



2017

IoT-Based Cost Saving Offloading System for Cellular Networks

Zhuojun Duan

the Department of Computer Science, Georgia State University, Atlanta 30303, GA, USA.

Mingyuan Yan

Department of Computer Science and Information Systems, University of North Georgia, 82 College Cir, Dahlonega 30597, GA, USA.

Qilong Han

the College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China.

Lijie Li

the College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China.

Yingshu Li

the Department of Computer Science, Georgia State University, Atlanta 30303, GA, USA.

Follow this and additional works at: <https://tsinghuauniversitypress.researchcommons.org/tsinghua-science-and-technology>



Part of the [Computer Sciences Commons](#), and the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Zhuojun Duan, Mingyuan Yan, Qilong Han et al. IoT-Based Cost Saving Offloading System for Cellular Networks. *Tsinghua Science and Technology* 2017, 22(4): 379-388.

This Research Article is brought to you for free and open access by Tsinghua University Press: Journals Publishing. It has been accepted for inclusion in *Tsinghua Science and Technology* by an authorized editor of Tsinghua University Press: Journals Publishing.

IoT-Based Cost Saving Offloading System for Cellular Networks

Zhuojun Duan, Mingyuan Yan, Qilong Han*, Lijie Li, and Yingshu Li

Abstract: Nowadays, with the new techniques available in hardware and software, data requests generated by applications of mobile devices have grown explosively. The large amount of data requests and their responses lead to heavy traffic in cellular networks. To alleviate the transmission workload, offloading techniques have been proposed, where a cellular network distributes some popular data items to other wireless networks, so that users can directly download these data items from the wireless network around them instead of the cellular network. In this paper, we design a Cost Saving Offloading System (CoSOS), where the Internet of Things (IoT) is used to undertake partial data traffic and save more bandwidth for the cellular network. Two types of algorithms are proposed to handle the popular data items distribution among users. The experimental results show that CoSOS is useful in saving bandwidth and decreasing the cost for cellular networks.

Key words: Internet of Things (IoT); cellular networks; cost minimization

1 Introduction

Traditionally, when a cellular network receives a request for a data item from a user, it transmits the data item to the user directly and charges them for this service. Most requests of voicemail and text messages require instant responses from the cellular network and data traffic generated by them^[1] is less than 300 petabytes monthly and increases slowly. However, nowadays, with the new techniques available in hardware and software, mobile devices (like smartphones and tablets), which support playing movies, reading ebooks online, and searching useful

information, can be regarded as compact laptops. Most of the mobile devices need to connect to a cellular network and download data items through it. Data traffic generated or requested by new emerging applications on mobile devices is different from previous voice calls and text messages. In 2014, data traffic generated by new applications is large^[2,3], reaches to more than 2000 petabytes monthly and will grow significantly. Because the data item is transmitted by the cellular network itself, like 3G or LTE, this process will occupy the bandwidth and generate transmitting costs. Compared with traditional voicemail and text messages, the new types of applications can tolerate transmission delays to a certain degree. In other words, users have low-time sensitivity levels to these data items when they use these new mobile device applications.

On the other hand, most mobile devices equipped with communication techniques (wireless interface, RFID tags) can communicate with each other for common goals, motivating a novel paradigm: the Internet of Things (IoT)^[4,5]. IoT is defined as a wireless network^[6,7] formed by electronic devices, vehicles, and other embedded items with a software system, sensors, and network connectivity that can sense and exchange data. In reality, IoT can be used in a wide range of areas: the healthcare service industry, firefighting, safer

• Zhuojun Duan and Yingshu Li are with the Department of Computer Science, Georgia State University, Atlanta 30303, GA, USA. E-mail: zduan2@student.gsu.edu; yli@gsu.edu.

• Mingyuan Yan is with Department of Computer Science and Information Systems, University of North Georgia, 82 College Cir, Dahlonega 30597, GA, USA. E-mail: mingyuan.yan@ung.edu.

• Lijie Li and Qilong Han are with the College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China. E-mail: lilijie@hrbeu.edu.cn; hanqilong@hrbeu.edu.cn.

* To whom correspondence should be addressed.

Manuscript received: 2016-11-18; revised: 2017-01-17; accepted: 2017-01-22

mining production, and so on. Furthermore, most nodes in IoT can store and forward data. Thus, we can offload some data traffic in IoT to alleviate the data traffic burden, decrease cost, and increase net profit for cellular networks. The technique of offloading in IoT not only allows cellular network to have more bandwidth to respond to new coming requests but also decreases the transmission cost for cellular networks^[8–11]. For example, if node *a* encounters node *b* which has stored some popular data items, *a* can download its interested data items from *b* directly, instead of receiving the data through the cellular network. Participation in the offloading system causes resource consumption (battery and storage) to the nodes in IoS, however, cellular networks will compensate for this. The compensation can be monetary rewards or coupons and we will discuss this in future works.

In this paper, we design a new Cost Saving Offloading System (CoSOS) for cellular networks based on the IoT, shown in Fig. 1. Assuming a set of users with mobile devices, all the devices could connect to a cellular network and form an IoT network by their Bluetooth interfaces. To release the workload, a subset of users is selected as the seed users. The cellular network distributes some popular data items in its buffer so that the seed users can respond to some data item requests directly within the IoT. When receiving a data item request from a user, the cellular network will decide whether to let the user search and download the data from the IoT. This is called the downloading decision process. When a user successfully downloads the requested data item from the IoT within a tolerant delay, the transmitting cost will not be generated and the cellular network could have more bandwidth to make responses to other requests. Otherwise, the cellular network should transmit the requested data item at the deadline, in this way, the Quality of Service

(QoS) is guaranteed. These two results generate two different costs for the cellular network. So, an accurate downloading decision process is critical to the system.

