i^+ \cap RIS_j^-|$.
5) Maximum weighted matching based Algorithm 3 is performed on *BG* to calculate the final **TA**.
6) Workers and subscribers are notified of the allocated tasks, i.e., what data, when and where they should download and relay based on the final **TA**.

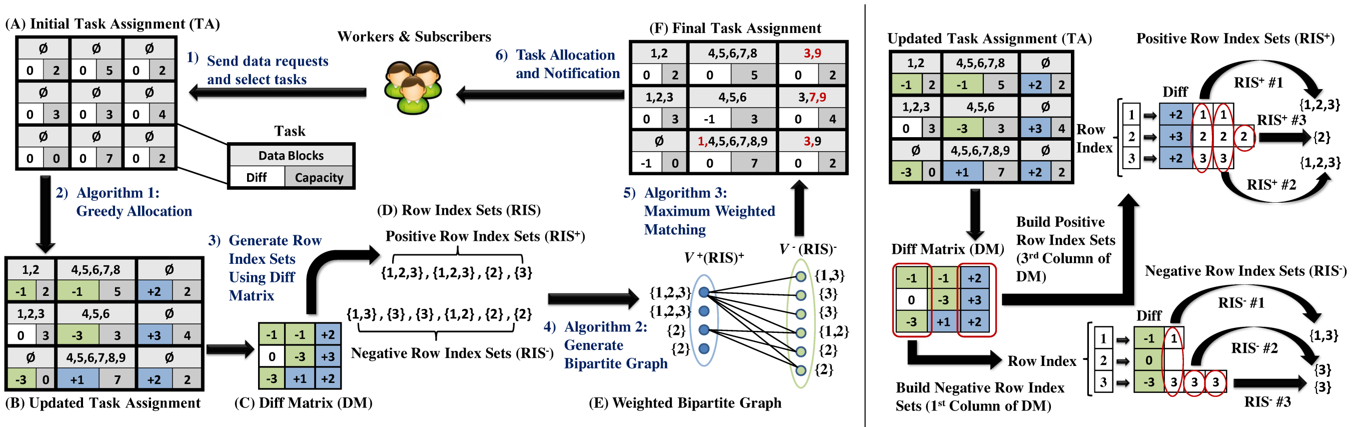
B. Data Structure

Several novel data structures are used in EventMe framework to reduce the computational cost and aid the organization of the procedures in a systematic way.

1) Task Assignment (TA): **TA** is used to store information related to the tasks between workers and subscribers based on the information from the matrices **P** and **O** from the constraint (1c) and (1b). **TA** is aligned in rows and columns and **TA**$_{i,j}$ is a 'Task' and its structure is shown in Fig.3(a)(A):

- **Data chunks** field in **TA**$_{i,j}$ is an array that records the data chunks to be transmitted from $\mathcal{W}_j$ to $\mathcal{S}_i$.
- **Capacity** field for **TA**$_{i,j}$ shows the maximum number of data chunks which can be transmitted from $\mathcal{W}_j$ to $\mathcal{S}_i$ when performing a task.
- **Diff** field is used to record the difference between 1) the number of assigned data chunks and 2) the maximum number of data chunks which can be assigned. It starts with 0 and can become positive or negative after performing Algorithm 1. Positive value means it has capacity remaining, negative means it misses certain data chunks compared to other tasks in the same column, while 0 indicates that the capacity for this task is fully

[1] This is different from relaxation in which an upper bound can be found for a maximisation problem with relaxation. However, a lower bound is obtained for a maximisation problem with more constraints.


