

RESEARCH

Open Access



On multi-copy forwarding protocols for large data chunk dissemination in vehicular sensor networks

Peng Liu^{1†}, Yue Ding¹, Tingting Fu^{1†}, Xingfa Shen¹ and Jianjiang Li^{2*}

Abstract

Moving vehicles have been sensing all kinds of data on the road in which multimedia data possesses a large portion. These data is often forwarded to vehicles in a region of interest or the monitoring center in an opportunistic manner. With respect to the large volume content, the storage space of relay vehicles is becoming the bottleneck of achieving higher performance, e.g., a data chunk may be rejected or dropped due to insufficient storage of intermediate vehicles. Thus, previous work that only focuses on the delivery metric without considering the data size is not likely to work efficiently in the proposed scenario. As deploying stationary infrastructures is of very large cost and not feasible everywhere, in this paper, we focus on the inter-vehicle data forwarding problem with storage and communication capacity constraints. First, we considered the situation when the vehicles are distributed sparsely. The multi-copy routing challenge is modeled as a multiple knapsack problem. Then, it is extended to a dense scenario. An optimization to the broadcast data forwarding is investigated. With the real data trace, the experiments show that our scheme achieves better performance than the competitors in terms of delay and delivery ratio. A better balance between duplication and performance is also achieved by the multi-copy algorithm.

Keywords: Multi-copy, Large data chunk, Forwarding, Vehicular sensor network

1 Introduction

Vehicles have been proved to be very useful in sensing traffic data, road accidents, and surrounding information with their high mobility and wide coverage. Camera-enabled devices, such as car dash cameras and smart phones, are popular nowadays in vehicles. In addition to reporting real-time road information, many accidents are also witnessed by vehicles, such as the air crash in Taiwan, 2015. A car dash camera records the process when the plane hit the viaduct and fell into Keelung River [1]. Therefore, these moving vehicles have formed a new type of sensor network, i.e., vehicular sensor network. The data collected by vehicles is expected to be delivered to the vehicles moving in a region of interest or to the monitoring server (a sink vehicle, in general), in a timely manner.

With the strong support of the backbone network, such as abundant connected road side units and 4G cellular network, the multimedia content can be delivered to anywhere at anytime. However, to build such a network or to use it will incur a large cost [2]. Furthermore, sometimes the backbone network will not be available, such as in an undeveloped area, or a disaster site. To this end, the vehicle-to-vehicle communication will be a desirable substitute choice. Regarding the large volume of video content, the short encountering duration and limited storage buffer have become the crucial bottleneck in the opportunistic transmission, which is seldomly considered by previous work.

According to traditional research assumptions, data size is omitted so that each relay can response to infinite number of requestors. Therefore, a routing algorithm that works on a single pair of source-destination vehicles also works on multiple pairs. The main concern is to justify the ability of reaching the destination when the data carrier meets another vehicle, such as contact frequency, contact

*Correspondence: lijianjiang@ustb.edu.cn

[†]Peng Liu and Tingting Fu contributed equally to this work.

²Department of Computer Science and Technology, University of Science and Technology Beijing, 30 Xueyuan Rd., Beijing, 100083 China

Full list of author information is available at the end of the article

duration, social relationship, and network topology. However, in terms of large data chunks, when there are a lot of source vehicles doing transmission simultaneously, the storage of an intermediate vehicle may be filled up and cannot respond to a new request. As shown in Fig. 1a, assume each relay vehicle can only accommodate just one large data chunk. According to their planned trajectories, vehicle R_2 has better opportunity to meet both D_1 and D_2 while vehicle R_1 has some possibilities to meet D_1 . Therefore, when R_1 encounters R_2 , it should forward the audio clip p_1 to R_2 . However, the storage limitation would not let it happen. A substitute solution is that R_1 and R_2 exchange their carried data as shown in Fig. 1b. They finally deliver data to destinations. Thus, according to the figure, vehicle R_1 carries p_2 and R_2 carries p_1 is better than the situation that vehicle R_1 carries p_1 and R_2 carries p_2 , but not as good as p_1 and p_2 are both with R_2 . Taking the variety of data size and delivery metrics into consideration, the decision of data exchange or forwarding will be more complicated when multiple copies are involved. According to [3], the probability of successful delivery exponentially decreases when the data size increases.

Although road side units (RSUs) have been widely deployed in many vehicular ad hoc networks (VANETs), it is still with large cost to coordinate data forwarding in a centralized way. In this paper, we seek to improve the forwarding efficiency with the algorithms executed on individual vehicles in a distributed manner, which will work no matter there are RSUs or not. First, we consider the sparse distribution case and model the forwarding decision problem as a multiple knapsack problem. Each vehicle makes the forwarding decision based on three parameters, i.e., data size, forwarding metric per each data, and forwarding capacity window. Then, in the dense situation, vehicles forward their data chunks in a broadcast manner. However, due to the mobility of vehicles, a vehicle will have dynamic neighboring vehicles. Therefore, how to arrange the data chunks in the limited communication capacity

becomes a vital problem. Finally, we model it as a bipartite matching problem and propose an algorithm to solve it.

The main contributions of this paper are threefold:

- To the best of our knowledge, this is the first work that models multi-copy routing of large data chunks as a multiple knapsack problem and gives a greedy solution which has the overhead under control.
- We further consider the limited communication bandwidth during encountering of vehicles and apply a selection mechanism based on 0-1 knapsack algorithm for vehicles to choose the most priority data chunks on top of the results of the above multiple knapsack solution.
- We are the first to consider the multiple different data chunks forwarding optimization in the broadcast manner, an algorithm solving the bipartite matching with communication and storage capacity for data chunk, and relay vehicle assignment is proposed.

The rest of the paper is organized as follows: Section 2 gives the overview of related work. Then, we discuss the problem formulation and preliminaries in Section 3. Then, we give details of our method implementation in sparse case in Section 4. Section 5 further introduces the solution for dense case. The effectiveness and improvement of our method are shown with the experimental results on the real data trace in Section 6. Section 7 concludes this paper and gives the future perspective.

2 Related work

Although there are many studies on vehicular networks, the vehicular sensor network is quite different with the previous work. We consider it more like a mobile opportunistic network but with more trajectory restrictions. Data forwarding is one of the hot research areas in mobile opportunistic networks (MONs). The key challenge is how to select appropriate relays such that data can be

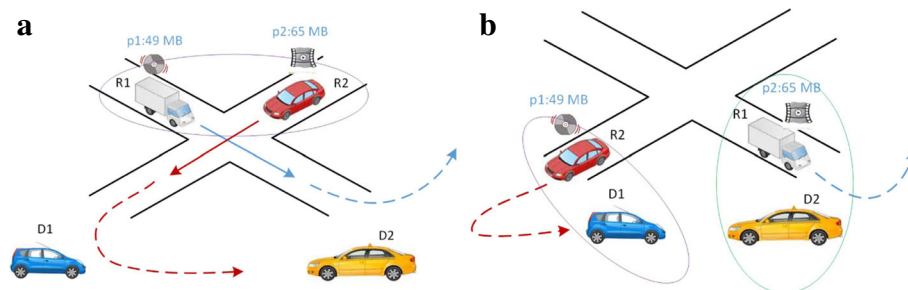


Fig. 1 An example to show the challenge in proposed scenario. The relay vehicle R_1 holds an audio clip p_1 to the destination vehicle D_1 while the relay vehicle R_2 tries to send a video clip p_2 to the destination vehicle D_2 . Each vehicle can only hold one data content due to storage limitation. The arrows show their moving directions. **a** First, R_1 meets R_2 . **b** Then, relays meet their destinations

forwarded to destinations with short latency and forwarding cost, where mobile nodes carry and forward messages upon intermittent contacts. Most routing methods are based on prior information such as the contact history [4], contact quality [5], and social-awareness [6, 7]. To further improve the efficiency of routings, network coding [8], multicast [9], multi path [10], UP-N-DOWN [11], and time-sensitive [12] methods are proposed. However, all these methods are based on the assumption that there is no limitation of storage space in nodes and data can finish transmission during a short period of contact. When considering the multimedia data which is very large in volume compared with small messages, the storage constraint will bring a lot of unexpected problems, especially worsens the end-to-end delay [13].