Implementing this system needs to solve the popular data items distribution problems. First, we need to choose the subset of seed users to store the popular data items. With more seed users, more requests will be satisfied within the IoT and this system will perform better. The cellular network should make some compensations to each seed user. However, more seed users mean more expense of the cellular network. Hence, seed users' selection should provide a balance between performance and cost. Second, users just grant part of their storage space for the cellular network and the space of each user to carry popular data items is limited. Thus it is possible that one user cannot store all data items in its buffer. For each user has been chosen to store popular data items, we need to choose the optimal subset of popular data items to be put into the user's buffer. Contributions of this paper can be concluded as follows:

(1) We establish a CoSOS for cellular networks. In this system, we combine the cellular network with the IoT to decrease the data traffic burden. As far as we know, we are the first one to offload data traffic to IoS.

(2) We prove that the popular data items distribution is an NP-hard problem, propose two types of algorithms, and apply the Poisson process to make the downloading decision.

(3) Simulations are conducted to verify the performance of our system, demonstrating that the system proposed in this paper could decrease the cost and leave more bandwidth for the cellular network and guarantee the service of quality simultaneously.

The remainder of this paper is organized as follows: Section 2 presents related works. In Section 3, we form the popular data items distribution problem. Algorithms are proposed in Section 4. Then the cost saving offloading system and its working process are shown in Section 5. Last two sections provide the experimental results and conclusion.

2 Related Work

Many works have been conducted for offloading problems. Authors of Refs. [12–14] proposed several methods to select seed users in wireless networks. In Ref. [12], the global and neighborhood VIPs are proposed to form the delegation. Global VIP focuses on users that are globally important in the network,

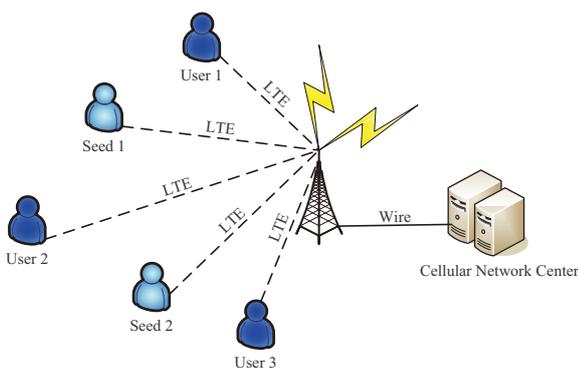


Fig. 1 Overview of the offloading system.

and neighborhood VIP selects users that are important within their social communities. In Ref. [15], the authors studied how to disseminate the data items into predefined seed users considering the buffer size of each seed user and the size of each offloading data item. In Ref. [16], a routing framework named SocialCast is proposed for the publish-subscribe function. It uses social interaction to select the best information carriers. In our new offloading system, we concentrate on the popular data items distribution problem which requires solving the users' selection problem and distributing the data items among selected seed users.

In Refs. [17, 18], the authors attempted to study the tradeoff between the amount of traffic being offloaded and the users' satisfaction, and proposed an incentive framework to motivate users to leverage their delay tolerance for 3G traffic offloading. This paper uses the reverse auction to minimize the incentive cost given an offloading target and selects users with high delay tolerance and large offloading potential to offload in wireless networks. The authors of Refs. [19–22] focused on how to motivate selfish nodes to participate in forwarding. In this paper, we use the Poisson process to analyze users' mobility behavior and evaluate their abilities in downloading requested data items in the DTN.

3 Problem Formulation

Assume that a group of users (denoted by V) with mobile devices form an IoT. To improve the overall QoS, all the users in V are willing to contribute part of their buffers to store popular data items. Once a user is selected to carry popular data items, the cellular network will give the user a promotion M as a reward. Assume the total budget of the cellular network is predefined, as B , so it can support at most $K = \lfloor B/M \rfloor$ users. Here S ($|S| = K$) is used to denote the set of users chosen to store popular data items, called the seed set. For each $v_s \in S$, b_{v_s} is its buffer size to store popular data items. In addition, D is used to denote the set of popular data items. For each data item d_j , its size is l_{d_j} . Each user has different frequencies to request different data items in D . The larger the interest in one data item, the more often a user will request it. Let $I_{v_i}(d_j)$ represent the degree of interest of user v_i in a data item d_j ,

$$\begin{aligned} 0 \leq I_{v_i}(d_j) \leq 1, \quad v_i \in V, \quad d_j \in D, \\ \text{and } \sum_{d_j \in D} I_{v_i}(d_j) = 1 \end{aligned} \quad (1)$$

The larger number of users could download their interested data items from the IoT, more bandwidth will be saved in the cellular network. If a user successfully downloads its requested data in the IoT before the deadline, we say the user's request is covered by the IoT. The coverage of seed users is defined as the total amount of data successfully downloaded from the IoT. Thus, the objective of our offloading system is to select K seed users to store popular data items which can maximize the coverage. For simplicity, the problem of popular data items distribution can be formulated as

$$\begin{aligned} \text{Max } C(S, D) &= \sum_{v_s \in S} \sum_{d_x \in \varphi_{v_s}} \sum_{v_j \in V \setminus S} w_{v_s v_j} I_{v_j}(d_x), \\ \text{s.t. } S &\subset V, \quad |S| = K, \\ \forall v_s \in S, \quad &\sum_{d_x \in \varphi_{v_s}} l_{d_x} \leq b_{v_s} \end{aligned} \quad (2)$$

where φ_{v_s} is the subset of popular data items distributed in the buffer of seed user v_s and $w_{v_s v_j}$ denotes the encounter probability between seed user v_s and user v_j . Because the buffer of a seed user for storing popular data items may not be large enough for all data items in D , the optimal subset of D should be selected.