(A) Initial Task Assignment (TA)
Workers & Subscribers
1) Send data requests and select tasks
6) Task Allocation and Notification
(F) Final Task Assignment
Task
Data Blocks
Diff
Capacity
2) Algorithm 1: Greedy Allocation
3) Generate Row Index Sets Using Diff Matrix
(D) Row Index Sets (RIS)
Positive Row Index Sets (RIS+)
{1,2,3}, {1,2,3}, {2}, {3}
{1,3}, {3}, {3}, {1,2}, {2}, {2}
Negative Row Index Sets (RIS-)
4) Algorithm 2: Generate Bipartite Graph
5) Algorithm 3: Maximum Weighted Matching
V+(RIS)+
V-(RIS)-
(B) Updated Task Assignment
(C) Diff Matrix (DM)
(E) Weighted Bipartite Graph
Updated Task Assignment (TA)
Positive Row Index Sets (RIS+)
Diff
Row Index
RIS+ #1
RIS+ #2
RIS+ #3
{1,2,3}
{2}
Diff Matrix (DM)
Build Positive Row Index Sets (3rd Column of DM)
Negative Row Index Sets (RIS-)
RIS- #1
RIS- #2
RIS- #3
{1,3}
{3}
Build Negative Row Index Sets (1st Column of DM)

(a) EventMe work flow

(b) Illustration of Step 3) in Fig.3(a): Generate Row Index Sets using Diff Matrix **DM**

Fig. 3. EventMe framework

```
Algorithm 1: Greedy allocation
Input  : $\mathbf{TA}_{i,j}$ (capacity fields $c_{i,j}$), data chunks size $S$
Output: Diff Matrix $\mathbf{DM}_{i,j}$, updated $\mathbf{TA}_{i,j}$
$j \leftarrow 1$
maximum capacity: $c_{max} \leftarrow 0$
maximum capacity for column $j$: $c_{max,j} \leftarrow 0$
while $c_{max} + c_{max,j} < S$ do
    $c_{max,j} \leftarrow \max_i(c_{i,j})$
    for $i \in [1 : I]$ do
        $\mathbf{TA}_{i,j} \leftarrow [c_{max} \ldots \min(c_{max} + c_{i,j}, S)]$
        $\mathbf{DM}_{i,j} = c_{i,j} - min(S - c_{max}, c_{max,j})$
    end
    $c_{max} \leftarrow c_{max} + c_{max,j}$
    $j \leftarrow j + 1$
end
```

```
Algorithm 2: Bipartite graph generation
Input  : Diff matrix $\mathbf{DM}_{i,j}$, $i = 1 \ldots I$, $j = 1 \ldots J$
Output: $\mathcal{RIS}^+$, $\mathcal{RIS}^-$
for $j \in [1 : J]$ do
    generate $\mathcal{RIS}^+$
    while $\max_i(\boldsymbol{DM}_{i,j}) > 0$ do
        append $\{i| \; \forall \mathbf{DM}_{i,j} > 0\}$ to $\mathcal{RIS}^+$
        $\mathbf{DM}_{i,j} \leftarrow \mathbf{DM}_{i,j} - 1, \forall \mathbf{DM}_{i,j} > 0$
    end
    generate $\mathcal{RIS}^-$
    while $\min_i(\boldsymbol{DM}_{i,j}) < 0$ do
        append $\{i| \; \forall \mathbf{DM}_{i,j} < 0\}$ to $\mathcal{RIS}^-$
        $\mathbf{DM}_{i,j} \leftarrow \mathbf{DM}_{i,j} + 1, \forall \mathbf{DM}_{i,j} < 0$
    end
end
```

used. "Diff" fields in **TA** are gathered to generate Diff Matrix (**DM**).

2) Diff Matrix (DM): After running the greedy allocation algorithm, Diff Matrix (**DM**) is generated from the computed "diff" fields of **TA** and then used to generate the bipartite graph using Algorithm 2.

3) Positive and Negative Row Index Sets ($\mathcal{RIS}$, $\mathcal{RIS}^+$, $\mathcal{RIS}^-$): Improvement on **TA** can be made using **DM**: $\mathbf{TA}_{i,j}$ with positive "diff" gets the data which are missed by negative "diff". For each column of **DM**, the positive/negative Row Index Set ($\mathcal{RIS}^+, \mathcal{RIS}^-$) are formed with the row indices of **TA** with positive "diff" fields. The integer value in the "diff" field indicates the maximum number of times this row index can be used in generating row sets. Moreover, because of the constraint (2a), each $\mathcal{RIS}$ can be used only once to make sure that data chunks received by subscribers from the same worker have the "subset" relationship.

