TRANSACTIONS ON EMERGING TELECOMMUNICATIONS TECHNOLOGIES
Trans. Emerging Tel. Tech. 2014; **25**:881–894
Published online 11 July 2013 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/ett.2685

RESEARCH ARTICLE

Knowledge-based replica deletion scheme using directional anti-packets for vehicular delay-tolerant networks

Xiaolan Tang[1], Juhua Pu[1,2]*, Yang Gao[1], Mohammad Aziz Z. Alshehri[3], Zhang Xiong[2] and Yueliang Wan[4]

[1] The State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China
[2] Research Institute of Beihang University in Shenzhen, Shenzhen 518057, China
[3] Royal Embassy of Saudi Arabia in China, Beijing 100027, China
[4] The Third Research Institute of Ministry of Public Security Run Technologies Co., Ltd. Beijing, Beijing 100044, China

ABSTRACT

As stable end-to-end paths seldom exist in vehicular delay-tolerant networks, replication-based opportunistic routing protocols are often utilised to improve the data delivery ratio and decrease the transfer delay. However, the network load increases heavily because of numerous data replicas. In this paper, we propose a knowledge-based replica deletion Scheme using Directional Anti-packets (SuDAs), in order to reduce the transmission of anti-packets by considering vehicle contact statistics. We firstly use the contact judgment algorithm to avoid invalid anti-packet transmissions and select better forwarder for the anti-packets to reach each relay node, which has redundant data replica. Then, in the directional anti-packet transmission algorithm, we set different anti-packet transmission thresholds according to different requirements for the quality of replica deletion in the network. Formal validations and extensive simulations evaluate the performance of our scheme, which show that SuDA has a distinct advantage over others in striking a balance between the replica deletion delay and the anti-packet overhead. Copyright © 2013 John Wiley & Sons, Ltd.

***Correspondence**
Juhua Pu, The State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China.
E-mail: pujh@buaa.edu.cn

Received 22 January 2013; Revised 31 May 2013; Accepted 17 June 2013

1. INTRODUCTION

Vehicular network is a kind of network with vehicular nodes equipped on fast-moving vehicles. It provides ubiquitous connectivity among mobile users and supports efficient vehicle-to-vehicle (V2V) communications. It is widely used in intelligent transportation systems to support various applications, such as safe driving, real-time traffic monitoring, highway toll payment, multimedia resource sharing and urban mobile surveillance [1, 2]. Vehicular delay-tolerant network (VDTN) is a disruptive network architecture based on delay-tolerant network (DTN) [3] and works for vehicular networks by gathering contributions from opportunistic and cooperative networks. In VDTNs, vehicles opportunistically carry data between terminal nodes, establishing network connectivity with unstable links under unreliable conditions [4]. Thus, unlike traditional wireless networks [5], VDTNs must cope with issues, such as highly dynamic network topology, short contact durations, connection disruption, variable vehicle density and frequent network fragmentation [6].

In VDTNs, because of high mobility and small communication radius, inter-vehicle connection cannot last long, and end-to-end paths seldom exist. Therefore, many researches use the opportunistic communication mechanisms to support V2V communication [7]. Hence, data messages are delivered in a store-carry-forward way [8]. In order to raise data delivery rate and keep short delivery delay, replication-based opportunistic routing protocols are widely used, such as Epidemic [9], Spray and wait (Spray&wait) [10], Prophet [11], MaxProp [12] and GeoSpray [13]. Although they show good performance on data transmission, the continuous existence of data replicas, after the data has been successfully delivered, wastes network resources heavily, such as node storage capacity and network bandwidth [14]. Especially, large data transmission (such as multimedia data) and the long lifetime of data message in vehicular opportunistic environment

further increase the waste of redundant data replicas on the limited storage capacity and the valuable contact time [15]. Thus, redundant data replicas worsen the performance of the network.

To present, there are already several but not many approaches to removing redundant replicas from vehicle storage spaces. The simplest is to use the lifetime of data message (*TTL*) to drop its copies when they are invalid. Although it is generally applied in routing protocols, it lacks flexibility and efficiency because the replicas just wait for their death time passively. Besides, several other schemes use anti-packets to actively delete data replicas. An anti-packet is an acknowledgement to show the delivery of a data message [16]. However, they often transmit the anti-packets by flooding, which also consume network resources if the flooding lasts until all the replicas are eliminated. Therefore, how to efficiently transmit anti-packets in vehicular networks is still a hard problem.

In this paper, we propose a novel and efficient replica deletion scheme named Scheme using Directional Anti-packet (SuDA), which uses directional anti-packets based on the prior knowledge of vehicle encounter statistics. There are two key algorithms in this scheme: the contact judgment algorithm and the directional anti-packet transmission algorithm. Firstly, we use the contact judgment algorithm to select better forwarder between the two meeting vehicles to forward the anti-packets to each relay vehicle, which has redundant data replica. Then, using the directional anti-packet transmission algorithm, we set the anti-packet transmission thresholds according to different requirements for the quality of replica deletion in the network. Finally, whether the encountering node forwards the anti-packet or not is decided by the anti-packet transmission rule.

Our contributions lie in three aspects. Firstly, our scheme SuDA is a novel attempt to implement directional anti-packet transmission instead of flooding anti-packets. Secondly, SuDA avoids invalid anti-packet transmission by using time comparison in the contact judgment algorithm. Thirdly, our scheme is flexible and applicable in various vehicular networks because it adjusts the anti-packet transmission threshold to strike a balance between the replica deletion latency and the deletion overhead according to network requirements.

The remainder of this paper is organised as follows. After surveying the related work in Section 2, we present a detailed description of our proposal in Section 3, followed by formal validations in Section 4 and simulation results in Section 5. Finally, we conclude this paper in Section 6.

2. RELATED WORK

Nowadays, there exist several but not many replica deletion schemes in opportunistic VDTNs, which use *TTL* attribute, anti-packet or other indicators to remove data replicas from the network.

TTL-based scheme is basic and simple. It deletes the data replicas of one data message when their *TTL* drops to be invalid. This deletion is executed in parallel at each node having the replica at the same time. This scheme is always integrated in routing protocols.

In anti-packet schemes, when all the destinations of one data message receive this message, they release anti-packets. Any node receiving the anti-packet drops its replica of this message. For instance, MaxProp routing protocol uses anti-packets sent from the destination and propagated to all peers in the network, in order to remove data replicas [12].

Other schemes use other indicators to delete data replicas in the network. For example, in [17], Bian *et al.* propose a scheme with contact counter threshold to decrease the number of data replicas. When the contact counter between two nodes reaches the threshold, the shared message replicas are removed from one node. With proper threshold, it reduces the number of data replicas obviously while keeping good data delivery ratio. But a proper threshold is difficult to select. In [18], Yu *et al.* present a routing scheme based on a stochastic process for Epidemic routing. Message redundancy is efficiently reduced, and the number of message copies is controlled reasonably.

As different kinds of replica deletion schemes can coexist in the network and our proposal concerns the anti-packet scheme, we discuss this kind of scheme in detail here. According to the transmission time of anti-packets, replica deletion schemes using anti-packets are classified into two main groups, passive and active.

In passive schemes, the anti-packet of one message is only transmitted to an encountering node when this node tries to send a copy of this message. In this case, anti-packets are slowly distributed, and data replicas are slowly deleted accordingly, while the overhead of anti-packets is small. In active schemes, a node holding the anti-packet of a message tries to share this acknowledgement with its encountering node no matter whether this node has the message replica or not. An extreme case of active schemes is to flood the anti-packets in the network (broadcast-like). Broadcast-like anti-packet scheme has the quickest replica deletion, but the anti-packets consume the most storage space and transmission bandwidth. Other active schemes are named as multicast-type or publish–subscribe schemes, which aim at just transmitting anti-packets to those nodes in concern.