Then, we prove the hardness of this optimization problem by the reduction technique. We know that the 0-1 Knapsack problem is NP-hard, the optimization problem (2) can be reduced to the 0-1 Knapsack problem if we set the size of the seed users to 1 ($|S| = 1$). The problem can be rewritten as

$$\begin{aligned} \text{Max } \sum_{d_x \in D} c(d_x) \delta(d_x), \\ \text{s.t. } \delta(d_x) \in \{0, 1\}, \quad \sum_{d_x \in D} l_{d_x} \delta(d_x) \leq b_s \end{aligned} \quad (3)$$

where

$$c(d_x) = \sum_{v_j \in V \setminus S} w_{v_s v_j} I_{v_j}(d_x) \quad (4)$$

In Formula (3) we regard the buffer of seed user s as a bag, each $d_x \in D$ as an item, $c(d_x)$ is the value of d_x , representing the coverage of data item d_x if putting it in the buffer user s and $\delta(d_x)$ shows the decision of whether to put this item in the buffer of user s . In addition, l_{d_x} is the weight of item d_x , and the total weight of the bag is b_{v_s} . Obviously, Formula (2) is NP-hard. Table 1 lists the frequently used notations.

4 Algorithms

4.1 Contact probability

To solve Formula (2), we first introduce several types of

Table 1 Table of notations.

Notation	Description
V, v_i	Set of users and i -th user
S	Set of seed users
b_{v_s}	Buffer size of v_s
d_j	Data item
l_{d_j}	Size of data item d_j
$I_{v_i}(d_j)$	Degree of interest of user v_i in data item d_j
φ_{v_s}	Data items distributed in the buffer of seed user v_s
$w_{v_s v_j}$	Encounter probability between seed user v_s and user v_j
$\lambda_{v_s v_i}$	Average contact frequency between user v_s and v_i

methods to evaluate the weight between any two users in V . As in Section 3, the weight between seed user v_s and user v_i represents the influence from v_s to v_i , which is the metric to evaluate the contact probability of the two users^[23]. There are several ways to calculate the weight between two users as follows.

Contact Frequency (CF) The contact frequency between users is used directly as the weight. Research in Refs. [24, 25] shows that the contact probability of two users can be estimated by their history contact frequency $\lambda_{v_s v_i}$.

$$w_{v_s v_i} = \lambda_{v_s v_i} \quad (5)$$

Proportional Seed-contact Frequency (PSF) The contact probability of v_s and v_i is proportional to the contact frequency between them to the total contact frequency from all seed users to v_i , namely

$$w_{v_s v_i} = \frac{\lambda_{v_s v_i}}{\sum_{v_k \in S \setminus v_s} \lambda_{v_k v_i}} \quad (6)$$

Poisson Process (PP) The contact process of two users can be modeled as the Poisson process^[24]. The contact probability of two users can be calculated by Eq. (7), where $\lambda_{v_s v_i}$ is the average contact frequency between user v_s and v_i and T_{v_i} is the largest delay user v_i can tolerate for receiving its requested data item.

$$w_{v_s v_i} = 1 - e^{-\lambda_{v_s v_i} T_{v_i}} \quad (7)$$

Advanced Poisson Process (APP) Sometimes more than one user could provide the requested data item to a user, which leads to the overlapping problem. To avoid this, we propose the APP method,

$$w_{v_s v_i} = \left(\prod_{v_k \in S \setminus v_s} e^{-\lambda_{v_k v_i} T_{v_i}} \right) (1 - e^{-\lambda_{v_s v_i} T_{v_i}}) \quad (8)$$

Based on the experiments in Section 6, APP outperforms CF, PSF, and PP methods, so we adopt the APP method to calculate the weight of two users in the following sections.

4.2 Heuristic algorithms

Considering the NP-hardness of Formula (2), two types of algorithms, the Greedy Algorithm (GA) and the Two-Step Algorithm (TSA), are proposed. In the GA (shown in Algorithm 1), the seed users are selected iteratively by K rounds, as shown in lines 1–10. Initially, S is empty. A seed user with the largest coverage is found and put into S in each round. Specifically, in each round, for each candidate seed user v_j in $V \setminus S$, we first calculate the weight $c_{v_j}(d_x)$ of each data item $d_x \in D$, which indicates the coverage when data item d_x is put in the buffer of user v_j , as shown in lines 3–5. Then, dynamic programming is used to calculate the optimal data items subset φ_{v_j} in D which leads to the largest coverage $c_{v_j}(\varphi_{v_j})$ of user v_j , shown in line 6. At the end of each round, select the user with the largest coverage as the seed user and put the corresponding optimal data items subset into its buffer (lines 8 and 9). The complexity of the GA algorithm is $O(K|V|(\beta|D|))$, where β is the upper bound of the user's buffer size.

In our daily life, the contact events occur when any two users are even, hence, all user pairs share a similar contact frequency. For example, when people participate in the same activity, like a conference or a game. In this case, users' contact abilities have minor influence on its coverage. If we choose the user who has a buffer with a larger size, as a seed, it will cover more requests in the network. We propose the TSA algorithm, shown in Algorithm 2, to distribute

Algorithm 1 Greedy Algorithm

Input:

User set V , data items set D ,
 Number of seed users K ,
 $S = \emptyset$.