C. Operations in EventMe Framework

We use the example in Fig. 2 to show how the framework works. To simply the presentation, only three workers, three subscribers and nine data chunks are used.

Based on the initial **TA** in Fig. 3(a)(A), the greedy allocation Algorithm 1 generates a new **TA** (Fig. 3(a)(B)) by assigning not yet assigned data chunks to $\mathbf{TA}_{i,j}$ for each column from left to right without violating two constraints: 1) "data chunks" field in each $\mathbf{TA}_{i,j}$ is a subset or superset of the other tasks in the same column; 2) the number of assigned data chunks does not exceed the task capacity. In this example, the algorithm stops after assigning the data chunks in the second column. The "diff" field also updated. For instance, "diff" in $\mathbf{TA}_{2,2}$ is 3 since it is assigned three data chunks but the maximum number of assigned data chunks in the same column is six. The constructed Diff Matrix (**DM**) is shown in Fig. 3(a)(C).

DM is then used as an input to Algorithm 2. By building a bipartite graph, improvement can be made to perform matchings between tasks $\mathbf{TA}_{i,j}$ with $\mathbf{DM}_{i,j} > 0$ and $\mathbf{DM}_{i,j} < 0$. Here, $\mathbf{DM}_{i,j} < 0$ means that there is at least another task $\mathbf{TA}_{i',j}$ in the same column having data chunks that $\mathbf{TA}_{i,j}$ does not have. Therefore, $\mathbf{TA}_{i,j'}$ in the same row with $\mathbf{DM}_{i,j'} > 0$ (remaining capacity) can be used to get those missing data chunks and improve the results. It can be imagined as tasks with $\mathbf{DM}_{i,j} < 0$ "borrow" capacity from tasks with $\mathbf{DM}_{i,j} > 0$

Algorithm 3: Maximum weighted matching improvement

```
Input  : $\mathcal{RIS}^+$, $\mathcal{RIS}^-$
Output: Final Task Assignment $\mathbf{TA}_{i,j}$
$\mathbf{W}_{m,n}$: weight matrix for pairwise intersection of $\mathcal{RIS}^+$
  and $\mathcal{RIS}^-$, $m = 1 \dots |\mathcal{RIS}^+|$, $n = 1 \dots |\mathcal{RIS}^-|$
for each $\mathcal{RIS}^+_m$ in $\mathcal{RIS}^+$ do
    for each $\mathcal{RIS}^-_n$ in $\mathcal{RIS}^-$ do
        $\mathbf{W}_{m,n} = |\mathcal{RIS}^+_m \cap \mathcal{RIS}^-_n|$
    end
end
$\mathcal{M}$: matching by applying Hungarian algorithm to $\mathbf{W}$
for each $M$ in $\mathcal{M}$ do
    append data from $\mathbf{TA}_{i,j}$ with $M_{RIS^-}$ to $\mathbf{TA}_{i',j'}$ with
      $M_{RIS^-}$, where $i, j \in M_{RIS^+}$ and $i', j' \in M_{RIS^+}$
end
```

or "lend" otherwise. For example, $\mathcal{S}_1$ only receives two data chunks from $\mathcal{W}_1$ and it misses data chunk #3 compared to $\mathcal{S}_2$. The only way it can receive data chunk #3 is to use $\mathcal{W}_3$, whose task with $\mathcal{S}_1$ has extra capacity with $\mathbf{DM}_{1,3} > 0$.

In (2a), it states that data received by subscribers from the same worker must be subset or superset of each other. When "lending" capacity for a **TA** task in a certain column, all other tasks in the same column with $\mathbf{DM}_{i,j} > 0$ must also "lend" the same data chunk. For example, if using the third column to receive the missing data chunk #3 for $\mathcal{S}_1$, all $\mathbf{TA}_{i,3}$ in the third column must also hold data chunk #3. Therefore, we group the "diff" fields for each column using $\mathcal{RIS}$. Besides, each row index only appears once in a certain set in $\mathcal{RIS}$.