Most existing replica deletion schemes using anti-packets are broadcast-like, such as MaxProp [12]. Some further research tries to improve the efficiency of anti-packet delivery. In [19], Kaveevivitchai *et al.* design a broadcast-like anti-packet scheme, which uses the heuristic knowledge of actual human mobility and deploys static or dynamic helper nodes to relay anti-packets. Thus, it achieves better network resource utilisation than basic broadcast-like scheme. However, the helper nodes increase network expense and complexity. Besides, for the localisation of helper nodes is strongly affected by the mobility models in the network, this scheme is not generally

Trans. Emerging Tel. Tech. **25**:881–894 (2014) © 2013 John Wiley & Sons, Ltd.
DOI: 10.1002/ett

applicable in different scenarios. In one word, the existing replica deletion schemes using anti-packets can not balance the deletion speed with the anti-packet overhead well in different network models.

In order to accommodate the deletion efficiency and the deletion cost, multicast-type anti-packet transmission is required. However, to the best of our knowledge, little research has been performed on this issue because of the complexity of highly dynamic vehicular networks. Thus, in this paper, with the help of prior knowledge of vehicle encounter statistics, we propose a replica deletion scheme using multicast-type anti-packets, in order to improve replica deletion efficiency and keep low deletion overhead in vehicular networks.

3. SCHEME USING DIRECTIONAL ANTI-PACKET

3.1. Network model

In this paper, we just discuss unicast data transmission in VDTNs. That is to say, each message is created at one vehicle and to be transmitted to another vehicle. This is because the definitions of successfully delivered message are different in unicast, multicast and broadcast data transmissions. Specifically, in unicast transmission, when the data reach their only destination, we say they are delivered, and then, the anti-packet is generated. In multicast, only when the data reach all their destinations could the anti-packets be released [20]. How to judge whether a data message has reached all its destinations or not in a distributed network is an open issue. It is beyond the research scope in this paper. In broadcast data transmission, there is no redundant data replica in the network because every node except the source is the destination of the data message [21]. Thus, there is no need to use replica deletion in broadcast transmission. Above all, we only discuss unicast data transmission in this paper.

To clarify the condition of the network before executing our replica deletion scheme, we make some assumptions about the network model. Firstly, we assume all the vehicles have the same communication radius because different communication ranges lead to heterogeneous links in two directions and hence hinder the bidirectional communication among encountering vehicles [22]. When two vehicles move into each other's communication range and construct connection between them, we say they meet. Each vehicle is called the neighbour of the other. Secondly, although one vehicle may be in the communication ranges of several other vehicles, we only analyse one-to-one connection here and leave one-to-many connection for our further research. That is to say, in one contact, each vehicle communicates with only one neighbour. After completing this contact, it can select another vehicle to be its neighbour and establish a new contact. Thirdly, we assume that data messages and anti-packets are transmitted successfully without collision, and leave the discussions about this collision to our future work [23].

In this paper, to clearly present our replica deletion scheme, we introduce relevant attributes first. The notations of data message attributes, anti-packet attributes and node attributes are listed in Table I. More comments of these attributes are listed as follows:

(1) For each source node generates at most one data message at a specific time, we use the triad (*SID*, *DID* and *TTS*) to be the identifier of data message. To simplify the representation, we further use S_D_T to be short for this triad. S_D_T is used in anti-packets.
(2) Both *TTL* in data message and TTL_M in anti-packet decline as time passes. When their values become invalid, the data message or anti-packet is dropped by

Table I. Relevant attributes in messages and nodes.

Type	Name	Expression	Remark
Data message attributes	Source node ID	***SID***	The triad (***SID***, ***DID***, ***TTS***) stands for the identifier of data message, *S_D_T* for short.
	Destination node ID	***DID***	
	Time to send	***TTS***	
	Data	***DATA***	***DATA*** means the data in the message.
	Time to live	***TTL***	***TTL*** drops as time passes and helps to clean invalid data messages.
	List of relay node IDs	***RList***	***RList*** records those nodes that have replicas of this message.
Anti-packet attributes	Message identifier	$S\_D\_T_M$	$S\_D\_T_M$ is the identifier of the data message *M*, which is successfully delivered and whose replicas are to be deleted.
	Time to live of this anti-packet	TTL_M	TTL_M drops as time passes and helps to clean invalid anti-packets.
	List of relay node IDs	$RList_M$	$RList_M$ records those nodes that have replicas of message *M* before the replica deletion process.
	List of cleaned relay node IDs	$CRList_M$	$CRList_M$ records those nodes having deleted replicas of message *M*.
Node attributes	Vehicle contact probability	***VCP***	***VCP***(*n*) is the probability of this vehicle to meet node *n*.
	Vehicle intermeeting time	***VIT***	***VIT***(*n*) is the waiting time for this vehicle to meet node *n*.
	Anti-packet transmission threshold	***ATT***	***ATT*** is the threshold in the anti-packet transmission rule.

the vehicular node. Besides, the values of *TTL* and TTL_M regarding to the same data message are the same. The reasons lie in two aspects. For one thing, larger TTL_M than *TTL* is nonsense because *TTL* expiration results in immediate replica deletion in the whole network. For another, larger *TTL* than TTL_M is inefficient because early expiration of TTL_M results in no anti-packets to remove the remaining data replicas.

(3) As different replicas of the same data message are often forwarded along different paths, the relay nodes recorded in *RList* of these replicas are different. In order to obtain the complete information about the relay nodes that have data replicas, *RList* in different replicas of the same data message are merged together. For instance, one replica M_1 of the data message M has attribute $RList = \{R1, R2\}$, while another replica M_2 of message M has $RList = \{R1, R3\}$. When the node A carrying M_1 meets the node B carrying M_2, they both update the *RList* in M_1 and M_2 to be $RList = \{R1, R2\} \cup \{R1, R2\} = \{R1, R2, R3\}$.

As the anti-packets are used to delete data replicas in relay nodes, *RList* in one replica, which is to be deleted by the anti-packet, should be included into $RList_M$ of the anti-packet. Besides, when two vehicles both having anti-packets regarding to the same data message meet with each other, the $RList_M$ in these two anti-packets are also merged to complete the record of the relay nodes.

(4) Similarly, the records of already cleaned nodes $CRList_M$ in different anti-packets regarding to the same data message are also merged, in order to improve the completeness of this record in distributed networks.

(5) *VCP* and *VIT* are the prior knowledge about vehicle encounter statistics, which are computed from the historical contact information.

(6) *ATT* is the threshold in the anti-packet transmission rule. Its usage will be discussed in detail in Section 3.3.

Now, we describe the communication process between two meeting vehicles briefly. When two vehicles meet, they first exchange anti-packets to delete redundant data replicas in the neighbour's storage space. Then, vehicles transmit data messages according to their routing protocols. Finally, they take our algorithms to decide on whether to store the new anti-packets from the neighbour node or not. Here, we take the replica deletion as the first step, in order to avoid useless data transmission in the second step. Moreover, we put our anti-packet transmission algorithms as the final step because the previous steps update the information about relay nodes, which is used in our algorithms. As the first and second steps are easy to understand and they are not our focus, we only describe the details of the final step later.

When two arbitrary vehicles A and B meet with each other, we take vehicle B sending the anti-packet of data message M to vehicle A as an example. If vehicle A also has the anti-packet of message M, then A and B merge the $RList_M$ and $CRList_M$ in their anti-packets; otherwise, vehicle A executes the contact judgment algorithm and the directional anti-packet transmission algorithm to decide on whether to carry the anti-packet of message M or not. In the next subsections, we depict the processes of these two algorithms in detail.

3.2. Contact judgment algorithm

In the contact judgment algorithm, we use time comparison based on the vehicle intermeeting time to avoid invalid anti-packet transmission, and further use probability comparison to select better forwarder between the encountering vehicles to transmit the anti-packet to each relay node, which has data replica. The process of this algorithm is shown in Figure 1.

We name those relay nodes, which have replicas of message M and the replicas have not been deleted because of anti-packets, as valid relay nodes. It is apparent that the valid relay nodes are the current destinations of this anti-packet. We analyse these valid relay nodes one by one. Take an arbitrary valid relay node R as an example. We set a variable *FWR* to stand for the better node to forward this anti-packet to node R. As an alternative choice in the neighbourhood, *FWR* is either vehicle A or vehicle B. We initialise this variable to be *NULL*.