Output:

seed user set S ,
 $s.buffer, s \in S$.

- 1: **for** $k = 1$ to K **do**
- 2: **for** each user $v_j \in V \setminus S$ **do**
- 3: **for** each data item $d_x \in D$ **do**
- 4: calculate $c_{v_j}(d_x)$
- 5: **end for**
- 6: $(\varphi_{v_j}, c_{v_j}(\varphi_{v_j})) =$
 Knapsack($b_{v_j}, D, (c_{v_j}(d_1), \dots, c_{v_j}(d_{|D|}))$),
- 7: **end for**
- 8: choose v , whose $c_v(\varphi_v)$ is the largest, $v \in V \setminus S$,
- 9: put data item subset φ_{v_j} into v 's buffer.
- 10: **end for**

Algorithm 2 Two-Step Algorithm**Input:**

User set V , data items set D ,
 Number of seed users K ,
 $S = \emptyset$.

Output:

seed user set S ,
 $s.buffer, s \in S$.

1: **Step 1:**

2: choose K seed users with the largest buffer sizes from V as seed user set S .

3: **Step 2:**

4: **for** each s in S **do**

5: **for** each data item $d_x \in D$ **do**

6: calculate $c_s(d_x)$

7: **end for**

8: $\varphi_{v_j} =$

9: Knapsack($b_{v_j}, D, (c_{v_j}(d_1), \dots, c_{v_j}(d_{|D|}))$),

10: put data item subset φ_{v_j} into s 's buffer.

11: **end for**

popular data items when the behaviors of users are heterogeneous. In the first step, K users with the largest buffer size are chosen as seed users (as shown in line 2). Then, for each seed user, the subset of data items which can maximize its coverage to requests is obtained through dynamic programming (lines 5–9). The complexity of using dynamic programming to solve the 0-1 Knapsack problem is $O(K|V| + K(\beta|D|))$. Compared with the GA, the complexity has been reduced by approximate $|V|$ times.

5 CoSOS Design

Offloading popular data items to an IoT could save bandwidth and reduce the cost for cellular networks. The cost of the cellular network includes energy consumption, device wear and tear, and so on. In this section, we discuss the cost of the cellular network in the offloading system and try to further increase its net profit.

Practically, when a user sends a data item request to the cellular network and receives the requested data item successfully, it should make a payment. For each MB of data traffic received by users, the net profit of the cellular network is calculated as $F - \pi$, where F is the charge per MB data traffic and π is the cost per MB data traffic. On the basis that F for each MB data traffic remains constant, the smaller is the cost in transmitting, and the larger is the net profit. Let α represent the transmission cost per MB data traffic

through the cellular network. If data item $q.d$ by request q is transmitted through 3G directly, the cost of the cellular network is

$$\pi_1 = \alpha \times q.d.size \quad (9)$$

In the offloading system, when a user generates a request for a popular data item, it tries to download the target data item from the IoT, which leads to two results. First, the user may encounter any seed user carrying the requested data item within the deadline and download it successfully. Then, the cost of cellular network is

$$\pi_2 = \beta \times q.d.size \quad (10)$$

where β is the coupon per MB data traffic given to users who try to download target data item from the IoT. Second, if the user cannot download the requested data item before the deadline, the requested data item will be transmitted through the cellular network in the end. In this case, the cost of the cellular network is

$$\pi_3 = (\alpha + \beta) \times q.d.size \quad (11)$$

Comparing Eqs. (10) and (11), we observe that failure in downloading the target data item from the IoT causes more cost than transmitting it through 3G directly. Only in the condition that the user can download the data item from other users in the IoT successfully before the deadline, is the cost of the cellular network the lowest. For example, assume a source user v_i with request q has little potential in getting the requested data item from the IoT, and the cellular network still decides to let v_i search by itself, Eq. (11) will appear most likely. Thus, in this case, it is better to transmit the requested data item $q.d$ through the cellular network directly. Because all users in the IoT are moving with time, the cellular network cannot have the full knowledge of their contact events. Based on the work in Ref. [24], the contact events of a user with any seed user in the future can be predicted. In addition, the cellular network knows the distribution of popular data items in the IoT. Thus, for user v_i , who generates a data item request, whose probability in successfully downloading the requested data item from the IoT can be calculated as

$$p = 1 - \prod_{s \in S' \subset S} e^{-\lambda_{sv_i} T_{v_i}}, \quad \forall s \in S', q.d \in s.buffer \quad (12)$$

where S' is the set of seed users carrying $q.d$. The process of the CoSOS is shown in Fig. 2. In the CoSOS, the cellular network makes the downloading decision for v_i about its requested data item $q.d$ depending on the probability calculated by Eq. (12).

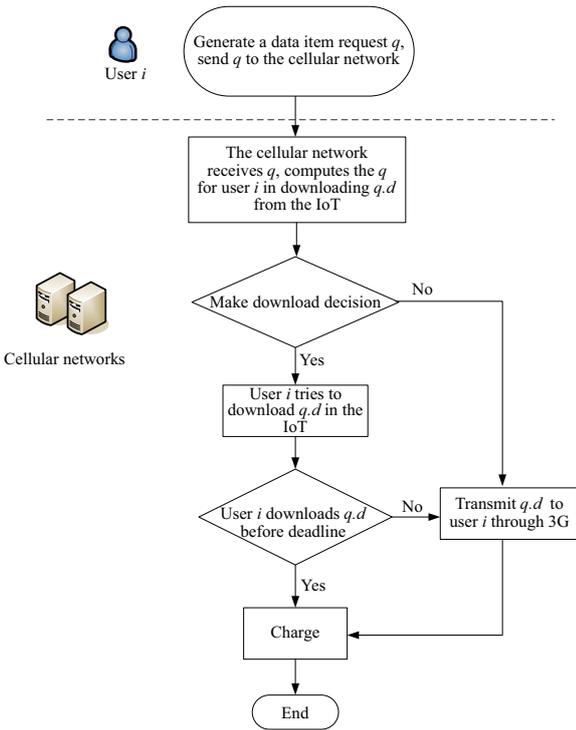


Fig. 2 Working process of the CoSOS.