The bipartite graph BG with $\mathcal{V}^+$ and $\mathcal{V}^-$ is constructed using $\mathcal{RIS}^+$ and $\mathcal{RIS}^-$. All $\mathcal{V}^+$ ($\mathcal{RIS}^+$) are in left partition and all $\mathcal{V}^-$ ($\mathcal{RIS}^-$) in the right. The bipartite graph is fully connected. The weight $w_{i,j}$ of each edge $e_{i,j}$ connecting $\mathcal{V}^+_{\rangle}$ and $\mathcal{V}^-_{|}$ is the number of row indexes two vertices have in common. When two $\mathcal{RIS}$ have same row index, capacities can be reallocated for that row in **TA**. The generated bipartite graph is shown in Fig. 3(a)(E) (some of the edges and weights are not shown due to the limited spaces).

The weighted maximum matching for bipartite graph is performed on a cost matrix to improve the results. It is shown in Fig. 3(a)(F) that the subscribers get 19 data chunks from greedy allocation and receive six more through the matching. The overall assigned data chunks is thus 25, which is suboptimal but close to the optimal value 27 for the original optimization problem.

D. Analysis of the Algorithms

Analysis is performed to show that 1) the algorithm gives a correct solution and has a $\frac{1}{2}$ lower bound to the optimal maximization problem; 2) the complexity is $\mathcal{O}(I(J + I^2))$.

Theorem 1 Solution provided by performing Algorithm 1, 2 and 3 provides a $\frac{1}{2}$ lower bound to problem described by (1a) to (1f). (To prove Theorem 1, we first introduce two Lemmas)

Lemma 1 The solution obtained from weighted maximum matching does not violate any of the constraints.

Proof By definition, $\mathcal{RIS}$ cover the maximum possible number of rows in one column. If there is a new data chunk assigned to that column, all rows will have it. Therefore, it does not violate the solution of the **TA**. □

Lemma 2 Algorithm 3 gives a $\frac{1}{2}$-approximation ratio to the solution after greedy allocation is performed.

Proof Each $\mathcal{RIS}^-$ for $\mathbf{DM}_{i,j} < 0$ can be covered by $\mathcal{RIS}^+$ by using each $\mathcal{RIS}^+$ only once. The wasted row indices (useful row indices which can cover other $\mathcal{RIS}^-$) in each $\mathcal{RIS}^+$ therefore make the solution suboptimal. We can prove by contradiction. If it cannot achieve an $\frac{1}{2}$ lower bound, there will be at least one $\mathcal{RIS}^+_p$ with more than half of the row indices wasted, which means that $\mathcal{RIS}^+_p$ can be used to cover some $\mathcal{RIS}^-_q$ which are not covered by any $\mathcal{RIS}^+$. However, for $\mathcal{RIS}^-_q$, the number of the uncovered negative row indices cannot exceed the number of covered ones, which means it does not waste more than half of its useful row indices. Otherwise, it will violate the maximum matching condition since the overall matching can be improved. □

Proof of Theorem 1 Based on Lemma 1 and Lemma 2, suppose the optimal solution is o, the achieved results from greedy allocation is g and the improvement from matching is m. The overall approximation ratio for the combined algorithm is $\frac{g+m}{o}$. Since greedy allocation does not cause any wasted capacity, $o - g - m < \frac{o}{2} \Leftrightarrow \frac{g+m}{o} > \frac{1}{2}$. □

Theorem 2 The computational complexity of EventMe is $\mathcal{O}(I(J + I^2))$.

Proof of Theorem 2 The greedy allocation is a linear algorithm with a complexity of $\mathcal{O}(J)$. Bipartite graph generation has outer loop of size J and inner loop of size I, so its complexity is $\mathcal{O}(J \cdot I)$. The weighted maximum matching has complexity of $\mathcal{O}(I^3)$ [12]. Therefore, total complexity is $\mathcal{O}(I(J + I^2))$. □

IV. Trace Driven Simulations and Prototype Implementation

In this section, we evaluate the network throughput achieved by EventMe in the D2D-enabled LECD system. Network throughput is defined as the amount of data delivered from workers to subscribers. Two baseline approaches are compared to EventMe:

- **Random task allocation (Random)** method randomly allocates task data to be delivered by workers.
- **Greedy task allocation** greedily choose the data to be delivered for a task. It is similar to the greedy allocation method introduced in Algorithm 1.