In the first step, we use time comparison to avoid invalid anti-packet transmission. Specifically, we compare the lifetime of this anti-packet with the intermeeting time for node A to meet node R. If the lifetime of the anti-packet is shorter than the intermeeting time, it means that the replica of message M at node R will be deleted because of TTL expiration rather than its anti-packet from node A. Thus, the transmission of this anti-packet from node A to node R is useless. In order to avoid the resource waste due to this kind of invalid anti-packet transmission, we select node B to be the forwarder to transmit this anti-packet to node R. Detailed proof of this statement is shown in Lemma 1, Section 4.

Contact Judgment Algorithm

for each relay node $R \in RList_M - CRList_M$, do
  Initialize the better node to forward the anti-packet of message M to node R as $FWR = NULL$
  if $TTL_M < A.VIT(R)$, then
    $FWR = B$
  else if $A.VCP(R) < B.VCP(R)$, then
    $FWR = B$
  else
    $FWR = A$
  end if
end for

Figure 1. Process of contact judgment algorithm.

Trans. Emerging Tel. Tech. **25**:881–894 (2014)
DOI: 10.1002/ett

In the second step, we use probability comparison to select forwarder to relay this anti-packet to its destinations. Specifically, we compare the contact probabilities of vehicles A and B to meet node R. Which has the larger probability is assigned to FWR. This is because higher probability means that this node has more chance to meet node R and delete its replica of message M. The motivation of this step is from the Prophet routing protocol [11], in which nodes only forward data to those encountering nodes that have larger probabilities to meet the destinations than themselves.

Using the contact judgment algorithm, we avoid useless anti-packet transmission and select the better relay node in the neighbourhood to forward this anti-packet to each of its destinations. This is the basis of the directional anti-packet transmission algorithm later.

3.3. Directional anti-packet transmission algorithm

Based on the selected forwarders for all the valid relay nodes in the contact judgment algorithm, we propose the directional anti-packet transmission algorithm to set transmission rule for anti-packets according to different requirements about the quality of replica deletion in the network. Figure 2 shows the process of this algorithm.

To support this algorithm, we first define a new attribute, the anti-packet transmission threshold *ATT*. The value of *ATT* is decided by the network requirements about the replica deletion quality. It is preset and recorded in all the vehicles. Specifically, if the network requires short replica deletion latency and allows large overhead of anti-packets accordingly, lower *ATT* is selected, and vice versa. We prove this statement in Lemma 2, Section 4.

Now, we describe the anti-packet transmission rule. We count the number of vehicle A selected as the forwarder to transmit the anti-packet to the valid relay nodes of message M. If the ratio of A's occurrence to the total number of valid relay nodes is larger than or equals ATT, then vehicle A stores the anti-packet of message M in its storage space and will transmit this anti-packet later to its encountering nodes and finally to the destinations of this anti-packet; otherwise, node A drops this anti-packet. The assignment of *ATT* affects the performance of anti-packet transmission rule and hence influences the quality of replica deletion. We will analyse this effect in our simulation in Section 5.3.4.

In this way, we only allow directional anti-packet transmission to good forwarders, which have more chance to deliver this anti-packet to its destinations. The control intensity of this algorithm is decided by the threshold *ATT*. By adjusting *ATT*, our scheme can provide the required replica deletion quality and achieve a balance between the deletion delay and the deletion overhead. Thus, our scheme is effective and flexible because of its adjustable multicast-type anti-packet transmission rule.

Directional Anti-packet Transmission Algorithm

Initialize the total counter as $Count = 0$
Initialize the counter about vehicle A as $Count_A = 0$
for each relay node $R \in RList_M - CRList_M$, do
    $Count = Count + 1$
    if $FWR == A$, then
        $Count_A = Count_A + 1$
    end if
end for
$Ratio = \frac{Count_A}{Count}$
if $Ratio \geq ATT$, then
    A stores the anti-packet of message M
else
    A deletes the anti-packet of message M
end if

Figure 2. Process of directional anti-packet transmission algorithm.

3.4. Performance analysis

As widely used in the Internet and some traditional wireless networks, anti-packets work well to remove data replicas from their carriers. However, because of the fast movement of vehicles and the distributed computing in VDTNs, the problem of data replica deletion using anti-packets is complex and difficult. Most previous schemes only flood the anti-packets in the network such that the overhead of anti-packets is very large.

In this paper, we use the directional transmission of anti-packets to improve the efficiency of data replica deletion. The main specialties of our proposal are listed later, which determines that our scheme outperforms previous schemes.

(1) Generally speaking, unlike those flooding schemes that broadcast anti-packets in the whole network, SuDA is a multicast-type anti-packet scheme, which aims at sending the anti-packets only to their destinations. Hence, the transmission overhead and the number of anti-packets can be greatly reduced.
(2) The lifetime checking in the contact judgment algorithm, that is, the comparison between TTL_M and $A.VIT(R)$, can avoid the case that when an anti-packet is transmitted to a carrier of the data replica, those replicas have been deleted because of their *TTL* expiration. Thus, the forwarding of anti-packets in our scheme avoids useless resource consumption.
(3) The contact probability comparison in the contact judgment algorithm, that is, the comparison between $A.VCP(R)$ and $B.VCP(R)$, aims at an optimised anti-packet distribution between two meeting vehicles. Specifically, for each relay node having the replicas of a delivered message, the one in the neighbourhood that has a higher probability to meet this carrier has the opportunity to forward this anti-packet. This results in the directionality of the anti-packet transmission in each pair of meeting vehicles, which

avoids the large overhead due to random transmission in anti-packet flooding schemes.

(4) The ratio checking in the directional anti-packet transmission algorithm, that is, the comparison between *Ratio* and *ATT*, arrives at a balance between the replica deletion overhead and the replica deletion speed. As an adjustable system parameter, the assignment of *ATT* determines the deletion performance. A suitable value leads to a quick replica deletion with an acceptable resource consumption. How to obtain this value will be discussed in Section 5.3.4.

The previous analysis shows that our scheme improves the replica deletion efficiency. The formal validations and the simulation results in the next sections will further evaluate the performance of our proposal. In addition, we understand that besides vehicular DTNs, the proposed scheme can be applied to other kinds of DTNs where the node mobility has some regularity and a large amount of data transfer occupies the limited resources.

4. FORMAL VALIDATION

In this section, we prove that our scheme SuDA improves the efficiency of data replica deletion and meets different quality requirements for replica deletion in VDTNs.

Lemma 1. *Using time comparison in the contact judgment algorithm, SuDA avoids invalid anti-packet transmissions.*

Proof. In Section 3.2, we have introduced the contact judgment algorithm. Here, we still take vehicle B sending the anti-packet of data message M to its encountering vehicle A as an example. Each valid relay node $R \in RList_M - CRList_M$ has replica of message M and has not deleted this replica by anti-packets. We attempt to select better forwarder node between the encountering vehicles A and B to transmit this anti-packet to each valid relay node.

In the time comparison step, we compare the lifetime of this anti-packet recorded as TTL_M with the intermeeting time for vehicle A to meet node R recorded as $A.VIT(R)$. As analysed in Section 3.1, the lifetime of the anti-packet regarding to one message equals the lifetime of this data message. That is $TTL_M = TTL$. Hence, we actually compare the lifetime of data replicas of message M with the transmission duration from A to R. If $TTL_M < A.VIT(R)$, we know $TTL < A.VIT(R)$. It means that when the anti-packet is successfully delivered from vehicle A to the relay node R, the TTL of the replica of message M in node R has expired, and the replica has been removed because of invalid TTL. Thus, the anti-packet transmission from A to R is nonsense. In order to avoid this kind of invalid transmission from vehicle A, we select vehicle B to be the forwarder node to transmit this anti-packet to node R in this case. Therefore, we see that the time comparison in the contact judgment algorithm helps SuDA to avoid invalid anti-packet transmissions. □

Lemma 2. *A lower ATT results in a shorter replica deletion latency and a larger deletion overhead, and vice versa.*

Proof. In Section 3.3, we have depicted the process of the directional anti-packet transmission algorithm. *ATT* is the threshold to make decision on whether the encountering vehicle A is qualified to forward the anti-packet or not. From the anti-packet transmission rule $Ratio \geqslant ATT$ in Figure 2, we see that lower *ATT* means that node A has more chance to store and transmit the anti-packet. In other words, lower *ATT* relaxes the limitations on anti-packet transmission. Thus, more anti-packets can be transmitted in the network. More anti-packets accelerate the process of data replica deletion, while the storage and bandwidth overhead of these anti-packets increase accordingly. Therefore, we conclude that a lower *ATT* results in a shorter replica deletion latency with a larger overhead.