6 Simulation and Results

In this section, we present the experimental results to demonstrate the performance of the CoSOS. Three real mobility traces and two synthetic movement traces are used to conduct the experiments. Among the three real traces, as shown in Table 2, MIT Reality trace records the connection of students or faculties in a campus, Infocom06 and Infocom05 are two traces that reflect the connection to participants of the Infocom conferences in 2006 and 2005, respectively. We use Random WayPoint (RWP) as a synthetic model and generate two movement traces RWP1 and RWP2 under different parameters. In RWP1 and RWP2, the size of scenario is $1500 \text{ m} \times 1500 \text{ m}$. The number of users is 100. In RWP1, we set 30 users who were picked up by 30 vehicles, whose speed is distributed randomly in the range of $[8 \text{ m/s}, 15 \text{ m/s}]$, the speed of the remaining users randomly was selected in the range of $[1 \text{ m/s}, 3 \text{ m/s}]$. In RWP2, all users are configured with a speed

randomly selected in the range of $[1 \text{ m/s}, 3 \text{ m/s}]$. We set $|D| = 30$, and the size of any data item $d_x \in D$ is distributed randomly in $[10 \text{ MB}, 100 \text{ MB}]$. The transmission range of each device carried by a user is 15 m. User's interest in each data item is generated according to Formula (1). The buffer size of each user is random in the range $[400 \text{ MB}, 1200 \text{ MB}]$. The cellular network obtains the contact frequency of any two users through a training process. During the training process, our system can learn contact frequencies of user pairs and get users' interest for popular data items. We use the ONE simulator to conduct the simulation experiments. The ONE simulator not only provides a good simulation environment for the IoT, but also supports generating the mobility models through configuration files and could import the real mobility traces.

The downloading ratio η indicates the percentage of data items traffic that can be downloaded from the seeds in the IoT. It is calculated by

$$\eta = \frac{\gamma}{\Theta} \quad (13)$$

where γ is the total size of the data items transmitted within the IoT, Θ is the total size of all data items requested by the users. In Section 4, the popular data items distribution is related to the weight of the users. The weight of any two users indicates their contact probability. Different ways to calculate the weight will influence popular data items distribution and the downloading ratio of our system. An accurate method to evaluate the weight of users is important. Figures 3 and 4 show the performance of the four types of weight computation methods in the downloading ratio under five mobility traces. From the results, we observe that the CF and the PSF methods are inferior to the PP and the APP methods. Even though the PSF method improves the offloading ratio, it could not reflect the contact of the users in an accurate way like the PP and the APP methods. The APP method could predict the behavior of users' movement in a more accurate way, leading to more data items downloaded in the IoT for the reason it avoids the overlapping problem in PP and

Table 2 Three real mobility traces.

Mobility trace	Communication type	Duration (d)	Number of devices	Training length (d)	Simulation length (d)
Infocom06	Bluetooth	3	98	3	3
Infocom05	Bluetooth	3	41	3	3
MIT Reality	Bluetooth	246	97	20	3

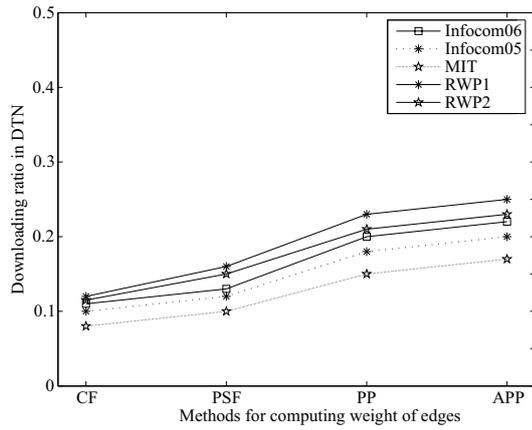


Fig. 3 Comparison of the downloading ratio under four ways to compute weight (GA).

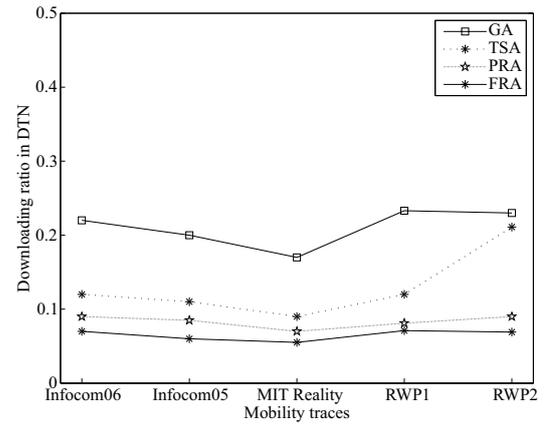


Fig. 5 Downloading ratio of the GA, the TSA, the PRA, and the FRA algorithms.