In the following, we first introduce the datasets, performance metrics, and evaluation parameters used in the evaluation. Then the stimulation results and prototype implementation are presented to verify the proposed framework.

A. Datasets

The trace driven simulation uses the following two datasets:

1) **GPS traces dataset** include over 400,000 pedestrian mobility data of 187 users collected in four locations [13]. This dataset provides the mobility data to simulate

the real-life mobility scenarios in the cities, which is important to evaluate the performance of EventMe because usually people are more likely to choose tasks without deviating too much from their schedule.

2) **Youtube video dataset** [14] provides the typical range of data size that the system must process and communicate. According to the dataset for $\sim$300,000 videos, the length of popular videos (views higher than one million) is mostly between one and ten minutes. With decent quality (480p - 1080p), we choose a range of 50 - 250 MB for the data size parameter in the simulation.
3) **WiFi access point (AP) data** [15] provide information of the typical coverage (density) of WiFi AP in urban areas. In the simulation, we consider a density of 20%, i.e., mobile users have access to WiFi 20% of the time.

B. Evaluation Parameters

We introduce three evaluation parameters, which potentially impact the performance of EventMe:

- **Data size** refers to event content size. Since video data are used in the simulations, a range of 50 - 250 MB will be expected as discussed earlier in section IV-A.
- **The range of workers' movement** indicates the geographical radius the workers choose to complete tasks. In the simulations, we evaluate worker movement range between one to five km, to show the effect of both small and large movement range.
- **The number of workers** impacts the duplication in terms of tasks preference in the system (many workers choose same tasks). Thus to evaluate its influence, we indicate the worker number as a percentage of total participants in the system from 10% to 90%. The remaining participants are subscribers.

C. Simulation Results

The simulation is performed using a Java implementation. We evaluate the performance of EventMe against random (labelled as 'Random' in the figures) and greedy task allocation (labelled as 'Greedy' in the figures) schemes based on the network throughput performance metric.

Figure 4 shows the average network throughput w.r.t. worker movement range, worker percentage, and data size, respectively. As seen from Fig. 4, EventMe can achieve an average throughput 500% higher compared to the random scheme and 25%-100% higher than the greedy scheme. Different evaluation parameters also show distinct impact on the metrics: 1) a stable throughput is achieved with varying worker movement range; 2) the network throughput is smaller with less workers due to limited number of tasks each worker is able to perform. While when number of workers increases, the throughput increases more; 3) the throughput increases when the data size is larger, which indicates the effectiveness of EventMe distributing data with large size. Overall, EventMe consistently outperforms the other competing methods.

The averaged network throughput performance value in Fig. 4 requires many parameters configurations. Therefore, the cumulative distribution function (CDF) of the network throughput performance is illustrated in Fig. 5, which shows that EventMe performs consistently better than the other methods given most of the percentiles of the distribution in terms of achieved network throughput. However, when workers' availability is low ($\approx$20%), greedy method tends to perform better than EventMe at the lower percentile of the CDF, which indicates a higher variance of the performance under low worker percentage.