Large file dissemination is a prospective trend in vehicular networks. For example, in [14], the authors study querying multimedia data such as video and voice clips in hybrid vehicular networks which consist of vehicles that are capable of both infrastructure-less short-range communication and infrastructure communication. Also, in [15], the authors study querying binary large objects (blobs) such as video and voice clips in a network of vehicles communicating wirelessly. They focus on the efficient query of the content while none of them considered the data dissemination problem. Ref. [16] uses dynamic network topology graph to model the content downloading problem in vehicular networks. However, it needs the preemptive knowledge of vehicular trajectories and perfect scheduling of data transmissions, which is impractical in highly dynamical VANET since DNTG will become very complicated. Ref. [17, 18] prove that vehicle mobility follows certain patterns and can be used to predict the encountering of vehicles. One way to support large data chunk is to split, such as SADF [19], it is an automatic data packet dividing algorithm. To improve the delivery ratio, it cuts the large file into small segments according to the network quality and duration of contacts. In [20], large files are divided into data chunks. However, the relay-to-relay transmission is not considered. The carrier vehicle can only deliver the data chunk to the downloader directly. Another way is to add more storage, METHoD [21] implements a platform for distributing multimedia contents in delay-tolerant networks. It does not give solution on how to prevent memory overflow but adding a lot of external storage to help the big data application. Abdelmoumen et al. [22] analyze the adverse effect brought by the insufficiency of nodes' storage. By adding some fixed nodes with large storage space, the problem can be solve to some extent. However, in many cases, the data file is not allowed to be portioned and additional infrastructure is costly. Again, improving the data exchange efficiency would be another option. Zhao et al. [23] turn the problem of global optimizing of forwarding utility into the local

optimizing of forwarding utility upon nodes' encounter. The proposed cooperative forwarding is modeled as a 0–1 knapsack problem and solved by a greedy algorithm. In [24], the authors propose a dynamic segmented network coding scheme to efficiently exploit the transmission opportunity that is scarce in DTNs. In particular, they adopt a dynamic segment size control mechanism, which makes the segmentation adapt to the dynamics of the network.

In [25], the authors consider the challenge of disseminating large volume content in VANETs. The proposed Lifetime-aware Beacon-less Routing Protocol (LBRP) is built on the lifetime of the link and tries to obtain a durable path with less number of links. In their model, a source vehicle is one of the neighboring vehicles of the subscribing vehicle, which means they are not far from each other. However, in our model, the source and the destination are far from each other so that it is not possible to maintain a static routing path between them. The data must be forwarded in an opportunistic manner. Many real-world network applications can be modeled as knapsack problems and solved by approximate algorithms. In this paper, a multiple knapsack [26] problem is modeled extended from 0-1 knapsack problem. Generally speaking, a set of n items and m bins are given (knapsacks) such that each item i has a profit $p(i)$ and a size $s(i)$, and each bin j has a capacity $c(j)$. The goal is to find an allocation of items such that they have a maximum profit packing in the bins. Gao et al. model the access point assignment problem as a multiple knapsack problem in MONs [27]. However, they did not consider the dynamic value of each item in different bins.

3 System preliminaries and model

In this section, we introduce the system preliminaries and the model. To focus on the forwarding decision scheme, we omit the detail on the communication side, i.e., it is assumed that every inter-vehicle transmission is successful without data loss. Also, none of the vehicle is selfish. Each multimedia data generated by a source vehicle only has one destination. To decrease delivery latency and improve possibility of successful data delivery, in MONs, data replication is often adopted. However, apparently, it will bring more storage overhead and severe burden to the whole network, which may worsen the situation and increase the cost. Therefore, to make full use of the buffers of intermediate vehicles, we develop a duplication friendly strategy which is under data replication and drop control. In some work, data is fragmented into small pieces and sent via multiple path for a better possibility to reach the destinations. But in vehicular sensor networks, since the topology of the network is highly dynamic, to ensure the quality of service and guaranteed delay, a data file should be sent without split during vehicle-to-vehicle

transmission. All notations in this paper can be found in Table 1.

3.1 Objectives

The objective of this paper is to develop an efficient multi-copy unicast forwarding scheme for large data chunks in vehicular sensor networks. The first performance metric is the total amount of data bits that have been successfully delivered. The second performance metric is the delivery delay which is the average delivery latency of all source-destination pairs. The third performance metric is the overhead of data duplication.

3.2 System model

We consider a vehicular sensor network with N vehicles, where they move on roads and opportunistically encounter each other. Some of them raise requests of delivering sensed data chunks to other vehicles. Each source has only one destination, and we call a set of source vehicle and destination vehicle a source-destination pair. A vehicle could be acting dual roles at the same time. Each vehicle v_i is associated with a quality vector $(q_i^1, q_i^2, \dots, q_i^N)$, where q_i^j indicates the contact quality between vehicle v_i and v_j . In this paper, we adopt the total number of contacts during a period T as the metric of contact quality. The metric is widely used in MONs [4, 5] and has been proved very efficient. If v_j is the

destination, then the higher the q_i^j , the more possibility that v_i can encounter the destination. Therefore, we have the following definition.

Definition 1 *The delivery possibility is defined as the possibility of reaching a particular destination without further forwarding. For delivery possibility of a data chunk on vehicle v_i to the destination vehicle v_j , it is the same as q_i^j since the better the contact quality, the more opportunities to deliver the data.*

The challenge lies in the limitation of storage of intermediate vehicles. For example, the decision of forwarding a data chunk to or not to the relay vehicle will affect not only the delivery possibility of the current data chunk, but also the other data chunks which may also potentially use the vehicle as a relay. In a word, the forwarding of a data chunk is not an independent event anymore. Each data chunk delivering will compete the storage space with other ones of the current forwarding as well as the ones of future forwarding. Since it is of large cost to build a central controller which can schedule all the forwardings optimally, we will develop the distributed forwarding algorithm based on the local knowledge.

Definition 2 *The time to live, TTL, is defined as the remaining time a data chunk can survive in the network.*

Multi-copy data packet disseminating can shorten the average delay, but it also incurs additional cost [18, 28]. To control the overhead of data duplication, each copy will be set with a timer called TTL as indicated in Def. 2. The data chunk will be dropped from the network when the TTL is decreased to ZERO. The length of the TTL should be carefully designed. If it is too short, most replications will be dropped before they have a chance reaching the destination vehicle, which wastes a lot of storage resources during the dissemination. If it is too large, after a copy reaches the destination, there still remain many replications and possess the storage resources. It is not possible to calculate the optimal value of the TTL since it relies on many factors, such as the number of source-destination pairs, the size of the data file, the storage availability in relay vehicles, and encounter interval of vehicles. The recommendation value of the TTL is the average delivery latency of the single-copy algorithm. Then, the value could be slightly adjusted according to the dynamic feedback of the results. The copy of the data will also be dropped due to other reasons stated in the following section. Compared with the existing multi-copy-based work, our methods can achieve less requirements in communication and computing resources.

Table 1 Notations

q_i^j	Contact quality between vehicle v_i and v_j
TTL	Remaining time a data chunk can survive in the network
FCC	Forwarding communication capacity
$P_{rio}(d_k)$	Priority of a data chunk d_k
s_{d_k}	Size of data chunk d_k
c_i	Storage capacity of vehicle v_i
c_{FTW}	Length of selected contact duration in the dense case
X	Data chunk set $X = \{d_1, d_2, d_3, \dots\}$
Y	Candidate vehicle set $Y = \{N_1, N_2, N_3, \dots\}$
$w(x, y)$	Delivery utility for $x \in X$ to execute $y \in Y$
	Bipartite matching between $x \in X$ and $y \in Y$ where 1
$m(x, y)$	denotes an assignment, 0 denotes no assignment, and “-” indicates an initial or inexecutable status
$L(v)$	Labeling function of Hungarian algorithm [34], $v \in X \cup Y$
$L'(v)$	Previous record of L for any given $v \in X \cup Y$
S	Data chunk set in the current consideration of allocation, $\subseteq X$ Relay vehicles ($\subseteq Y$) that are reachable by data chunks $\in S$,
$N(S)$	i.e., $\{j \mid \exists m(i, j) = 0 \text{ or } 1, j \in N(S), i \in S\}$
T	Set of vehicles that have been saturated (assigned)
E_x^*	An alternating tree starting from x

3.3 Priority of data chunks

When there arises the need of dropping data and denial-of-relaying due to insufficient storage, especially in the multi-copy scenario, the priority of the data must be considered. The priority scheme should keep the fairness and efficiency of the data chunks. The priority could be based on many aspects such as the priority of the sender and the receiver, the significance of the data itself, and the elapsed time of the data file. However, many of the schemes could be very complex and hard to manage.