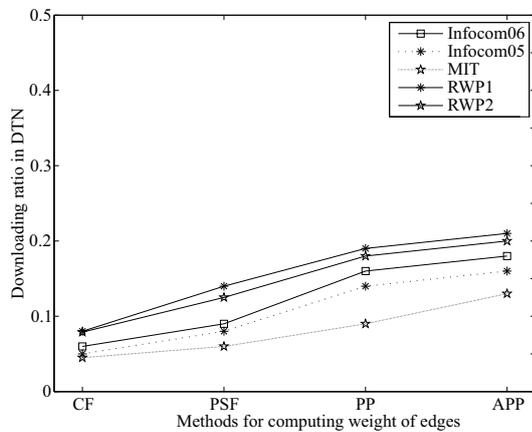


Fig. 4 Comparison of downloading ratio under four ways to computing weight (TSA).

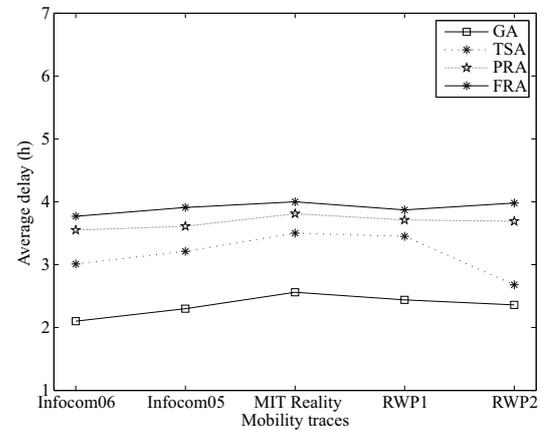


Fig. 6 Average delay of the GA, the TSA, the PRA, and the FRA algorithms.

selects the seed users to have the largest coverage in practice. So we choose the APP method to calculate the weight of any two users in the following experiments.

Then, we compare the performance of the GA and the TSA algorithms with two other algorithms, the PRA and the FRA, under five different mobility traces.

Partial Random Algorithm (PRA), in which the seed users set is chosen as the first K users with the largest buffer size. Then for each seed user, putting data items in D iteratively until no more data item can be put.

Fully Random Algorithm (FRA), where the seed users set is chosen randomly and the buffer of each seed user is filled as in the PRA. We set the number of seed users as 10 and set the tolerant delay as 24 h. Figures 5 and 6 show the results of the downloading ratio and average delay, respectively. We can see that our GA algorithm outperforms the other three algorithms in downloading ratio and average delay, which demonstrates these popular data items

distribution algorithms are useful. The performance of the TSA, under RWP2, is similar to the GA, which indicates that it is better to use the TSA when the contact frequencies among any pair nodes are similar, for the reason that the complexity of the TSA is lower than that of the GA. We know that, in the TSA and the PRA, the seed users are selected in the same way, but popular data items are distributed randomly in the PRA. The performance of the TSA is better than the PRA, which indicates data item distribution through the dynamic programming method could improve the coverage range in the IoT. Furthermore, the average delay of GA is about 2 h, which is smaller than that of the PRA and the FRA by about 40%. Conclusions from Figs. 5 and 6 indicate that the GA and the TSA outperform the PRA and the FRA, and the GA could be applied to most situations.

Then, several experiments are implemented to test the CoSOS performance in cost saving. For each MB of data traffic, its transmission costs α through the cellular

network. Any user who decided to search its requested data item in the IoT will receive coupon β per MB data traffic. To guarantee a net profit, the cost in the offloading system should not be larger than the cost in the system without considering offloading, that is,

$$\sum_{q \in Q_I \subset Q} \beta \cdot l_{q,d} + \sum_{q \in Q_{II} \subset Q} (\alpha + \beta) \cdot l_{q,d} \leq \sum_{q \in Q} \alpha \cdot l_{q,d} \quad (14)$$

where Q is the set of requests in this system, Q_I is the set of requests downloaded from their target data items from the IoT successfully, and Q_{II} is the set of requests failed to download their target data items from the IoT. The expected size of all popular data items is \bar{l} , so Formula (14) can be rewritten as

$$\beta \cdot \bar{l} \cdot \eta \cdot |Q| + (\alpha + \beta) \cdot \bar{l} \cdot (1 - \eta) \cdot |Q| \leq \alpha \cdot \bar{l} \cdot |Q| \quad (15)$$

After simplification, it becomes

$$\frac{\beta}{\alpha} \leq \eta \quad (16)$$

So $\alpha \cdot \eta$ can be regarded as the upper bound of β .

Then we try to verify the performance of the CoSOS. In the CoSOS, we use the GA method to implement the popular data items distribution and compare it with the other three systems. The first two systems are similar to the CoSOS with some slight change.

CoSOS Without Prediction (CWP) For each request of popular data items, CWP will make the user try to download it from the IoT first. If the user cannot download its requested data items from IoT before deadline, the data item should be transmitted through the cellular network.

Random CoSOS (RC) For each request of popular data item, RC will randomly decide whether the user attempts to download it from the IoT or not.

Baseline Method (BM) This system has not been considered for offloading. That is, the cellular network will transmit all data items to users through 3G directly.