Similarly, for a higher *ATT* puts more restrictions on anti-packet transmission, there are fewer anti-packets in the network. Hence, the replica deletion speed declines, while the extra overhead due to anti-packets also drops. We conclude that a higher *ATT* results in a smaller replica deletion consumption with a longer deletion latency. □

5. PERFORMANCE EVALUATION

5.1. Network configurations

We simulate SuDA on the opportunistic networking environment (ONE) simulator [24, 25]. For one thing, ONE has been recognised as a standard simulator for opportunistic communication mechanisms. For another, several typical replica deletion schemes have been implemented in ONE, as shown in [19]. Although ONE does not support elaborate realistic vehicular network models, such as car following, lane changing and so on, the store-carry-forward communication paradigm in ONE is helpful and adequate for us to testify the performance of our proposal. The simulation environment configurations are listed in Table II. More comments of these configurations are listed as follows:

(1) Based on 802.11p standard, we use actual communication radius 200 m and data rate 5 Mbps.

(2) To simulate a vehicular network with regular vehicle contact probability and intermeeting time, we use the bus movement model [26], which is integrated in ONE. Different mobility speeds are set for the vehicles to improve the variety of the scenario. Other mobility models, such as the random waypoint movement model (RWP) [27], are analysed in Section 5.3.1. Note that we use a grid network scenario here to simplify the bus path selection, while a real map scenario around Haidian district (Beijing, China) is discussed in Section 5.3.1.

(3) Although the size of anti-packet changes with the number of relay nodes, which are recorded in the

Trans. Emerging Tel. Tech. **25**:881–894 (2014) © 2013 John Wiley & Sons, Ltd.
DOI: 10.1002/ett

attributes *RList* and *CRList*, the threshold of this number is definite in Spray&wait routing protocol. Besides, for anti-packets just carry basic information about data messages, to simplify the analysis, we assume that all the anti-packets have the same size.

(4) The buffer management policy used in our simulation is first in first out, as a default policy in ONE. As the buffer size affects the storage states of data messages and, hence, the replica deletion efficiency, we will analyse this effect in Section 5.3.2.

(5) For a given data message, its source node is definite because each vehicle creates one message randomly in every 20 s before 2000 s; its destination node is randomly selected aiming at improving the robustness of the simulation.

(6) As the initial *TTL* indicates network requirements for data delivery latency, it affects the performances of replica deletion schemes. Details about this influence are discussed in Section 5.3.3.

To evaluate the performance of our scheme SuDA, we compare SuDA with passive anti-packet scheme (PA) and broadcast-like anti-packet scheme (BA), which are depicted in Section 2. Besides, in order to show the effects of anti-packets, we also make contrast experiments on the routing protocol without anti-packets, which is named the Basic scheme (Basic). All the schemes have *TTL* validation to delete out of date messages immediately.

In our scheme, the anti-packet transmission threshold *ATT* is adjustable. Here, we set three typical values for *ATT* to show the performance of SuDA in detail: (1) SuDA-1: $ATT = 1\%$; (2) SuDA-2: $ATT = 50\%$; and (3) SuDA-3: $ATT = 100\%$. Note that in the first group, we use 1% instead of 0. This is because considering the rule $Ratio \geqslant ATT$ in Figure 2, the assignment $ATT = 0$ results in broadcast-like anti-packet transmission. Besides, we will discuss the effect of *ATT* value in Section 5.3.4.

In the simulation, we analyse seven criteria: the data delivery ratio, the average delivery delay, the number of data replicas, the number of anti-packets, the transmission overhead of anti-packets, the number of delivered anti-packets and the average number of data replicas deleted by an anti-packet. The data delivery ratio shows the ratio of successfully delivered messages to totally created messages during the simulation time. A higher data delivery ratio means a better data transmission performance. The average delivery delay is the average delay of successfully delivered messages. A shorter delay means that the messages can be delivered earlier. Besides, the number of data replicas and the number of anti-packets generally show the deletion speed and the storage overhead of anti-packets in these schemes. The transmission overhead of anti-packets shows the communication resource consumption caused by anti-packets, while the number of delivered anti-packets shows the efficiency of the anti-packet transmission. The lower the transmission overhead is and the larger the number of delivered anti-packets is, the better the replica deletion scheme works. The average number of data replicas deleted by an anti-packet shows the storage efficiency of anti-packets. The larger this number is, the higher storage utilisation the replica deletion scheme has.

Table II. Simulation environment configurations.

Parameter	Value
Network area	Grid map 1000 × 1000 m^2, each grid 100 × 100 m^2
Number of vehicles	100
Simulation time	2400 s (steps)
Communication radius	200 m
Data rate	5 Mbps
Mobility model	Bus movement model
Mobility speed	75%: 10–20 m/s; 25%: 20–30 m/s
Routing protocol	Spray&wait
Number of replicas for each data message	10
Buffer size	1 GB
Buffer management policy	First in first out (FIFO)
Size of data message	10 MB
Size of anti-packet	10 KB
Data creation event	Each vehicle creates one message randomly in every 20 s before 2000 s
Destination node	Randomly selected
TTL	400 s

Note that all the values in the simulation results are average in 100 experiments to improve the accuracy and the 95% confidence intervals are discussed in Section 5.2 to validate the confidence level of our results. Besides the results in Section 5.2, which are obtained with the network configuration shown in Table II, we still simulate other configurations to see the influence of relevant parameters in Section 5.3. Although we cannot give a complete theoretical validation of the simulation results at the present time, the qualitative performance analysis in Section 3.4, which is in accord with the data analysis in Sections 5.2 and 5.3, validates the simulation results to some extent.

5.2. Simulation results

In this section, we present the detailed simulation results of the six replica deletion schemes, that is, Basic, PA, BA, SuDA-1, SuDA-2 and SuDA-3, respectively. The results are shown in Figure 3.

From Figure 3(a), we see that the data delivery ratios of Basic and PA are the lowest while other schemes have ratios at a similar level. The reasons lie in three aspects. (1) Because the Basic scheme has no anti-packets, the redundant replicas of those messages, which have been delivered successfully, cannot be removed from nodes in time. They occupy some storage spaces and communication bandwidth and thus hinder the transmission of undelivered messages. (2) Although PA has passive anti-packets, the anti-packets are transferred only when the node meets valid relay nodes. The transmission of anti-packets is severely restricted by the rare chance to meet those nodes. (3)