Since the main goal is to maximize the amount of successfully transmitted data bits, we assign the priority of a data chunk d_k as its data size (indicated as $P_{rio}(d_k)$), i.e., the larger the size is, the higher the priority is. Since forwarding decisions are only made between encountered vehicles, there is no need to maintain the global priority, so as to reduce the overhead. No further priority information is necessary to be kept. Priority only exists between encountering vehicles in a distributed and real-time manner. However, the replicate data copy will always have lower priority than those original ones without considering their size. Finally, we give the definition of the delivery utility DU as shown in Def. 3 for data chunk d_k from vehicle v_i to v_j .

Definition 3 *The delivery utility, DU, for a data chunk d_k from vehicle v_i to v_j is defined as the product of its priority and delivery possibility, i.e., $q_i^j(d_k) \times P_{rio}(d_k)$.*

3.4 Basic idea

When two or more vehicles encounter, each of them may contain several large data chunks. A data chunk has a different possibility of being delivered to the destination in different vehicles which depends on the contact quality between the data carrying vehicle and the destination vehicle. We cannot simply allocate the data chunk to the vehicle where it has the best possibility since there are storage bound. Therefore, we consider a joint allocation of data chunks in the buffer of both vehicles to maximize the entire delivery possibility of all data without storage overflow. Therefore, it is like there are two knapsacks and we try to put items in them to achieve maximal value under the constraint of storage. Besides that, the duration of each encountering is limited so that not all desired data chunks can be exchanged but also we need consider the communication constraint. In this paper, we model data chunk assignment problem as a multiple knapsack problem [26] where we try to achieve the maximal delivery possibility of all data chunks. When two vehicles meet, the first step is to exchange necessary information. Then, the second step is that all the data chunks are re-arranged according to their delivery possibility in different vehicles as a multiple knapsack problem. When the re-arrangement is applied,

the spare space may not be enough for the movement of these data chunks. Another limitation is the capacity of the channel, due to the bandwidth and the contact duration, there exists a bound of maximal amount of data that could be transmitted, which also applies in the dense case. In the multi-copy scenario, since the data copy with lower priority can be dropped, therefore, the storage allowance is not only the free space in the buffer but also data chunks to be dropped.

4 Large data forwarding with capacity constraints in sparse case

4.1 Multiple knapsack-based problem formulation

In sparse case, most communication will take place between two vehicles. Suppose the capacity of two encountered vehicle v_i and v_j is c_i and c_j respectively. The size of data chunk d_k is s_{d_k} . The delivery possibility of d_k on vehicle v_i is $q_i^{dest}(d_k)$. Assume two encountered vehicles have n data chunks in total. Then, the multiple knapsack problem can be formulated as:

$$\begin{aligned} & \max \sum_{k=1}^n q_m^{dest}(d_k) \times P_{rio}(d_k), \quad m = \begin{cases} i, & d_k \text{ on } v_i \\ j, & d_k \text{ on } v_j \end{cases} \\ \text{s.t. } & \sum_{k=1}^n s_{d_k} \leq c_i \text{ where } d_k \text{ is on } v_i & \text{(i)} \\ & \sum_{k=1}^n s_{d_k} \leq c_j \text{ where } d_k \text{ is on } v_j & \text{(ii)} \end{aligned}$$

Definition 4 *Generalized assignment problem, GAP*
INSTANCE: A pair (\mathbb{B}, \mathbb{D}) where \mathbb{B} is a set of m knapsacks and \mathbb{D} is a set of n items. Each knapsack $i \in \mathbb{B}$ has a capacity c_i , and for each item d_k , it has a size s_{d_k} and a profit (delivery possibility) $q_i^{dest}(d_k)$.

OBJECTIVE: Find a subset $\mathbb{U} \subseteq \mathbb{D}$ that has a feasible packing in \mathbb{B} and maximizes the profit of the packing.

The conditions (i) and (ii) are the constraints of the storage limitation on vehicle v_i and v_j respectively. We can convert it into a *generalized assignment problem* (GAP) (Def. 4) and apply the algorithm in ref. [29] to get the result. In [29], Shmoys and Tardos give a (1,2) bi-criteria approximation for Min GAP. A paraphrased statement of their precise result is shown as Theorem 1 [26].

Theorem 1 *Given a feasible instance for the cost assignment problem, there is a polynomial time algorithm that produces an integral assignment such that*

- cost of solution is $(1 - \varepsilon)$ OPT.
- each item k assigned to a knapsack i satisfies $s_k \leq c_i$, and
- if a knapsack's capacity is violated then there exists a single item that is assigned to the knapsack whose removal ensures feasibility.

4.2 Further consideration of communication capacity

The duration of encountering between two vehicles varies from time to time. Therefore, during each contact duration, only limited data can be transferred. In many cases, it is even smaller than the storage capacity allowance. Like in [27], data chunks are assumed to be indivisible, only those data chunks can be transmitted within the contact duration will be forwarded. To make it simple, we use term *forwarding communication capacity* (FCC) as defined in Def. 5 to represent the capacity of the contact. Note that the channel is a bidirectional one so that FCC only stands for one way capacity. Because of the constraint of FCC, vehicles may not have enough time to finish exchanging data chunks. Therefore, we must then achieve the optimal forwarding exchange under the additional FCC constraint. The FCC can be calculated if the mobility model of vehicles is known [30]. However, each contact would be quite unique so that we propose to compute FCC during information exchange, by using the transmission rate and the time that the vehicle stay in the other's communication range [27].

Definition 5 *The forwarding communication capacity, FCC, stands for the maximal amount of data bits that can be transferred one way during a contact. It depends on many factors such as communication bandwidth and contact duration. It can be calculated during information exchange as stated above.*

Suppose the capacity of FCC is noted as c_{FTW} , we have the new formulation of the problem.

$$\begin{aligned} \max \sum_{k=1}^n q_m^{dest}(d_k) \times P_{rio}(d_k), \quad m &= \begin{cases} i, & d_k \text{ on } v_i \\ j, & d_k \text{ on } v_j \end{cases} \\ \text{s.t. } \sum_{k=1}^n s_{d_k} \leq c_i \text{ where } d_k \text{ is on } v_i & \quad (i) \\ \sum_{k=1}^n s_{d_k} \leq c_j \text{ where } d_k \text{ is on } v_j & \quad (ii) \\ \max \sum_{k=1}^n s_{d_k} \leq c_{FTW} \text{ where } d_k \text{ is forwarded} & \\ \text{from } v_i \text{ to } v_j \text{ or viceverse} & \quad (iii) \end{aligned}$$

Equation 3) guarantees the exchanged data will not exceed the communication capacity between two vehicles.

We use Algorithm 1 to solve the multiple knapsack problem with the FCC constraint and a dynamic programming-based algorithm as illustrated in Algorithm 2. Finally, the problem will be solved by solving subproblems in a dimensionality reduction manner.