An experiment is used to test the average cost of the cellular network for each MB of data traffic in the CoSOS. We set the value of α as 1 unit, set β as 0.15 unit, and use the GA method to implement the popular data items distribution. Figure 7 shows the cellular network cost in the four systems. Here, BM is a baseline, where all data items will be transmitted through 3G directly. So the cost in BM is constant. We can see that the cost in the CoSOS and the CWP are smaller than the other two because both the CoSOS and the CWP offload some data burden to the IoT. Even though the RC also offloads popular data items to the IoT, its cost has not been saved obviously, because

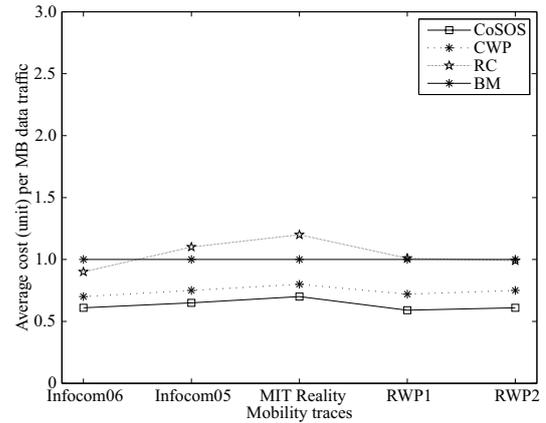


Fig. 7 Average cost per MB of data traffic of the CoSOS, the CWP, the RC, and the BM.

RC has not considered users' ability in downloading requested data items from the IoT. The CoSOS performs better than CWP, for the reason that the CoSOS will directly transmit the requested data items to the users with little ability to download them from the IoT. About 40% of the cost can be decreased in the CoSOS, which demonstrates it is useful for cellular networks.

At last, we test the performance of the CoSOS when the users can tolerate longer waiting time. The results of these experiments are shown in Fig. 8. We observe that when users' tolerant waiting time is 48 h, more than 50% cost can be reduced, because users have more opportunities to encounter seed users carrying their requested data items. However, the cost has not been obviously decreased when the tolerant waiting time is longer than 4 days. On the other hand, with longer waiting time, users may lose patience and will not join in the offloading system. So the value of waiting time should consider a balance between efficiency and practice.

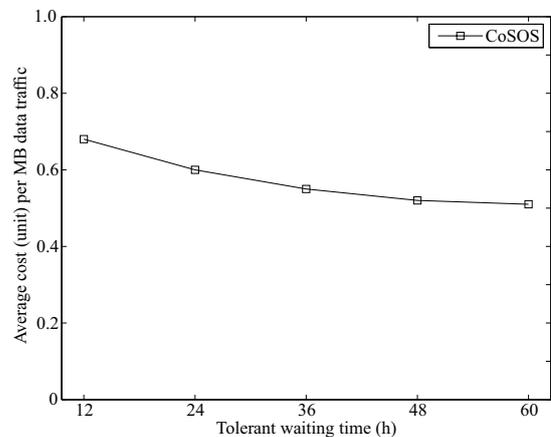


Fig. 8 Average cost per MB data traffic of CoSOS.

7 Conclusion

In this paper, we design a cost saving offloading system to reduce the traffic burden in cellular networks. In this system, we propose two algorithms to implement popular data items distribution. Then we adopt the Poisson process to predict users' ability in downloading the requested data items from the IoS, which could help the cellular network to decrease cost and increase net profits. Real trace-based simulation results show that our system could conserve the bandwidth and decrease the cost for the cellular network.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 61300207, 61370084, and 61502116).