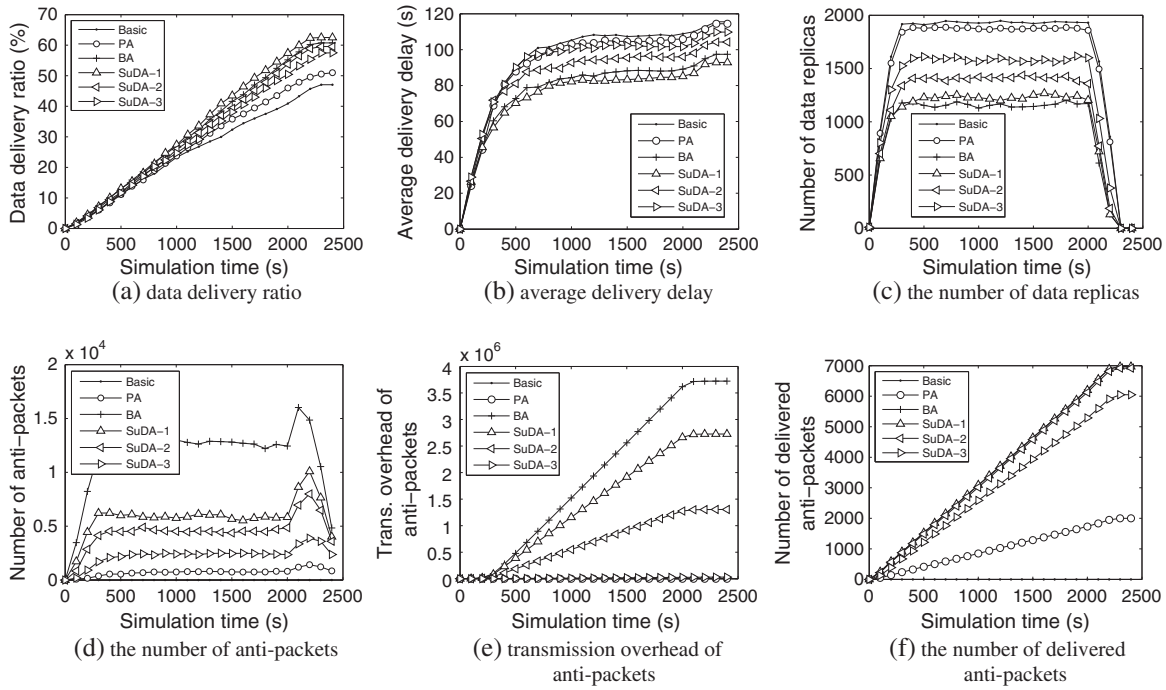


Figure 3. Simulation results. (a) Data delivery ratio, (b) average delivery delay, (c) the number of data replicas, (d) the number of anti-packets, (e) transmission overhead of anti-packets and (f) the number of delivered anti-packets.

As BA and SuDA transmit enough anti-packets to delete data replicas, they improve the delivery ratio to some extent. However, SuDA-1 and SuDA-2 perform better than SuDA-3 because with increasing *ATT*, fewer anti-packets can be transmitted in the network as Lemma 2 shows. In addition, we see that the delivery ratio of SuDA-1 is a little higher than BA. There are two main reasons: (1) The flooding of anti-packets in BA results in excessive resource consumption, and (2) SuDA supports efficient directional anti-packet transmission and hence saves network resources to deliver data messages.

As Figure 3(b) shows, the average delivery delays of Basic, PA and SuDA-3 are the largest. This is because these three schemes have no or few anti-packets to remove redundant data replicas. These useless replicas make some undelivered data messages be deleted because of buffer overflow or be transmitted late because of bandwidth competition. In the figure, BA and SuDA-1 performs much better than others for their sufficient anti-packets. Especially, SuDA-1 has the lowest delivery delay, which shows that our scheme works well in high delay restriction scenarios.

Figure 3(c and d) shows the numbers of data replicas and anti-packets in the network. In the beginning of the simulation, these two numbers increase greatly because of continuous data message creation and delivery. Then, they become stable for the balance between data generation and data deletion. Finally, they drop greatly after 2000 because there are no new data created. Note that at around 2000 s, the number of anti-packets has a quick increase before the sharp drop. This is because the existing data messages are still transferred after 2000 s, which increase the data delivery ratio and hence raise the number of anti-packets.

Comparing these schemes in Figure 3(c and d), we see that Basic and PA have the fewest anti-packets and the most data replicas, while BA has the fewest data replicas at the cost of much more anti-packets than others. Our three schemes strike a balance between the numbers of data replicas and anti-packets. Note that SuDA-1 deletes data replicas as efficiently as BA does, while its resource consumption due to anti-packets is about a half of that in BA. It is further verified by Figure 3(e and f) that SuDA delivers a similar number of anti-packets with a much less transmission overhead than BA.

In this simulation, the average number of data replicas deleted by an anti-packet in Basic, PA, BA, SuDA-1, SuDA-2 and SuDA-3 are 0, 2.74, 0.01, 0.35, 0.58 and 0.82, respectively. PA has a much higher number than others because its anti-packets are only forwarded to their destinations. Although PA has a high storage efficiency of anti-packets, its overall performance is not good because of its small number of anti-packets. Besides, we see that SuDA has more replicas deleted by one anti-packet than BA, which shows that SuDA has a better storage utilisation of anti-packets.

In one word, the overall performance of our proposal is better than that of PA and BA schemes. Using directional anti-packet transmission, SuDA deletes redundant data replicas efficiently with little extra overhead.

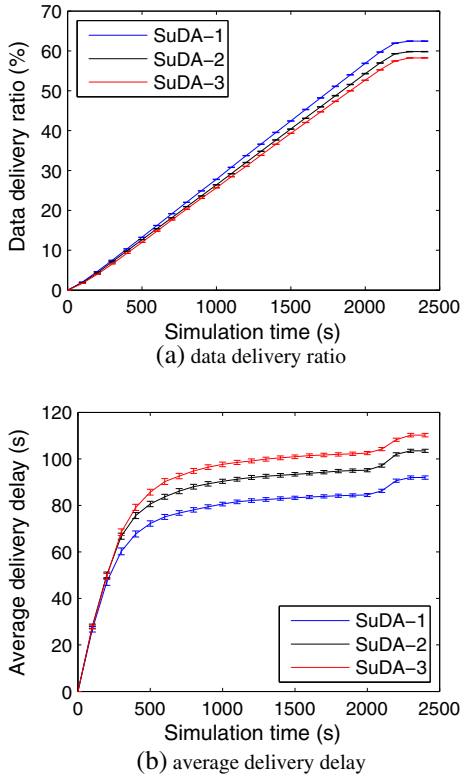


Figure 4. 95% confidence interval. (a) Data delivery ratio and (b) average delivery delay.

Compared with traditional schemes, SuDA increases the data delivery ratio and reduces the average delivery delay to some extent. Thus, we conclude that SuDA improves the data transmission efficiency in the whole network.

Besides, in order to testify the confidence level of the simulation results, we compute the 95% confidence interval of these results in the 100 independent experiments. Considering the limited space and the clarity of the figure, we just show the intervals of the data delivery ratios and the average delivery delays in SuDA-1, SuDA-2 and SuDA-3 schemes in Figure 4. From this figure, we see that the confidence intervals are very narrow, which indicates that the fluctuation of the simulation results in the 100 experiments is small. Besides, the same conclusion can be reached in other criteria and other schemes. Thus, these results are robust and reliable.

5.3. Parameter analysis

To show the effects of the simulation parameters, including the scenario parameters (e.g. map-based scenario, mobility model, routing protocol and node density), the node parameters (e.g. buffer size and data generation rate), the message and anti-packet parameters (e.g. message size, anti-packet size and *TTL*) and the parameter *ATT* in SuDA, we make a large number of supplementary experiments and analyse the performances of all the replica deletion schemes. Considering the limited space, we show the overall performance criteria, that is, the data delivery ratio and the average delivery delay, at 2400 s using different schemes under different environment configurations.

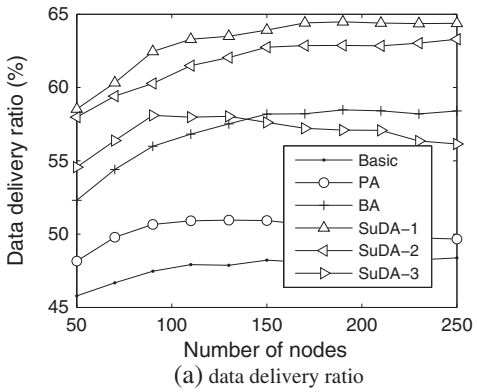
5.3.1. Scenario parameter analysis.