4.3 Multi-copy scenario

To get fast delivery, flooding-based schemes can achieve very short routing latency. However, taking data size into consideration, some data and their replications will soon occupy most of storage space and prevent other data from being relayed. In this section, we aim to reduce the cost of forwarding while retaining high routing performance

Algorithm 1 Large data chunk forwarding with FCC constraint based on Multiple Knapsack

Require: Data chunk list from both vehicles (including size, destination information), capacity size of two vehicles, size of the FCC

Ensure: an arrangement of data chunks in both vehicles

- 1: **FOR** the two encountering vehicles v_i and v_j **DO**
- 2: **IF** v_i or v_j is a destination to any current data chunks **THEN**
- 3: forward the data chunk to v_i and v_j ;
- 4: **ELSE**
- 5: get the list of the remaining data chunks
- 6: solve the arrangement as a multiple knapsacks problem;
- 7: **FOR** the data chunks in each direction **DO**
- 8: model the problem as a 0-1 knapsack problem where the one way data chunks are the items and the capacity of the FCC is the knapsack;
- 9: **END FOR**
- 10: exchange the data chunks according to above result;
- 11: **END IF**
- 12: **END FOR**

Algorithm 2 find solution matrix of Multiple Knapsacks problem

Require: Size set $\theta = \theta_1, \theta_2, \dots, \theta_\mu$, delivery possibility set $\gamma = \gamma_1, \gamma_2, \dots, \gamma_\mu$, capacity space C_1 and C_2

Ensure: Output solution matrix m

- 1: $n = \text{length}[\Omega]$
- 2: **FOR** $j = 0$ to unit length of $C_1 + C_2$ **DO**
- 3: $m[0, j] \leftarrow 0$
- 4: **END FOR**
- 5: **FOR** $\mu = 1$ to n **DO**
- 6: $m[\mu, 0] \leftarrow 0$
- 7: **FOR** $j = 1$ to unit length of $C_1 + C_2$ **DO**
- 8: $m[\mu, j] \leftarrow m[\mu - 1, j]$
- 9: **IF** $\theta_\mu \leq j$ **THEN**
- 10: **IF** $\gamma_\mu + m[\mu - 1, j - \theta_\mu] > m[\mu - 1, j]$ **THEN**
- 11: $m[\mu, j] \leftarrow \gamma_\mu + m[\mu - 1, j - \theta_\mu]$
- 12: **END IF**
- 13: **END IF**
- 14: **END FOR**
- 15: return m

by controlling the number and threshold of replications. In previous work, the *STOP* condition could be a fixed number of copies, a given time-to-live [31], or a dynamic threshold [32]. Different from previous work, we do not give any fixed number of copies since it is allowed to drop data replications in the forwarding process. Also

with the replication threshold, there will generate less copies in the network. Thus, the overhead is under control while performance is greatly improved. There are two main operations in the multi-copy scenario: one is the data replicating, and the other is data dropping.

As shown in Fig. 2, the solid line with arrow stands for data replication between vehicles. The dashed line with arrow stands for data forwarding between vehicles. The other line means there is no action between two encountered vehicles. There are many reasons that cause no action, such as no storage space, lower threshold, and lower delivery possibility. The source vehicle S meets R_1 and R_2 sequentially; they satisfy the replication condition so that S gives duplications of the data to both vehicles. Then, R_1 replicates the data to R_3 and forwards its data chunk to R_4 . Finally, the data chunk is delivered to destination vehicle D by R_4 . Other data duplications will be dropped when TTL decreases to 0 as seen in Fig. 2.

4.3.1 Data replicating

When two vehicles encounter, some data chunks may be replicated under given conditions. These conditions ensure the number of copies will not exceed the cost budget for performance improvement. Suppose when vehicle v_i meets vehicle v_j and v_i has the data chunk d_k while v_j does not have it. We describe these conditions as follows:

Rule 1: the vehicle without the data chunk can provide better possibility to reach the destination, i.e., $q_i^{dest}(d_k) \leq q_j^{dest}(d_k)$.

Even with the rule 1, v_i may encounter many vehicles that have better delivery possibilities. Therefore, the highest delivery possibility among all the vehicles v_i met will be recorded, say $h(d_k)$. If vehicle v_j is the first vehicle that has

better delivery possibility than v_i , then $h(d_k) = q_j^{dest}(d_k)$. The next vehicle must have higher delivery possibility than $h(d_k)$ before it gets a duplication.

Rule 2: the vehicle without the data chunk must have room to accommodate the duplication.

Here, the room can be a current free space, or a free space going to be released after data dropping. Data dropping happens when its TTL is 0 or it has lower delivery possibility than the prospective duplication.

Rule 3: the FCC has enough capacity for replicated data transmission.

Same as rule 2, there must be enough spare capacity for duplication data transmission before the data chunk can be replicated. The FCC must ensure it first serves the original data.

Rule 4: the original data chunk always stays in the vehicle with better delivery possibility.

The data replication operation must acquire entire information of all data chunks on both vehicles and follow above rules.

4.3.2 Data dropping

Since data can be replicated, other copy may reach the destination prior to it. Then, it should not possess the network resource anymore. In another aspect, the current data copy should be able to be removed when another data copy with higher delivery possibility appears. Therefore, data can be dropped under following rules:

Rule 5: original data chunk cannot be dropped unless it reaches the destination or has a 0 value TTL.

This rule is to make sure, for each data chunk, there at least exists one copy being carried in the network so that the delivery ratio can be guaranteed.

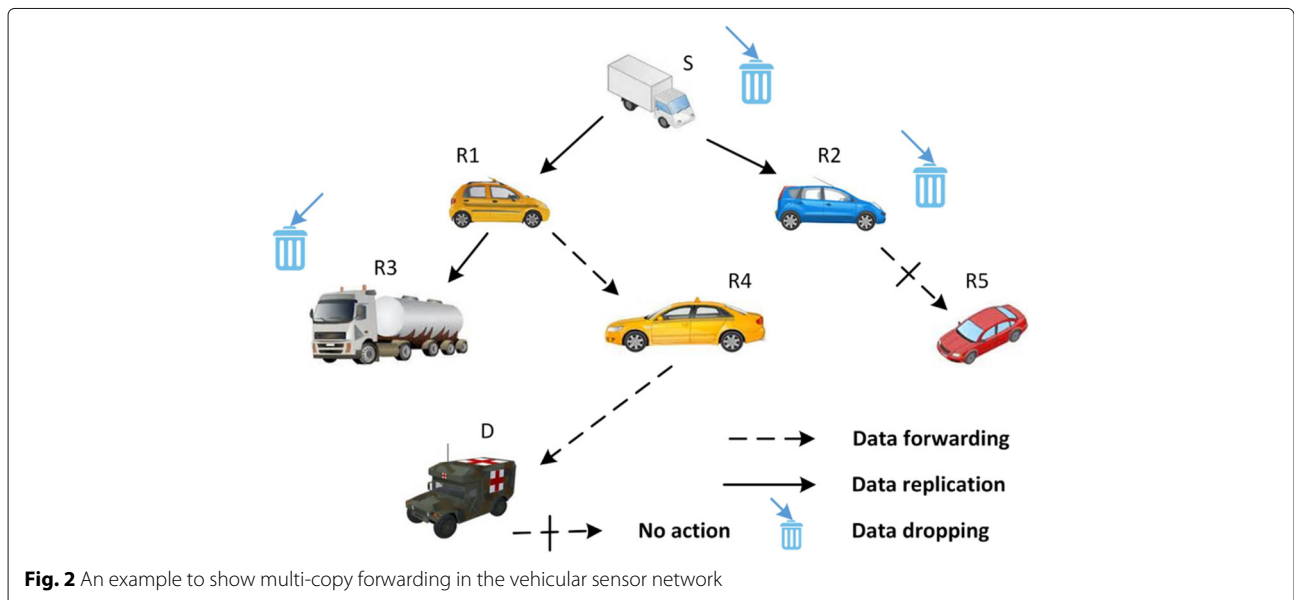


Fig. 2 An example to show multi-copy forwarding in the vehicular sensor network

Rule 6: when the TTL of a data copy decreases to zero, it will be dropped immediately.

This rule is kind of garbage collecting rule. Network resource will be released by those useless data chunks in the end.

Rule 7: combined with data replication rules, if data copies have to be dropped to give space to better data chunks, those with lowest delivery possibility will be firstly considered of removal. When delivery possibility is equal, the data copy with smaller TTL will be dropped.

Rule 8: overhead control should be adopted if there is an upper bound for the budget of replicated copies. Assume there is a threshold of the maximal number of replications in the network for a period of time of TTL, denoted by Θ (not including the original copy), and the average number of contacts between any vehicles is η . When a data chunk d_k meets all the rules from 1 to 7, we apply a probability p_{d_k} to replicate the data chunk, where $p_{d_k} = \frac{\Theta}{\sum_{i=1}^{\Theta} \eta^i}$.

In a long-term view, the total number of replicated data chunks will not exceed the upper bound.