References

- [1] Ericsson Mobility Report (2013, June). Available: <http://www.ericsson.com/res/docs/2013/ericsson-mobilityreport-june-2013.pdf>.
- [2] Ericsson Mobility Report (2014, June). Available: <http://www.ericsson.com/res/docs/2014/ericsson-mobilityreport-june-2014.pdf>.
- [3] Z. He, Z. Cai, and X. Wang, Modeling propagation dynamics and developing optimized countermeasures for rumor spreading in online social networks, in *Distributed Computing Systems (ICDCS)*, 2015, pp. 205–214.
- [4] D. Bandyopadhyay and J. Sen, Internet of things: Applications and challenges in technology and standardization, *Wireless Personal Communications*, vol. 58, no. 1, pp. 49–69, 2011.
- [5] X. Zheng, Z. Cai, J. Li, and H. Gao, An application aware scheduling policy for real-time traffic, in *Distributed Computing Systems (ICDCS)*, 2015, pp. 421–430.
- [6] X. Zheng, Z. Cai, J. Li, and H. Gao, A study on application-aware scheduling in wireless networks, *IEEE Transactions on Mobile Computing*, doi: 10.1109/TMC.2016.2613529.
- [7] L. Zhang, X. Wang, J. Lu, M. Ren, Z. Duan, and Z. Cai, A novel contact prediction-based routing scheme for DTNs, *Transactions on Emerging Telecommunications Technologies*, doi: 10.1002/ett.2889.
- [8] J. Zhuo, H. Lu, Z. Lu, and Y. Li, Scalable routing algorithm for partial matches in P2P systems, (in Chinese), *Journal of Tsinghua University (Science and Technology)*, vol. 44, no. 10, pp. 1389–1393, 2004.
- [9] Q. Zhao, L. Xu, Y. Mao, S. Leng, G. Min, J. Hu, and N. Najjari, Service-oriented wireless multimedia multicasting with partial frequency reuse, *Tsinghua Science and Technology*, vol. 21, no. 6, pp. 598–609, 2016.
- [10] X. Zheng, Z. Cai, J. Li, and H. Gao, Location-privacy-aware review publication mechanism for local business service systems, presented at the 36th Annual IEEE International Conference on Computer Communications (INFOCOM 2017), Atlanta, GA, USA, 2017.
- [11] M. Han, Y. Liang, Z. Duan, and Y. Wang, Mining public business knowledge: A case study in SEC's EDGAR, in *Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom)(BDCloud-SocialCom-SustainCom)*, 2016, pp. 393–400.
- [12] M. V. Barbera, J. Stefa, A. C. Viana, M. D. Amorim, and M. Boc, VIP delegation: Enabling VIPs to offload data in wireless social mobile networks, in *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, 2011, pp. 1–8.
- [13] B. Han, P. Hui, V. S. Kumar, M. V. Marathe, G. Pei, and A. Srinivasan, Cellular traffic offloading through opportunistic communications: A case study, in *Proc. 5th ACM Workshop on Challenged Networks*, 2010, pp. 31–38.
- [14] Z. Cai, Z. He, X. Guan, and Y. Li, Collective data sanitization for preventing sensitive information inference attacks in social networks, *IEEE Transactions on Dependable and Secure Computing*, doi: 10.1109/TDSC.2016.2613521.
- [15] Y. Li, G. Su, P. Hui, D. Jin, L. Su, and L. Zeng, Multiple mobile data offloading through delay tolerant networks, in *Proc. 6th ACM Workshop on Challenged Networks*, 2011, pp. 43–48.
- [16] P. Costa, C. Mascolo, M. Musolesi, and G. P. Picco, Socially-aware routing for publish-subscribe in delay tolerant mobile ad hoc networks, *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 748–760, 2008.
- [17] X. Zhuo, W. Gao, G. Cao, and Y. Dai, Win-Coupon: An incentive framework for 3G traffic offloading, in *19th IEEE International Conference on Network Protocols*, 2011, pp. 206–215.
- [18] Z. Duan, M. Yan, Z. Cai, X. Wang, M. Han, and Y. Li, Truthful incentive mechanisms for social cost minimization in mobile crowdsourcing systems, *Sensors*, vol. 16, no. 4, p. 481, 2016.
- [19] L. Anderegg and S. Eidenbenz, Ad hoc-VCG: A truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents, in *Proc. 9th Annual International Conference on Mobile Computing and Networking*, 2003, pp. 245–259.
- [20] W. Wang, X. Li, and Y. Wang, Truthful multicast routing in selfish wireless networks, in *Proc. 10th Annual International Conference on Mobile Computing and Networking*, 2004, pp. 245–259.
- [21] J. Li, Z. Cai, M. Yan, and Y. Li, Using crowdsourced data in location-based social networks to explore influence maximization, in *The 35th Annual IEEE International Conference on Computer Communications, IEEE INFOCOM 2016*, 2016.
- [22] Z. He, Z. Cai, J. Yu, X. Wang, Y. Sun, and Y. Li, Cost-efficient strategies for restraining rumor spreading

in mobile social networks, *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2789–2800, 2016.

- [23] L. Zhang, X. Wang, J. Lu, M. Ren, Z. Duan, and Z. Cai, A novel contact prediction based routing scheme for DTNs, *Transactions on Emerging Telecommunications Technologies*, doi: 10.1002/ett.2889.
- [24] G. Liu, J. Krishnamani, R. Sunderraman, and Y. Li, Prediction-based routing with packet scheduling under

temporal constraint in delay tolerant networks, presented at the IEEE 32nd International Performance Computing and Communications Conference (IPCCC), 2013, pp. 1–7.

- [25] W. Gao, Q. Li, B. Zhao, and G. Cao, Multicasting in delay tolerant networks: A social network perspective, in *Proc. Tenth ACM International Symposium on Mobile Ad hoc Networking and Computing*, 2009, pp. 299–308.



Zhuojun Duan received the MS degree from Shaanxi Normal University in 2011. She is currently a PhD candidate in the Department of Computer Science at Georgia State University. Her research interest is wireless network and social network.



Mingyuan Yan received the BS degree in computer science and technology and the MS degree in information security from Wuhan University, China, in 2008 and 2010, respectively. In 2012, she received another MS degree in computer science from Georgia State University. She is currently an assistant professor at

University of North Georgia. Her research interests include data management and protocol design in wireless networks, influence maximization, and information dissemination in mobile social networks. She is also interested in other topics such as information security and big data management. Mingyuan Yan is an IEEE member and an IEEE COMSOC member.



Lijie Li received the MS degree in 2006 and PhD degree in 2012 from Harbin Engineering University, China. Currently, she is a lecturer in the College of Computer Science and Technology, Harbin Engineering University, China. Her research interests include privacy protection, social network, and pattern

recognition.



Qilong Han received the PhD degree in computer science from Harbin Institute University, China in 2006. He is currently a professor and Deputy Dean in the College of Computer Science and Technology, Harbin Engineering University. His research interests include data security and privacy, mobile computing, and distributed and networked systems. He has more than 70 publications as edited books and proceedings, invited book chapters, and technical papers in refereed journals and conferences. He is a senior member of CCF, and the Chair of CCF YOCSEF Harbin. He has served as programme committee members and co-chairs of a number of international conferences/workshops for areas including web intelligence, e-commerce, data mining, intelligent systems, etc.



Yingshu Li received the PhD and MS degrees from University of Minnesota Twin Cities in 2005 and 2003, respectively. She received the BS degree from Beijing Institute of Technology, China in 2001. She is currently an associate professor in the Department of Computer Science at Georgia State University. Her research interests include wireless networking, sensor networks, sensory data management, social networks, and optimization. Her research has been supported by the National Science Foundation (NSF) of USA, the National Science Foundation of China (NSFC), the Electronics and Telecommunications Research Institute (ETRI) of South Korea, and GSU internal grants. She is the recipient of an NSF CAREER Award.