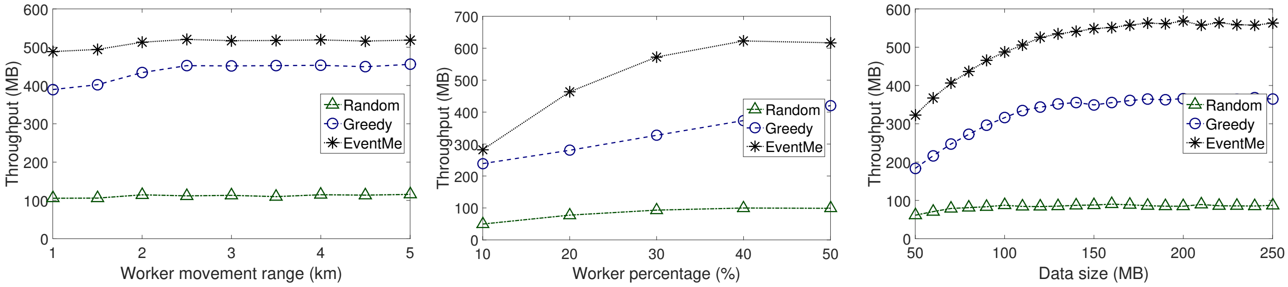
D. Prototype Implementation

We implemented EventMe as a mobile application on Android devices. The prototype is aimed to 1) show that various components of EventMe, such as event subscription, mobile D2D communication and task allocation, can work together, and 2) measure the communication overhead caused by task allocation. In the application, workers and subscribers can view on-going events and indicate their needs for data and tasks. After workers are assigned tasks, both workers and subscribers will be notified of the expected task completion information. In order to assist subscribers and workers to complete tasks, the application also shows a visual map guide to illustrate where and when the data delivery will take place, as shown in Fig. 6.

EventMe requires communications of control data between the smartphone application and the server during task completion. Therefore, to justify the its practicality, the communication overhead of EventMe is measured "in the wild" to be compared with the achieved network throughput. During the experiments, we recruit ten people, four as workers and six as subscribers, to use the application to distribute a 100 MB video clip for a virtual event. In this experiment, 186 MB throughput is achieved by EventMe with only 2.79 MB (1.5%) control data generated. It shows that the communication overhead is extremely low compared to the data delivery capacity achieved during the experiment.

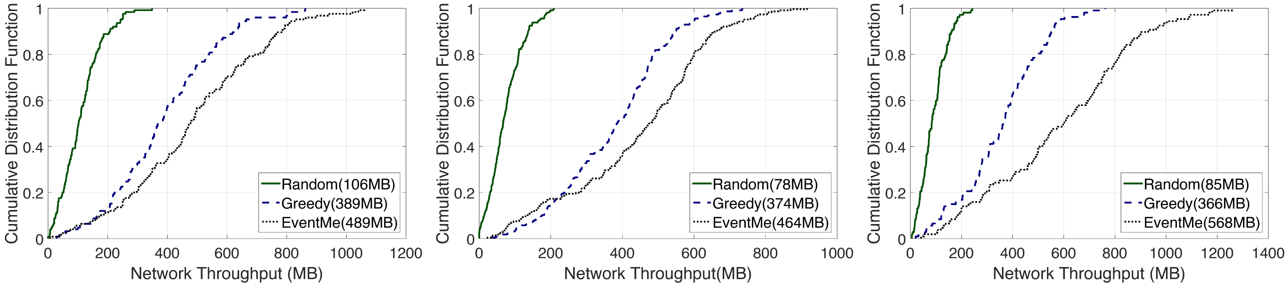
V. Conclusion

In this paper, we investigated the human centric D2D-enabled location-based event content distribution (LECD) systems, in which event subscribers obtain event related data from workers through D2D communications. We formulated the task assignment problem in the data distribution system as the integer quadratically constrained quadratical program (IQCQP) with the purpose of maximising data distribution throughput. To solve the problem, we proposed EventMe framework, which computes a minimally sub-optimal solution in polynomial time with guaranteed lower bound. By adopting the real-world GPS traces, simulation results showed that the proposed EventMe framework can improve the network throughput by 20% to 100%. Furthermore, by implementing a prototype in Android smartphones, we evaluated the communication overhead of the system in terms of control data compared to the data delivery capacity and showed that minimal data overhead was generated.