5 Large data forwarding with capacity constraints in dense case

In a VANET, vehicles move following the road planning. Therefore, in many cases, vehicles are moving in platoons [33] or in a dense manner, such as along a highway or main roads in a city. Sometimes, the ad hoc networking of vehicles are very stable. Due to the nature of wireless communication, data transmission is more like a broadcast rather than a node-to-node communication in the sparse case. It is not possible for a relay vehicle talking to its neighboring vehicles in turn. Therefore, to achieve better delivery performance, the data chunks to be broadcasted must be carefully arranged. We assume that there are limited vehicles doing broadcasting simultaneously, as the data chunk replicating is limited in our unicast scenario. If two or more broadcasting vehicles have interference with others, we adopt a first-come-first-serve-based scheme.

That is, before a vehicle broadcasts the control signal, if it received one from another vehicle or an ongoing broadcasting process, it will wait until the other one finishes broadcasting or moves out of the range of interference. Small overlap is allowed since it will have little impact on the interest of majority of other vehicles.

Hence, in this section, we focus on improving data forwarding efficiency based on a single vehicle broadcast model, which is, forwarding data chunks in the relay to its neighboring vehicles such that the total delivery utility DU is maximized. As shown in Fig. 3, in a road segment (highway), there are many vehicles moving in two opposite directions. A vehicle R is trying to forward its data chunks to the neighboring vehicle. In its communication range, seven vehicles are involved, in which five of them are moving in the same direction as R , two are moving in the opposite direction. Due to the dynamic nature of the moving vehicles, the duration of each vehicle staying in the communication range of R varies. Therefore, one of the constraints is the limited forwarding communication capacity just the same as in the sparse case. The second constraint is that each neighboring vehicle has different available capacity. Under these two conditions, we will make arrangement of data chunks in the broadcasting. For simplicity, we will adjust and define parameters so that the problem is solvable. The solution is composed of two steps.

Step 1: FCC selection. As can be seen in Fig. 3, the duration of each neighboring vehicles of R from $N1$ to $N7$ staying in the communication range of R is quite different. Some may be too short for data forwarding, such as $N5$ (leaving due to opposite direction) and $N7$ (leaving due to faster speed). However, most vehicles will stay for a relative similar duration. Thus, we select the shortest duration among neighboring vehicles which have similar FCCs and name it c_{FTW} . Note that, it is possible to estimate the contact duration [30].

Step 2: Data chunk assignment. Note that in the previous section, as stated in **rule 1**, for any data chunk d_k ,

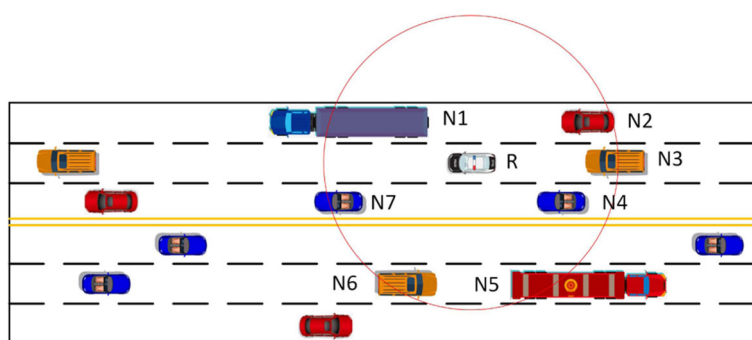


Fig. 3 Utilizing broadcast as a forwarding strategy in dense situation

although there may exist more than one neighbors in $N1$ to $N7$ which have better delivery possibility than R , only one neighbor will be selected for d_k to be forwarded. Therefore, there will be no replication of data chunks in the broadcast scenario. Each data chunk will be assigned no more than one vehicle, and multiple data chunks can be assigned to one vehicle.

5.1 System model

Here, we summarize the system model. In the network, there is a relay vehicle R which would find next relays for its data chunks. In its wireless communication range, there are a few candidate vehicles that have similar contact duration with R denoted as $(N1, N2, \dots, Nm)$. The length of selected contact duration is marked as c_{FTW} which can be estimated using the same method in [30]. The data chunks that the vehicle R holds are represented as (d_1, d_2, \dots, d_n) . Each candidate vehicle has an available capacity denoted as (c_1, c_2, \dots, c_m) . If a candidate vehicle Ni has better deliver possibility for data chunk d_k than R , at the same time Ni has enough space to hold d_k , we say it is a possible assignment. For this assignment, we could achieve the delivery utility DU as $P_{rio}(d_k) \times q_{Ni}^{dest}(d_k)$. The goal is to seek assignments for all data chunks that the sum of DU is maximized while the constraints of FCC and individual storage capacity are satisfied.

We use the scenario of Fig. 3 as an example. After FCC selection, we get a figure as shown in Fig. 4a. In vehicle R , there are four data chunks which have different data sizes as shown to the left of the data chunk icon. Five out of seven candidate vehicles are selected, which also have different available capacity as shown to the right of the vehicle icon. If a data chunk can be accommodated by a neighboring vehicle that has better delivery possibility, we connect them using a dashed line. The delivery utility is marked beside the line. Finally, we have the figure with all possible assignments. Now, we can turn it

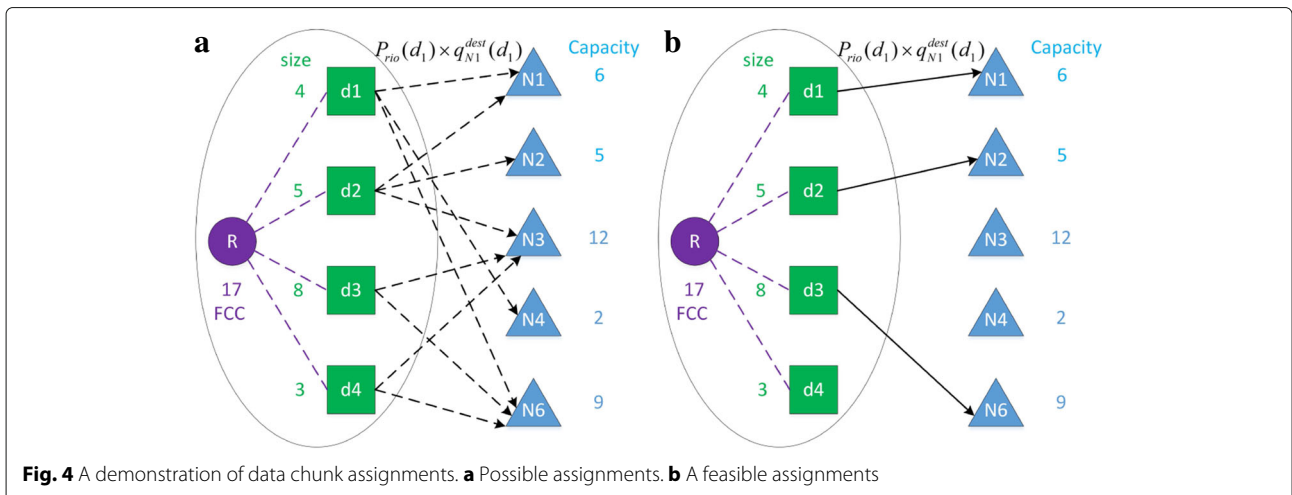
into a bipartite matching problem, with communication capacity and storage capacity.

In the broadcast scenario, we use the bipartite graph $G = (X, Y, E)$ to model the assignment optimization problem of n data chunks and m candidate vehicles, where $X = \{d_1, d_2, d_3, \dots, d_n\}$ denotes data chunk set, $Y = \{N1, N2, N3, \dots, Nm\}$ denotes candidate vehicles set, and $E = \{w(x, y) | x \in X, y \in Y, \}$ denotes set of delivery utility $w(x, y)$ when data chunk x is assigned with candidate vehicle y . $w(x, y) = P_{rio}(x) \times q_y^{dest}(x)$. Each candidate vehicle has limited storage capacity so that data chunks assigned to it should not exceed the capacity. A feasible assignment (not optimal) could be seen in Fig. 4b. If each data chunk has only one possible assignment, it is simply a knapsack problem. However, each assignment will possibly affect other assignments. We use $m(x, y)$ to indicate if there is an assignment between x and y . "0" means no assignment. "1" means assignment is chosen. "-" indicates an initial or inexecutable status.

