Secure Cooperative Data Downloading in Vehicular Ad Hoc Networks

Yong Hao, Jin Tang, *Member, IEEE,* and Yu Cheng, *Senior Member, IEEE*

***Abstract*—In this paper, we propose a secure cooperative data downloading framework for paid services in vehicular ad hoc networks (VANETs). In our framework, vehicles download data when they pass by a road side unit (RSU) and then share the data after they travel out of the RSU's coverage. A fundamental issue of cooperative data downloading is how vehicles effectively share data with each other. We develop an application layer data sharing protocol which coordinates the vehicles to relay data for sharing according to their positions. Such coordinated sharing can avoid collisions in the medium access control (MAC) layer and the hidden terminal issue in multi-hop transmissions. A salient feature of the proposed sharing protocol is that it can guarantee the receipt of the requested data file for each applicant vehicle passing a road side unit. Moreover, we also address security and privacy issues in the process of data downloading and sharing, ensuring applicants' exclusive access to the applied data and privacy of the vehicles involved in the application. We carry out NS2 simulations to thoroughly examine the performance of the proposed cooperative downloading protocol implemented over an 802.11p based VANETs.**

***Index Terms*—Vehicular ad hoc networks, data downloading, paid services, security, privacy.**

I. Introduction

DATA downloading[1] is a promising and practical application in vehicular ad hoc networks (VANETs), which can bring convenience and entertainment to users. In a data downloading application, vehicles send requests for services and then get the data from current or the following road side units (RSUs).

In the data downloading application, the amount of data that a vehicle can download in one drive-through from a RSU is very limited, due to the short connection time. *Cooperative downloading* is a promising scheme in which vehicles download data when they pass by an RSU and then share the data when they travel out of the communication range of the RSU. In this manner, the total amount of data that a certain vehicle can download will be increased. A fundamental issue in the cooperative downloading is how vehicles share data with others. There are some existing studies on data sharing in VANETs [2]–[4]. However, the existing sharing protocols are constrained with issues of medium access control (MAC) layer collisions, limited applicability to sharing multiple data units, and no guarantee of complete data receiving.

We propose an application layer data sharing protocol with an assumption that each vehicle knows the positions of itself and its neighboring vehicles (which can be obtained through a global positioning system (GPS) device and safety related messages regularly broadcasted by neighboring vehicles [5]). In the proposed protocol, vehicles employ a coordination channel to coordinate the relay transmissions in the VANETs for data sharing based on vehicles' GPS locations. With such cooperative sharing, the MAC layer collisions and the hidden terminal effect can be avoided in the data channel. Furthermore, we design an elegant relay vehicle selection mechanism so that the space between two RSUs can be fully exploited for data sharing. A salient feature of the proposed sharing protocol is that it can guarantee the receipt of the requested data file for each applicant vehicle passing an RSU.

Security and privacy are also critical issues. In the paid services, vehicles purchase data, such as a popular video clip, from service providers via RSUs. The characteristics of the paid services require applicants' exclusive access to the corresponding data. In this paper, we exploit a novel security protocol called broadcast encryption (BE) [6] to encrypt the data that only the applicants can decrypt. The overhead of BE is more reasonable than an intuitive method. Moreover, our framework also ensures vehicles' privacy which is considered as one of the most important security features of VANETs. Specifically, in a privacy preserved system, eavesdroppers should not be able to link any two messages sent by the same vehicle [7]. In summary, we develop a secure cooperative data downloading framework with the following contributions:

1) We design an application layer data sharing protocol to facilitate data downloading. With coordinated relay transmission in sharing, MAC layer collisions and the hidden terminal issue are avoided.
2) Security and privacy protocols for paid services in VANETs are developed, which can guarantee the applicants' exclusive access to the applied data and the privacy of the vehicles involved in the application.
3) Analytical models are derived to quantitatively evaluate the impact of the distance between RSUs on the amount of data that a vehicle can download in a drive through.
4) We carry out extensive NS2 simulations of 802.11p based VANETs to examine the performance of the proposed cooperative data downloading protocol, in terms of average throughput, proportion of guaranteed receiving, and instantaneous downloading rate.