In order to see the effects of the scenario parameters, on the one hand, we conduct three groups of comparative experiments using different maps (e.g. grid map and real map), mobility models (e.g. bus movement model and RWP movement model) and routing protocols (e.g. Spray&wait, Prophet and GeoSpray). The default values for these three parameters, that is, the map, the mobility model and the routing protocol, are the grid map, the bus movement model and the Spray&wait routing protocol, respectively, as shown in Table II. In each comparative group, only one parameter is variable while the other two parameters have the default values for a clear analysis of each parameter. The results are listed in Table III. On the other hand, we range the number of vehicular nodes from 50 to 250 to see the influence of node density on the overall performance. The results are shown in Figure 5.

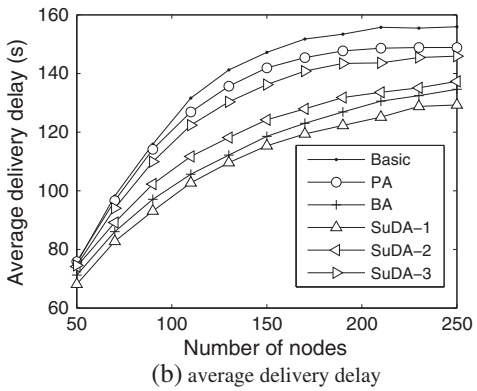
From Table III, we see that in a real map of Haidian District (Beijing, China), which is obtained from OpenStreetMap project [28, 29], SuDA also has the best performance as it does in the grid map. Besides, the RWP mobility model weakens the advantage of SuDA because

Table III. Scenario parameter analysis.

	Data delivery ratio (%)						Average delivery delay (s)					
Scenario	Basic	PA	BA	SuDA-1	SuDA-2	SuDA-3	Basic	PA	BA	SuDA-1	SuDA-2	SuDA-3
Grid map	47.5	50.8	60.2	62.3	59.8	58.0	115.6	114.4	98.3	93.2	102.6	110.2
Real map	42.8	45.8	55.6	58.8	55.0	54.9	120.3	119.8	105.2	100.1	109.6	116.9
Bus	47.5	50.8	60.2	62.3	59.8	58.0	115.6	114.4	98.3	93.2	102.6	110.2
RWP	44.1	48.0	55.1	55.5	55.0	53.2	123.2	124.0	94.7	96.0	101.8	118.4
Spray&wait	47.5	50.8	60.2	62.3	59.8	58.0	115.6	114.4	98.3	93.2	102.6	110.2
Prophet	45.1	50.8	66.7	68.5	64.6	53.4	189.5	187.6	134.2	135.5	142.3	181.4
GeoSpray	56.2	62.5	69.1	70.4	67.8	63.7	101	99.6	68.7	65.1	72.5	96.8

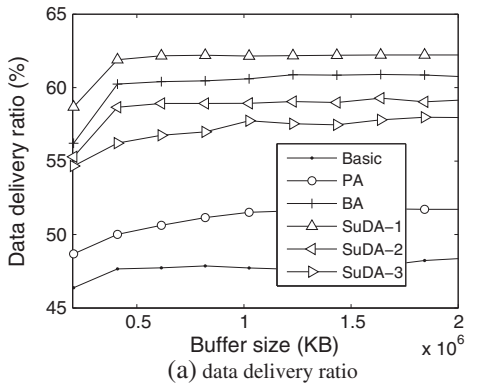


(a) data delivery ratio

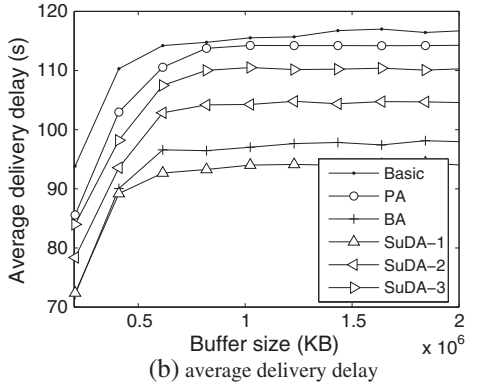


(b) average delivery delay

Figure 5. Simulation results with different node densities. (a) Data delivery ratio and (b) average delivery delay.



(a) data delivery ratio



(b) average delivery delay

Figure 6. Simulation results with different buffer sizes. (a) Data delivery ratio and (b) average delivery delay.

the random movement hinders the vehicles determining the anti-packet forwarding trace, which is according to the regular contact probability and intermeeting time. For the different routing protocols, Prophet has a higher delivery ratio than Spray&wait because of its improved data transmission based on the contact probabilities among vehicles, while its delivery delay is prolonged because of its long waiting time for a good forwarder. As a recently proposed routing protocol, GeoSpray has a higher data delivery ratio and a shorter delivery delay than Spray&wait and Prophet because it is a geographical-location-based hybrid routing approach between multi-copy and single-copy schemes. Note that using different routing protocols, SuDA-1 always outperforms other schemes.

As Figure 5(a) shows, the data delivery ratios in Basic, BA, SuDA-1 and SuDA-2 increase with an increasing number of nodes because there are more forwarder nodes to help the delivery of data messages. For PA and SuDA-3, the delivery ratios rise first and then drop down because they have fewer forwarding chances than others and a larger number of nodes increases the total number of created data messages. From Figure 5(b), we see that in all the schemes, the more vehicular nodes there are, the longer the average delivery delays are. The main reason is that more nodes increase the number of created data messages and thus the transmission competition is more fierce.

5.3.2. Node parameter analysis.

Now, we discuss the effects of the buffer size and the data generation rate on the data delivery ratio and the average delivery delay in all the replica deletion schemes. The simulation results with the buffer size ranging from 200 MB to 2 GB are shown in Figure 6, and those with the data generation interval ranging from 10 to 100 s are presented in Figure 7.

From Figure 6(a), we see that a small buffer size constrains the data delivery ratio obviously, while a very large buffer size also does no good to improve the delivery ratio. This is because when the buffer size is large enough to avoid message deletion because of storage overflow, the key factors that affect the performance enhancement change to be the limited data lifetime and the valuable communication chance. As Figure 6(b) shows, a larger buffer size increases the average delivery delay because more data replicas in node's storage space result in fewer chances for one message to be forwarded in one contact. Thus, many messages wait for a longer time to be forwarded, and their delivery delays are prolonged. Besides, compared with others, our scheme SuDA-1 performs the best with different buffer sizes, as we analysed in Section 5.2.

Figure 7 shows that the larger the data generation interval is, the higher the data delivery ratios in these replica deletion schemes are and the shorter the average delivery

Trans. Emerging Tel. Tech. **25**:881–894 (2014) © 2013 John Wiley & Sons, Ltd.
DOI: 10.1002/ett

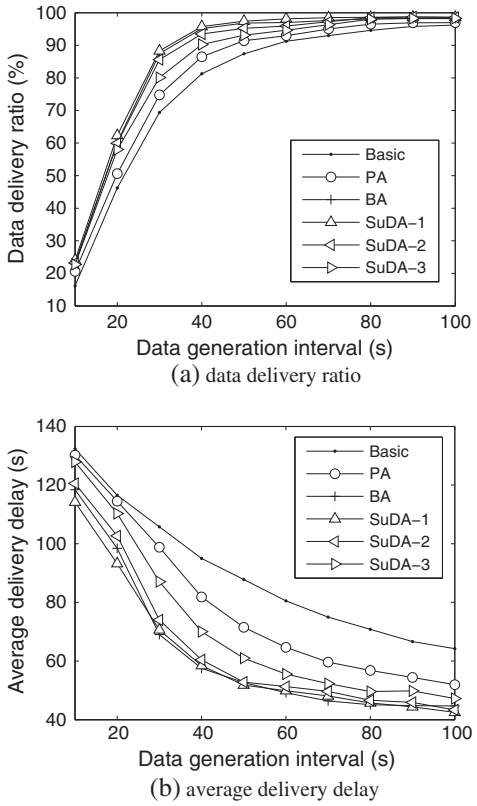


Figure 7. Simulation results with different data generation intervals. (a) Data delivery ratio and (b) average delivery delay.

delays are. It is apparent that the data generation rate varies inversely with the data generation interval. A lower data generation rate means that there are fewer data messages generated, such that the resource competition becomes weaker and more data messages can be forwarded to their destinations with a shorter latency. Besides, we see that with a low data generation rate, BA has a similar excellent performance with SuDA because the sufficient network resources enlarge the advantage of anti-packet flooding. However, SuDA-1 always keeps the highest data delivery ratio and the lowest average delivery delay with different data generation rates.