The objective of the solution is to find a data chunk allocation that maximizes the delivery utility of all data chunks. Therefore, the problem is a weighted bipartite perfect matching which can be formalized in the following:

$$\begin{aligned} & \max \sum_{x \in X} \sum_{y \in Y} w(x, y) m(x, y) \\ & \text{s.t. every } m(x, y) = 0, 1, \text{ or } "-" \quad (i) \\ & \sum_{y \in Y} m(x, y) = 1 \text{ for every } x \in X \quad (ii) \\ & \sum_{x \in X} s_x m(x, y) \leq c_y \text{ for every } y \in Y \quad (iii) \\ & \sum_{x \in X, y \in Y} s_x m(x, y) \leq c_{FTW} \quad (iv) \end{aligned}$$

Constraint (i) ensures the data chunk assignment as a bipartite matching. Constraint (ii) guarantees each data chunk is only assigned to one vehicle. Constraint (iii) asserts data chunks assigned to a vehicle cannot exceed its available capacity. Constraint (iv) ensures that the total



data chunks transmitted will not exceed communication capacity.

5.2 Method to solve assignment problem with two constraints

5.2.1 Cost matrix build-up

In this paper, we extend the Hungarian algorithm-based K-M algorithm [34] to solve the problem. The first step of using bipartite matching is the build-up of cost matrix. In our scenario, the main weight is the delivery utility DU. A sample cost matrix is shown in Eq. 1 as derived from the scenario from Fig. 3.

$$\text{CostMatrix} = \begin{matrix} & N1 & N2 & N3 & N4 & N6 \\ \begin{matrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & 4 & 3 \\ 5 & 2 & 3 & 0 & 0 \\ 0 & 0 & 6 & 0 & 5 \\ 0 & 0 & 5 & 0 & 3 \end{pmatrix} \end{matrix} \quad (1)$$

5.2.2 Assignment algorithm

First, we have following definitions and equations [35].

Definition 6 Any $x \in X$ that has not seized the reservation is called *unsaturated*, and it has $m(x, y) \neq 1$ for every $y \in Y$. Any $y \in Y$ still available for allocation is called *available*, and it has $\sum_{x \in X} s_x m(x, y) < c_y$.

$$\alpha = \min_{x \in S, y \in Y \setminus T} \{L(x) + L(y) - R(x, y)\} \quad (2)$$

$$L(v) = \begin{cases} L'(v) - \alpha, & \text{if } v \in S \\ L'(v) + \alpha, & \text{if } v \in Y \text{ has been considered} \\ & \text{before for } S, \text{ i.e., } \{y \in Y \mid \\ & \exists m(x, y) = 1 \text{ where } x \in X\} \\ L'(v), & \text{otherwise} \end{cases} \quad (3)$$

$$m(x, y) = \begin{cases} 0 & L(x) + L(y) = R(x, y) \\ 1 & \text{if } x \text{ is located to } y \\ \text{" - " } & \text{otherwise} \end{cases} \quad (4)$$

The major difference from K-M algorithm is that in the table construction phase, the proposed algorithm will check the two capacity constrains accordingly.

6 Simulation

6.1 Simulation settings

Our simulation is carried out on the real vehicle trace umass/diesel [36]. The parameters are shown in Table 2. The source-destination pairs are randomly picked in the node set. Each sender will initiate a data file with the size around 3 to 10 GB. Also, each node will be assigned a storage space between 25 and 75 GB. Three sets of experi-

Table 2 Simulation parameters

Parameter	Value	Unit
Total nodes	23	Number
Number of contacts	1706	Number
Experiment duration	29	Day(s)
Range of data file size	3 10	GB
Range of storage space	25 75	GB

ment are conducted to observe the effect of three metrics, which is closely related to the large data file transmission. First is the number of concurrent forwardings, second is the storage capacity of vehicles, and the last is the size of data files.

Algorithm 3 Broadcast assignment based on Hungarian algorithm with communication and storage capacity

Require: X, Y, W, c_{FTW} , and c_y for each $y \in Y$

Ensure: a bipartite matching in Y for each $x \in X$

- 1: *Initialization* (i.e., $w(i, j)$, $m(i, j)$, and $L(v)$ with Eq.(3)).
- 2: *Completion check:*
For any unsaturated (Def. 6) data chunk $x \in X$, prepare S , T and E_x^* to start the matching process in the following; otherwise, successfully end the entire process.
- 3: *Label update for new opportunities for x to match with y in m :*
IF $N(S) \neq T$ *GOTO* phase 4 *ELSE*
Update reaching opportunities of data chunks by (first α with Eq.(2), and then L with Eq.(3), in order to reset m with Eq.(4).
- 4: *Table construction (E_x^*, m, T and S):*
Find any $y \in N(S) \setminus T$; *IF* y is available (Def. 6) and total data size does not exceed c_{FTW} , an alternating path from x (root of the tree E_x^*) to y exists, apply augment matching along this path as the Hungarian algorithm, decrease capacity of y by s_x , *GOTO* phase 2. *ELSE* find all z that are matched with y , update E_x^* with (z, y) respectively, $S = S \cup \{z\}$, $T = T \cup \{y\}$, *GOTO* phase 3.

Three algorithms were applied on this data trace. *MultiKnap-Single* represents the single-copy version of multiple knapsack-based forwarding algorithm. *MultiKnap-MC* is the proposed multi-copy strategy. The third algorithm (called *GreedyKnap-MC*) is a combination of [4] and [23]. When two vehicles meet, each of them will evaluate the other's forwarding metric and make duplication if possible. To achieve the maximal

incremental benefit, they model it as a 0-1 knapsack problem and solve it using a greedy algorithm. Each simulation is repeated for 100 times and record the average value.

6.2 Simulation result

6.2.1 Number of concurrent forwardings

We set the criterion of source-destination pairs as 15; therefore, most of nodes are included in the process of forwarding data files. Then, we slightly decrease and increase the number of source-destination pairs to see how it will affect the performance.

As shown in Fig. 5a, the proposed *MultiKnap-MC* method achieves the least delay where there is a 20% improvement compared with the runner-up *GreedyKnap-MC* method. The *GreedyKnap-Single* method gets the worst delay, about 25% more than that of the *MultiKnap-MC* method. All three algorithms will have worse performance when the number of source-destination pairs increases. *GreedyKnap-MC* is more sensitive to the increase since it does not have a data dropping strategy nor does it consider the communication capacity.

As shown in Fig. 5b, we evaluate the delivery ratio by how many bits have been received by the destinations. The total amount of bits comes from the files that have been successfully received. The reason for not using the number of data files is because our methods are designed to maintain the balance of both large and small files. With the increasing of the number of source-destination pairs, the delivery ratio of all three methods drops. However, our proposed method can achieve better ratio than the greedy methods since the solution to multiple knapsacks can optimize the total value (size) of the data files.

6.2.2 Storage capacity of nodes

We then fix the number of source-destination pairs and adjust the storage capacity of nodes. As shown in Fig. 6a,

when the storage space is very limited, all three methods suffer from a very large delay. As the storage space increases from 50 to 100%, our proposed *MultiKnap-MC* method reduces the delay quickly. Although the other three methods also reduce the delay, but the rate is slower than the *MultiKnap-MC* method. When the storage space is large enough such that a single node can accommodate the forwarding requests from multiple senders, each method gets approaching to others. Although the *MultiKnap-MC* method still achieves the best, the result of using multiple knapsack algorithm is not apparent any more.

It is with no doubt that the delivery ratio goes up when the capacity increases. In the end, they will be very close as can be seen in Fig. 6b.