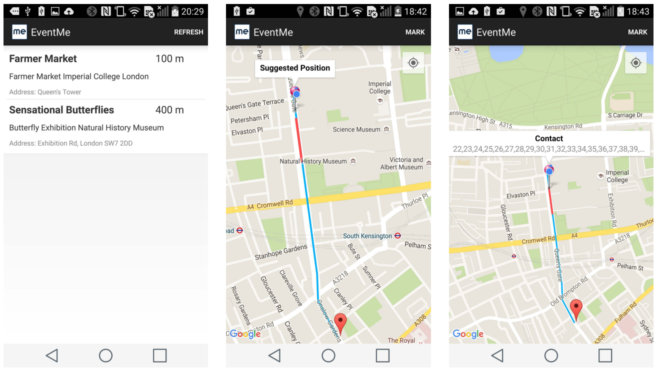
(a) Worker movement range (b) Worker percentage (c) Data size

Fig. 4. Average network throughput with different evaluation parameters



(a) Worker movement range = 1 km (b) Worker percentage = 20 % (c) Data size = 200 MB

Fig. 5. CDF of network throughput (average throughput) with different evaluation parameters



20:29

EventMe REFRESH

Farmer Market 100 m

Farmer Market Imperial College London

Address: Queen's Tower

Sensational Butterflies 400 m

Butterfly Exhibition Natural History Museum

Address: Exhibition Rd, London SW7 2DD

18:42

EventMe MARK

Suggested Position

18:43

EventMe MARK

Contact

22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,...

(a) Event list (b) Visual guide (c) Assigned task

Fig. 6. Screenshots of EventMe prototype

References

[1] Guo L, Zhang D, Li G, et al. "Location-aware pub/sub system: When continuous moving queries meet dynamic event streams". in *Proc. ACM SIGMOD*. Melbourne, Victoria, Australia, May 2015, pp. 843-857.

[2] Bao J, Mokbe M F, Chow C Y. "Geofeed: A location aware news feed system". in *Proc. IEEE ICDE*. Washington, DC, USA, Jul 2012, pp. 54-65.

[3] Li G, Wang Y, Wang T, et al. "Location-aware publish/subscribe". in *Proc. ACM KDD*. Chicago, Illinois, USA, Aug 2013, pp. 802-810.

[4] Periscope. https://www.periscope.tv/

[5] Ganti R K, Ye F, Lei H. "Mobile crowdsensing: current state and future challenges". *IEEE Commun. Mag.*. Nov 2011, vol. 49, no. 11, pp. 32-39.

[6] Open Garden: https://www.opengarden.com/

[7] Cheung M H, Southwell R, Hou F, et al. "Distributed time-sensitive task selection in mobile crowdsensing". in *Proc. ACM MobiHoc*. Hangzhou, China, Jun 2015, pp. 157-166.

[8] Zhao Q, Zhu Y, Zhu H, et al. "Fair energy-efficient sensing task allocation in participatory sensing with smartphones". in *Proc. IEEE INFOCOM*. Toronto, ON, Canada, Apr 2014, pp. 1366-1374.

[9] Qin Z, Gao Y, Plumbley M, et al. "Wideband Spectrum Sensing on Real-time Signals at Sub-Nyquist Sampling Rates in Single and Cooperative Multiple Nodes", *IEEE Trans. Signal Process.*, Jun 2016, vol. 64, no. 12, pp. 3106-3117.

[10] Liu Y, Qin Z, Elkashlan M, et al. "Enhancing the Physical Layer Security of Non-orthogonal Multiple Access in Large-scale Network", *IEEE Trans. Wireless Commun.*, Mar 2017, vol. 16, no. 3, pp. 1656-1672.

[11] Belotti P, Kirches C, Leyffer S, et al. "Mixed-integer nonlinear optimization". *Acta Numerica*, May 2013, vol. 22, pp. 1-131.

[12] Munkres J. Algorithms for the assignment and transportation problems. *SIAM Journal*, 1957, vol. 5, no. 1, pp. 32-38.

[13] Injong R, Minsu S, Seongik H, et al. CRAWDAD dataset ncsu/mobilitymodels, doi:10.15783/C7X302, 2009.

[14] Cheng X, Dale C, Liu J. "Statistics and social network of youtube videos". in *Proc. IEEE IWQoS*. Enschede, Netherlands, Jun 2008, pp. 229-238.

[15] WiFi Map: https://www.wifimap.io/