5.3.3. Message and anti-packet parameter analysis.

Here, we analyse the effects of message and anti-packet parameters on the overall performances of these replica deletion schemes. The simulation results with different message sizes and different anti-packet sizes are shown in Table IV, while the results with *TTL* ranging from 100 to 1000 s are presented in Figure 8.

The first group of comparative experiments in Table IV shows that large data messages reduce the data delivery ratio and the average delivery delay. There are two main reasons. (1) Large data messages make the resource competition fiercer, and thus, more data messages are dropped because of buffer overflow. (2) For a large data message size, each vehicle carries a small number of data replicas, and each message has a large scheduling opportunity, such that the messages can be delivered in a short delay. The second group of comparative experiments in Table IV indicates that large anti-packets result in great performance degradations of BA and SuDA-1 for a large number of resource consumptions because of substantial anti-packets in these schemes.

As Figure 8(a) shows, a longer lifetime of data message increases the data delivery ratio until it reaches some stable stage. There are two main reasons. (1) For a larger *TTL* allows data messages to wait for a longer time to reach their destinations, it raises the data delivery ratio. (2) When no data message is dropped because of *TTL* expiration, the main factors to the performance improvement change to be the limited storage space and the short contact duration. From Figure 8(b), we see that a large *TTL* increases the average delivery delay with a similar reason as analysed for buffer size. In addition, we see that our scheme SuDA-1 has obvious advantage over others in these two criteria with different *TTL* values.

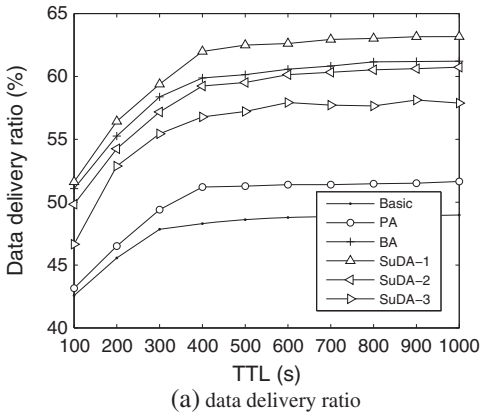
5.3.4. *ATT* analysis.

We range the assignment of *ATT* from 0% to 100%, in order to see the effect of this threshold on the performance of our proposal. The simulation results are shown in Figure 9.

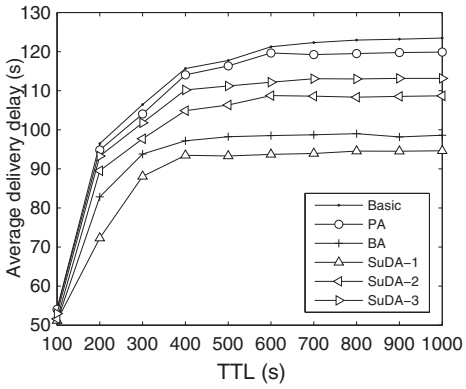
As Figure 9 shows, the value of *ATT* really affects the overall performance of the network greatly. From Figure 9(a), we see that the value of *ATT*, which results in the

Table IV. Message and anti-packet size analysis

Message size,	Data delivery ratio (%)						Average delivery delay (s)					
anti-packet size	Basic	PA	BA	SuDA-1	SuDA-2	SuDA-3	Basic	PA	BA	SuDA-1	SuDA-2	SuDA-3
5 M, 10 K	51.0	56.1	62.2	62.5	61.0	58.6	161.0	158.2	134.5	131.2	140.0	155.4
10 M, 10 K	47.5	50.8	60.2	62.3	59.8	58.0	115.6	114.4	98.3	93.2	102.6	110.2
15 M, 10 K	48.4	52.3	57.0	57.4	56.3	54.8	76.2	80.1	67.4	66.5	70.5	80.2
10 M, 5 K	47.5	52.6	64.0	64.4	63.3	60.7	115.6	110.5	90.3	90.0	99.2	106.0
10 M, 10 K	47.5	50.8	60.2	61.3	61.8	58.0	115.6	114.4	98.3	97.2	100.6	118.2
10 M, 25 K	47.5	50.3	51.2	52.2	53.8	53.6	115.6	118.2	125.5	123.2	120.4	120.2

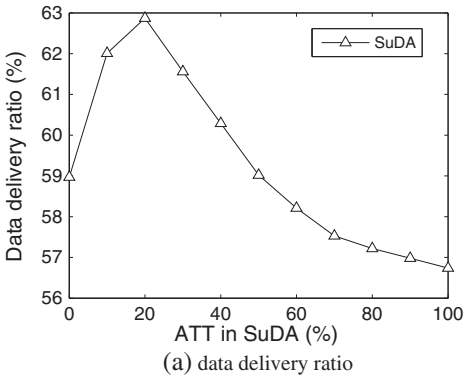


(a) data delivery ratio

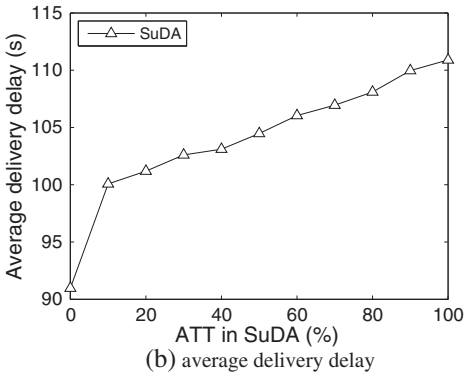


(b) average delivery delay

Figure 8. Simulation results with different *TTL*. (a) Data delivery ratio and (b) average delivery delay.



(a) data delivery ratio



(b) average delivery delay

Figure 9. Simulation results with different *ATT*. (a) Data delivery ratio and (b) average delivery delay.

largest data delivery ratio, may be an intermediate value in the interval (0%, 100%). This is because too low *ATT* increases the number of anti-packets in the network and too high *ATT* raises the number of data replicas. Besides, the specific value of *ATT* at the peak of delivery ratio is decided by the particular network configuration. As Figure 9(b) shows, higher *ATT* increases the average delivery delay because there are more data replicas in the vehicular nodes to share the scarce communication chance. By comprehensive analysis of the effects of *ATT* assignment, we know that the suitable value of *ATT* is decided by the configurations and requirements of a particular network scenario.

Lemma 2 in Section 4 gives us a qualitative principle of the assignment of *ATT*. When the network requires a short replica deletion delay and allows a large overhead of anti-packets accordingly, a lower value should be set to *ATT*; when the network cannot stand for a large overhead due to anti-packets but it can wait for a longer time to remove data replicas, a higher value should be selected for *ATT*. However, this is just a qualitative suggestion for the assignment of *ATT*. Aiming at obtaining the most suitable value for *ATT* in an actual vehicular application, we suggest that a warm-up process be conducted in advance, which utilises the sampling statistics and the trend analysis. A warm-up process is often required in the transmission mechanisms in wireless networks, such as the process in the Prophet routing protocol to compute the contact probabilities. Thus, we think that the warm-up process here is acceptable and capable of computing the suitable *ATT* value in a particular scenario.

Above all, we conclude that our proposal works well in various networks with different environment configurations. Besides, we give some suggestions about how to find a proper assignment of *ATT* in our scheme.