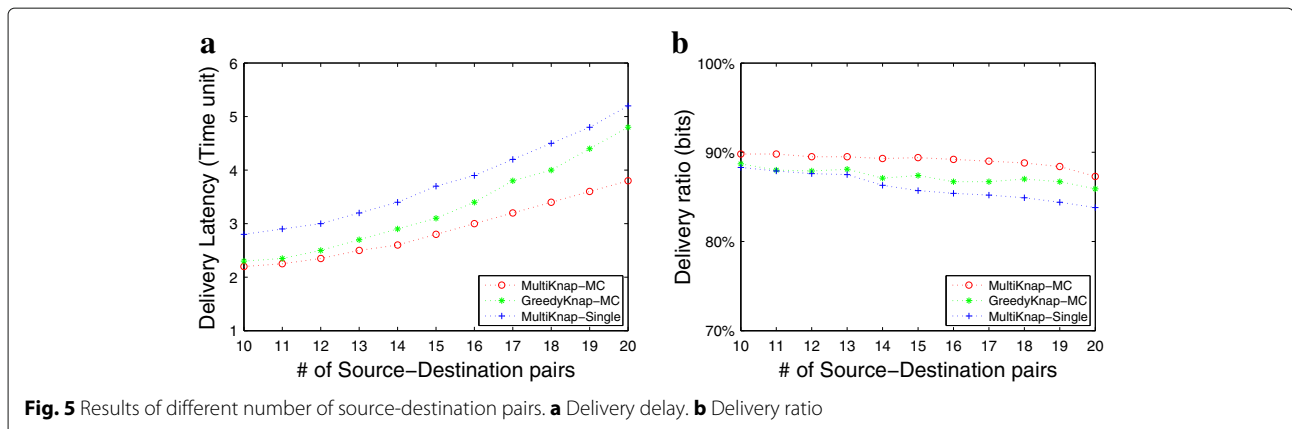
6.2.3 Size of data files

Figure 7a demonstrates the result when the size of data file increases from 50 to 150%. The larger the size of the data file, the less the number of data files a single vehicle can hold. Therefore, without the optimal arrangement of the forwarding selection, the *GreedyKnap-MC* and *MultiKnap-Single* methods will soon decrease to a bad performance. The multiple knapsack-based method can slow down this process so that it is 15% better than the greedy knapsack method when the ratio of data file is 100%. However, when the size of data files is too large to make any schedulings, all methods turn to get a worse delay. Figure 7b is similar to that of Fig. 5b; the *MultiKnap-MC* method achieves better delivery rate than the other methods.

6.2.4 Number of forwardings

We then evaluate how the data dropping rule can balance the cost and the performance so that the total number of forwardings of each strategy is considered.

As shown in Fig. 8, the number of forwardings of multi-copy scheme exceeds that of the single-copy scheme.



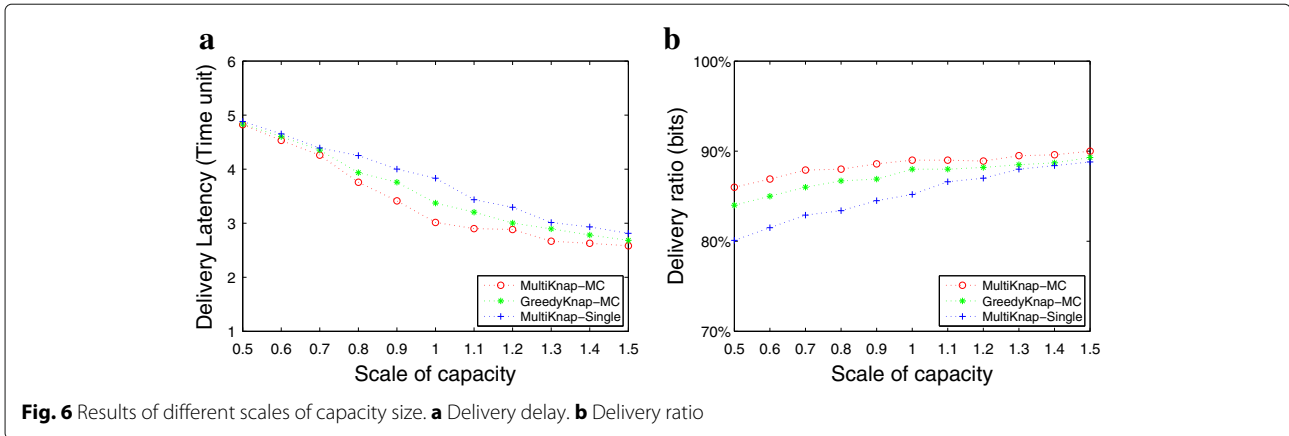


Fig. 6 Results of different scales of capacity size. **a** Delivery delay. **b** Delivery ratio

However, it achieves much better performance than the single-copy scheme. Compared with *GreedyKnap-MC*, it fairly controls the cost brought by data replication in the multi-copy algorithm.

6.3 Summary of simulation

After investigating into the real trace data, we draw a conclusion that the proposed *MultiKnap-MC* method can achieve not only the best delay performance under all simulation settings, but also a better delivery ratio. Since the storage occupancy rate of the network is almost 50%, the number of forwarding of *MultiKnap-MC* will not exceed the bound of two times of the single copy. If the storage occupancy is low throughout the network, *MultiKnap-MC* can have more forwardings than the single-copy scheme. When the number of replicated data chunks is doubled in the multi-copy algorithm, it is not necessary, meaning that the performance can be doubled too. Besides that the replicated data may be dropped during dissemination, the results also depend on the profile of the data trace, e.g., the average contact strength between any two nodes. Hence, we must consider the trade-off between the overhead and the performance. In most cases, there will be an upper bound

of the overhead. We then use rule 8 to control the overhead.

7 Conclusions

To solve the contradiction between the limited storage capacity and the large data size in the bulk-data dissemination, we propose the multiple knapsack-based solution which can achieve the local optimal in a distributed manner. In the sparse case, it first solves full arrangement of large data chunks when two vehicles meet as a multiple knapsack problem. Then, the communication capacity is considered. The proposed multi-copy algorithm can make full use the available storage resources in the network. In the dense case, data forwarding is implemented as a broadcast-like style. An optimization for the data chunk selection and assignment is proposed. Simulation is conducted using real trace data. The results show that our schemes achieve better performance both in delivery delay and ratio than the greedy knapsack-based competitor. In the future, to further improve the data dissemination efficiency, we will consider possible coding of data chunks during broadcasting, so that multiple data chunks can be transmitted to surrounding vehicles in the same time slot.

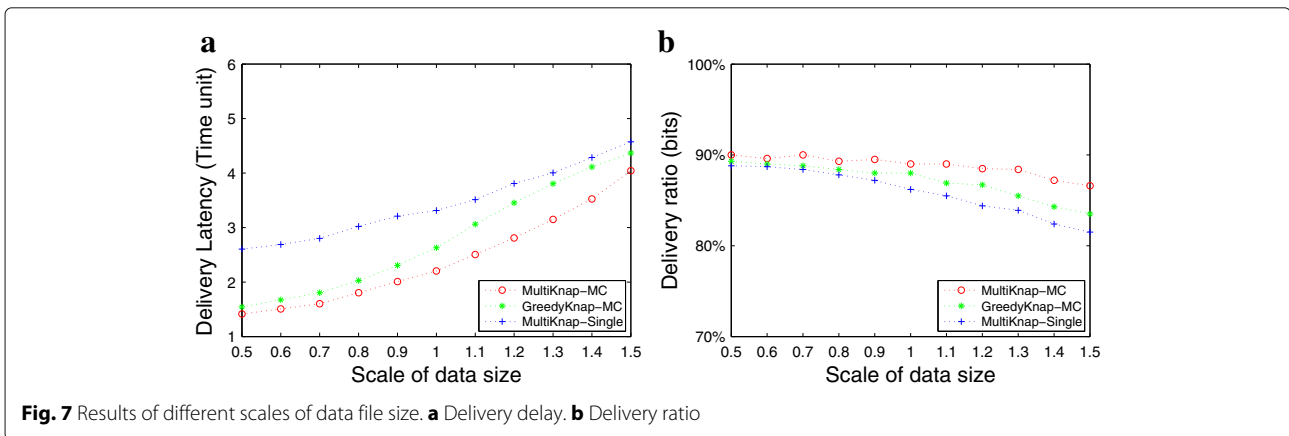
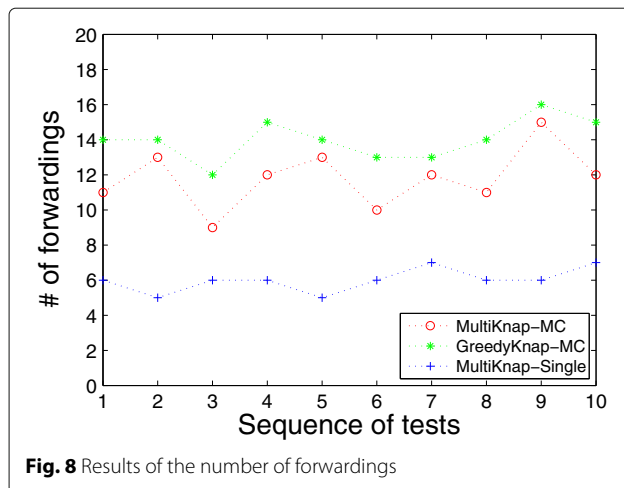


Fig. 7 Results of different scales of data file size. **a** Delivery delay. **b** Delivery ratio



Acknowledgements

This work is supported by the Natural Science Foundation of China (61601157 and 61672198), the cross-discipline innovation team building project of Hangzhou Dianzi University (Intelligent decision optimization and system operating security), the Scientific Research Foundation for the Returned Overseas Chinese Scholars (State Education Ministry, China), and the Chinese Scholarship Council (201208330096).