Manuscript received March 1, 2012; revised July 23, 2012.

Y. Hao was with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, 60616, USA. He is now with Juniper Networks (e-mail: yhao4@iit.edu).

J. Tang was with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, 60616, USA. He is now with AT&T Labs (e-mail: jin.tang@att.com).

Y. Cheng is with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, 60616, USA (e-mail: cheng@iit.edu).

This work was supported in part by NSF grant CNS-1053777.

Digital Object Identifier 10.1109/JSAC.2013.SUP.0513047

The rest of the paper is organized as follows. Section II reviews the related work. Section III describes the system model. In Section IV and Section V, we present the secure data downloading protocol and the efficient data sharing protocol, respectively. The security performance of our framework is analyzed in Section VI. Section VII presents the simulation studies. The conclusion remarks are given in Section VIII.

II. Related Work

A. Efficient Data Sharing

There are a few studies, [2]–[4] and the references therein, on efficient data sharing in VANETs. For most of the existing work, the performance of data sharing is constrained by the MAC-layer collisions and the hidden terminal issue in multi-hop transmissions. The RTS/CTS dialogue proposed in [8] can mitigate data collision by exchanging short RTS/CTS packets before data transmission. However, it can only be used in unicast communication which is not efficient. Several TDMA protocols have been proposed in [9], [10] to address the MAC-layer collisions and the hidden-terminal issue. A constraint of these protocols is that time synchronization should be autonomously ensured among all mobile stations. *Broadcast* is an efficient way to maximize the system throughput especially in the scenario when several receivers are located in the communication range of the sender. The authors in [4] extend the concept of RTS/CTS to relay-RTS/relay-CTS that is suitable for broadcast. A protocol that defines an optimal relay set and a backoff mechanism is also proposed in the paper to optimize the performance. However, in their protocol, the overall performance is constrained by the time that vehicles pass by RSUs.

B. Security and Privacy

In the paid services, applicants' exclusive access to the applied data must be protected. An intuitive method is that each applicant reports a public key in the request message and then the service provider distributes encrypted data to the corresponding applicants [11], [12]. This method is acceptable as long as the data is not frequently applied. But, it is not scalable. In addition, in these papers, unicast is employed in the data distribution procedure. It is less efficient than the broadcast when multi-users apply for the same data.

Privacy is regarded as one of the most important security requirements of VANETs. Currently, two major categories of techniques are proposed to provide privacy for vehicles which are group signature based protocols [5] and pseudonym based protocols [13]. However, neither of them can guarantee vehicles' privacy in the data downloading system. The authors in [14] propose a protocol based on pseudonym exchange called AMOEBA to provide privacy for vehicles. In AMOEBA, vehicles form groups and a group leader will be chosen randomly in each group. All messages that are transmitted between group members and RSUs should be forwarded by the group leader. Therefore, the group leader may cache and shuffle several requests and responses. In other words, group members' privacy is protected by sacrificing that of group leaders. However, group leaders are randomly selected vehicles who may reveal group members' privacy.

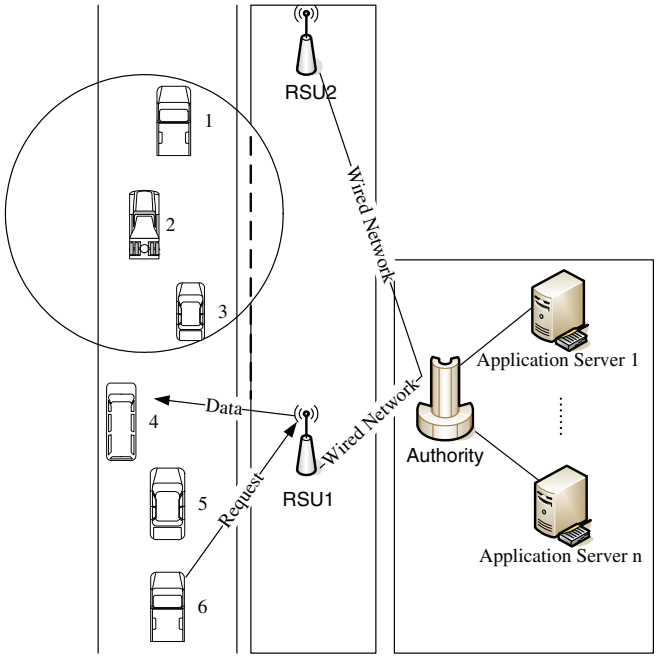


Fig. 1. Vehicular ad hoc networks.