6. CONCLUSION

In this paper, we propose a knowledge-based replica deletion scheme with directional anti-packets in VDTNs, in order to decrease the anti-packet overhead with a high replica deletion speed. We achieve this by two algorithms, that is, the contact judgment algorithm and the directional anti-packet transmission algorithm. In the contact judgment algorithm, we firstly use time comparison to avoid invalid transmission of anti-packets, which arrive at the relay nodes having data replicas later than the expiration of *TTL*, and then use contact probability comparison to select a better forwarder between two meeting vehicles to transmit the anti-packet to each relay node. In the anti-packet

Trans. Emerging Tel. Tech. **25**:881–894 (2014) © 2013 John Wiley & Sons, Ltd.
DOI: 10.1002/ett

transmission algorithm, we analyse the selection results for all relay nodes and decide on whether to forward the anti-packet to the encountering node or not according to the anti-packet transmission rule. As the anti-packet transmission threshold *ATT* is adjustable, our scheme strikes a balance between the deletion latency and the deletion overhead to meet different requirements in various networks. Overall, our scheme SuDA improves the replica deletion efficiency and is applicable to different VDTNs.

However, there is still much profound research to be performed to improve our scheme. Considering that some attributes are used to record the relay nodes, which carry data replicas, our scheme is not suitable in the Epidemic routing protocol because Epidemic leads to a large number of relay nodes and thus greatly increases the size of these extra attributes. Here, we give a qualitative guideline to judge whether a routing protocol is suitable to use our scheme. If a routing protocol leads to a small number of nodes carrying data replicas or the size of data replicas is large, then it is more probable that our scheme works well with this protocol. The method to determine whether our scheme is suitable with some routing protocol in a particular vehicular scenario is left to our future work. Besides, more experiments in actual VDTNs are needed to test the performance of our scheme and adjust it well for real-world scenarios.

ACKNOWLEDGEMENTS

We gratefully acknowledge the support from China's Natural Science Foundations (61173009), the National High Technology Research and Development Program of China (2011AA010502), the International S&T Cooperation Program of China (2010DFB13350), the Doctoral Fund of Ministry of Education of China (20091102110017), the Science Foundation of Shenzhen City in China (JCYJ20120618170520900), the Innovation Foundation of BUAA for PhD Graduates and Fundamental Research Funds for the Central Universities.

REFERENCES

1. Garla M, Kleinrock L. Vehicular networks and the future of the mobile Internet. *Computer Networks* 2011; **55**: 457–469.
2. Li X. *Research on the Key Techniques of Vehicular Sensor Networks and Applications*, 2009.
3. Li Y, Wang Z, Jin D, Su L, Zeng L. Optimal beaconing control for epidemic routing in delay-tolerant networks. *IEEE Transactions on Vehicular Technology* 2012; **61**(1): 311–320.
4. Pereira PR, Casaca A, Rodrigues J. JPC, Soares V. NGJ, Triay J, Cervello-Pastor C. From delay-tolerant networks to vehicular delay-tolerant networks. *IEEE Communications Surveys & Tutorials* 2012; **14**(4): 1166–1182.
5. Chen Y, Li M, Shu L, Wang L, Hara T. A proportional fairness backoff scheme for funnelling effect in wireless sensor networks. *Transactions on Emerging Telecommunications Technologies* 2012; **23**(6): 585–597.
6. Harras KA, Almeroth KC. Transport layer issues in delay tolerant mobile networks. In *Proceedings of IFIP Networking 2006*. Springer: Coimbra, Portugal, May 15–19,2006; 463–475.
7. Yan Z, Zhang Z, Jiang H, Shen Z, Chang Y. Optimal traffic scheduling in vehicular delay tolerant networks. *IEEE Communications Letters* 2012; **16**(1): 50–53.
8. Schwartz RS, Barbosa RRR, Meratnia N, Heijenk G, Scholten H. A directional data dissemination protocol for vehicular environments. *Computer Communications* 2011; **34**(17): 2057–2071.
9. Vahdat A, Becker D. *Epidemic Routing for Partially-Connected Ad Hoc Networks, Technical Report CS-200006*, 2000.
10. Spyropoulos T, Psounis K, Raghavendra C. C, Spray and wait: an efficient routing scheme for intermittently connected mobile networks, In. *Proceedings of ACM SIGCOMM workshop on Delay-tolerant networking* 2005: 252–259.
11. Lindgren A, Doria A, Schelen O. Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE Mobile Computing and Communications Review* 2003; **7**(3): 19–20.
12. Burgess J, Gallagher B, Jensen D, Levine BN. MaxProp: routing for vehicle-based disruption-tolerant networks, In. *Proceedings of IEEE INFOCOM* 2006: 1–11.
13. Soares V. NGJ, Rodrigues J. JPC, Farahmand F. GeoSpray: a geographic routing protocol for vehicular delay-tolerant networks. *Information Fusion* 2011. DOI: doi:10.1016/j.inffus.2011.11.003.
14. Yaacoub E, Al-Kanj L, Dawy Z, Sharafeddine S, Filali F, Abu-Dayya A. A utility minimization approach for energy-aware cooperative content distribution with fairness constraints. *Transactions on Emerging Telecommunications Technologies* 2012; **23**(4): 378–292.
15. Lucas-Estan MC, Gozalvez J, Sanchez-Soriano J. Bankruptcy-based radio resource management for multimedia mobile networks. *Transactions on Emerging Telecommunications Technologies* 2012; **23**(2): 186–201.
16. Zhang X, Neglia G, Kurose J, Towsley D, Wang H. Benefits of network coding for unicast application in disruption-tolerant networks, IEEE/ACM Transactions on Networking. *Article first published online* Nov. 2012: 19. DOI: 10.1109/TNET.2012.2.224369.
17. Bian H, Yu H. An efficient control method of multi-copy routing in DTN, In. *Proceedings of the 2nd International*

Conference on Networks Security, Wireless Communications and Trusted Computing 2010: 153–156.
18. Yu H, Ma J, Bian H. Reasonable routing in delay/disruption tolerant networks. *Frontiers of Computer Science* 2012; **5**(3): 327–334.
19. Kaveevivitchai S, Ochiai H, Esaki H. Message deletion and mobility patterns for efficient package delivery in DTNs, In. *Proceedings of the 8th IEEE International Conference on Pervasive Computing and Communications Workshops* 2010: 760–763.
20. Tang X, Pu J, Gao Y, Xiong Z, Weng Y. Energy-efficient multicast routing scheme for wireless sensor networks, Transactions on Emerging Telecommunications Technologies. *Article first published online* May 2013: 7. DOI: 10.1002/ett.2661.
21. Pandey T, Garg D, Gore MM. Publish/subscribe based information dissemination over VANET utilizing DHT. *Frontiers of Computer Science* 2012; **6**(6): 713–724.
22. Calabuig D, Monserrat JF, Cardona N. Proportionally fair scheduler for heterogeneous wireless systems. *Transactions on Emerging Telecommunications Technologies* 2012; **23**(1): 1–5.
23. Wang B, Li H, Cao J. An efficient MAC scheme for secure network coding with probabilistic detection. *Frontiers of Computer Science* 2012; **6**(4): 429–441.
24. Keranen A, Ott J, Karkkainen T. The ONE simulator for DTN protocol evaluation, In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, 2009. Article No. 55.
25. Tang X, Pu J, Cao K, Zhang Y Y, Xiong Z. Integrated extensible simulation platform for vehicular sensor networks in smart cities. *International Journal of Distributed Sensor Networks* 2012; **2012**. article id: 860415.
26. Zhang X, Kurose J, Levine BN, Towsley D, Zhang H. Study of a bus-based disruption-tolerant network: mobility modeling and impact on routing, In. *Proceedings of ACM International Conference on Mobile computing and networking* 2007: 195–206.
27. Bettstetter C, Resta G, Santi P. The node distribution of the random waypoint mobility model for wireless ad hoc networks. *IEEE Transactions on Mobile Computing* 2003; **2**(3): 257–269.
28. Haklay M, Weber P. OpenStreetMap: user-generated street maps. *IEEE Pervasive Computing* 2008; **7**(4): 12–18.
29. Bennett J. *OpenStreetMap*. Packt Publishing Ltd. 32 Lincoln Road Olton Birmingham: UK. ISBN 978-1-847197-50-4.