Authors' contributions

PL and JL proposed the main challenges and ideas. PL and TF completed the writing and formatting of the paper. YD did the experiments and simulations. XS helped in finalizing the solution and amending the manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

¹The Key Laboratory of Complex Systems Modeling and Simulation, School of Computer Science and Technology, Hangzhou Dianzi University, 1158 2nd Ave., Hangzhou, 310018 China. ²Department of Computer Science and Technology, University of Science and Technology Beijing, 30 Xueyuan Rd., Beijing, 100083 China.

Received: 2 December 2017 Accepted: 1 May 2018

Published online: 24 May 2018

References

- Wikipedia, TransAsia Airways Flight 235. https://en.wikipedia.org/wiki/TransAsia_Airways_Flight_235
- N Wang, J Wu, in *INFOCOM*. Opportunistic WiFi offloading in a vehicular environment: waiting or downloading now (IEEE, San Francisco, 2016), pp. 1–9
- Z Lu, X Sun, TF La Porta, Cooperative data offload in opportunistic networks: from mobile devices to infrastructure. *IEEE/ACM Trans. Netw.* **25**(6), 3382–3395 (2017)
- V Erramilli, M Crovella, A Chaintreau, C Diot, in *MobiHoc*. Delegation forwarding (ACM, Hongkong, 2008), pp. 251–260
- Q Ayub, S Rashid, MSM Zahid, AH Abdullah, Contact quality based forwarding strategy for delay tolerant network. *J. Netw. Comput. Appl.* **39**, 302–309 (2014)
- W Jiang, J Wu, F Li, G Wang, H Zheng, Trust evaluation in online social networks using generalized network flow. *IEEE Trans. Comput.* **65**(3), 952–963 (2016)
- W Jiang, J Wu, G Wang, On selecting recommenders for trust evaluation in online social networks. *ACM Trans. Internet Technol.* **15**(4), 14–11421 (2015)
- D Zeng, S Guo, A Barnawi, S Yu, I Stojmenovic, An improved stochastic modeling of opportunistic routing in vehicular CPS. *IEEE Trans. Comput.* **64**(7), 1819–1829 (2015)
- W Rao, K Zhao, Y Zhang, P Hui, S Tarkoma, Towards maximizing timely content delivery in delay tolerant networks. *IEEE Trans. Mob. Comput.* **14**(4), 755–769 (2015)
- J Wu, Y Wang, Hypercube-based multipath social feature routing in human contact networks. *IEEE Trans. Comput.* **63**(2), 383–396 (2014)
- H Zheng, J Wu, in *IWQoS*. Up-and-down routing in mobile opportunistic social networks with bloom-filter-based hints (IEEE, Hongkong, 2014), pp. 1–10
- M Xiao, J Wu, C Liu, L Huang, in *INFOCOM*. TOUR: time-sensitive opportunistic utility-based routing in delay tolerant networks (IEEE, Turin, 2013), pp. 2085–2091
- G Iosifidis, I Koutsopoulos, G Smaragdakis, in *INFOCOM*. The impact of storage capacity on end-to-end delay in time varying networks (IEEE, Shanghai, 2011), pp. 1494–1502
- B Xu, O Wolfson, J Lin, in *The Eighth International Conference on Advances in Mobile Computing and Multimedia*. Multimedia data in hybrid vehicular networks (ACM, Paris, 2010), pp. 109–116
- O Trullols-Cruces, B Xu, A new paradigm for querying blobs in vehicular networks. *IEEE MultiMedia*. **21**(1), 48–58 (2014)
- F Malandrino, C Casetti, C-F Chiasserini, M Fiore, Optimal content downloading in vehicular networks. *IEEE Trans. Mob. Comput.* **12**(7), 1377–1391 (2013)
- H Zhu, S Chang, ML 0001, K Naik, SX Shen, in *INFOCOM*. Exploiting temporal dependency for opportunistic forwarding in urban vehicular networks (IEEE, Shanghai, 2011), pp. 2192–2200
- B Wu, H Shen, K Chen, in *ICCCN*. Exploiting active sub-areas for multi-copy routing in VDTNS (IEEE, Las Vegas, 2015), pp. 1–8
- L Feng, Y Zhang, H Li, in *CSNT*. Large file transmission using self-adaptive data fragmentation in opportunistic networks (IEEE, Gwalior, 2015), pp. 1051–1055
- Ó Trullols-Cruces, M Fiore, JM Barceló-Ordinas, Cooperative download in vehicular environments. *IEEE Trans. Mob. Comput.* **11**(4), 663–678 (2012)
- S Siby, A Galati, T Bourchas, M Olivares, TR Gross, S Mangold, *METHoD: a framework for the emulation of a delay tolerant network scenario for media-content distribution in under-served regions*. (IEEE, Las Vegas, 2015), pp. 1–9
- M Abdelmoumen, M Frikha, T Chahed, in *ISNCC*. Performance of delay tolerant mobile networks and its improvement using mobile relay nodes under buffer constraint (IEEE, Yasmine Hammamet, 2015), pp. 1–6
- G Zhao, M Chen, Q Zuo, Data dissemination based on system utility in cooperative delay tolerant networks. *J. Comput. Res. Dev.* **50**, 1217–1226 (2013)
- D Zeng, S Guo, J Hu, Reliable bulk-data dissemination in delay tolerant networks. *IEEE Trans. Parallel Distrib. Syst.* **25**(8), 2180–2189 (2014)
- M Hu, Z Zhong, M Ni, A Bajocchi, Design and analysis of a beacon-less routing protocol for large volume content dissemination in vehicular ad hoc networks. *Sensors*. **16**(11), 1–27 (2016)
- C Chekuri, S Khanna, in *the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*. A PTAS for the multiple knapsack problem (ACM Press, San Francisco, 2000), pp. 213–222
- G Gao, M Xiao, J Wu, K Han, L Huang, in *SECON*. Deadline-sensitive mobile data offloading via opportunistic communications (IEEE, London, 2016), pp. 1–9
- Y Liu, AMAE Bashar, F Li, Y Wang, K Liu, in *WoWMoM*. Multi-copy data dissemination with probabilistic delay constraint in mobile opportunistic device-to-device networks (IEEE Computer Society, Coimbra, 2016), pp. 1–9
- DB Shmoys, É Tardos, An approximation algorithm for the generalized assignment problem. *Math. Program.* **62**, 461–474 (1993)
- J Hu, L-L Yang, HV Poor, L Hanzo, Bridging the social and wireless networking divide: Information dissemination in integrated cellular and opportunistic networks. *IEEE Access*. **3**, 1809–1848 (2015)
- C Liu, J Wu, On multicopy opportunistic forwarding protocols in nondeterministic delay tolerant networks. *IEEE Trans. Parallel Distrib. Syst.* **23**(6), 1121–1128 (2012)

32. S Zhang, D Huang, Z Chen, G Wu, Optimal stopping decision method for routing of opportunistic networks. *J. Softw.* **25**(6), 1291–1300 (2014)
33. H Hu, R Lu, Z Zhang, J Shao, REPLACE: a reliable trust-based platoon service recommendation scheme in VANET. *IEEE Trans. Veh. Technol.* **66**(2), 1786–1797 (2017)
34. xray, Assignment Problem and Hungarian Algorithm. <https://www.topcoder.com/community/data-science/data-science-tutorials/assignment-problem-and-hungarian-algorithm/>
35. P Liu, B Xu, Z Jiang, J Wu, in *ICCCN*. HAEP: hospital assignment for emergency patients in a big city (IEEE, Las Vegas, 2015), pp. 1–8
36. J Burgess, *et al.*, CRAWDAD Trace Umass/diesel/20080914. <https://crawdad.org/umass/diesel/20080914/>

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