Our previous work [15] also presents a protocol to provide privacy for vehicles by introducing some randomness. But it only supports unicast communication.

III. System Model

A. Network Model

Network entities can be classified into three categories [16] in VANETs, the authority and application servers, road side infrastructure and nodes, as shown in Fig. 1.

The authority and application servers are powerful workstations which are responsible for management and service data provision respectively. The authority knows all keys and is in charge of service scheduling. Application servers provide service data to vehicles. They can be maintained either by the authority or by third party operators. We assume that the authority and application servers have powerful processing ability. Thus, we ignore their computation time in this paper.

Road side infrastructure consists of RSUs deployed at the road sides which are in charge of data collection and distribution. RSUs are connected to the authority through wired network and communicate with vehicles through radio.

Nodes are ordinary vehicles on the road that can communicate with each other and RSUs through wireless. We assume that each vehicle is equipped with a differential GPS receiver with an accuracy on the order of one meter [17], [18] and an on board unit (OBU) which is in charge of all communication and computation tasks.

In this paper, we focus on non-real time data downloading in the highway scenario. We assume that RSUs are deployed along the highway which are at least several kilometers far from each other. On the highway, maybe some vehicles travel faster or slower than average, but we assume the majority of vehicles travel in platoons with similar velocities. The velocity of a platoon (V) is defined as the median value of

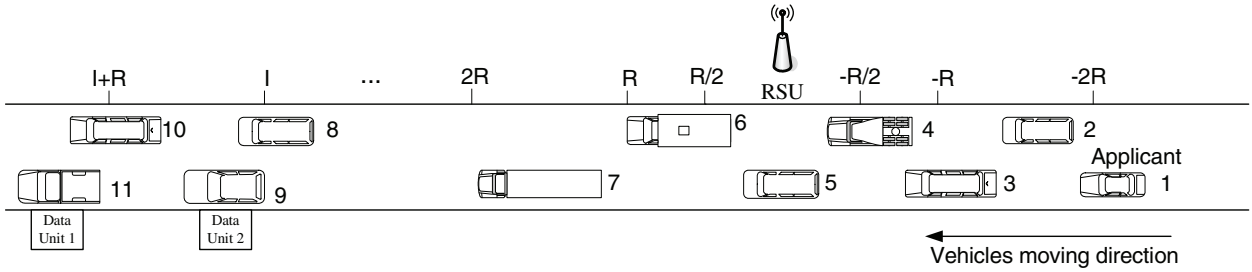


Fig. 2. Data downloading framework.

vehicles' speeds in the platoon [13]. We further assume that the density of vehicles reaches a threshold, such as 50 vehicles per kilometer [19] and leave the low density case to the future work. A *file* is defined as a particular block of information that vehicles request to download. Vehicles may apply for several files when they pass by an RSU.

1) Channel Assignment: In the VANETs, vehicles share the wireless spectrum according to the 802.11p [20], [21] which has seven 10-MHz wide communication channels. Among these seven channels, one channel is used as the *control channel* which is reserved for short, high-priority messages. The two channels at the edges of the spectrum are reserved for future usages. The remainder is service channels which are available for both safety and non-safety applications. In this paper, we define one service channel as *coordination channel* in which vehicles periodically broadcast their geographic information every 300 milliseconds. Some control messages of our protocol will also be transmitted in this channel[1]. Two other service channels, named *data channels*, will be employed to share data among vehicles. We assume that vehicles and RSUs use the same transmission power in all channels with communication range R. Moreover, we further assume that multi-radios are equipped in each vehicle and RSU, so vehicles and RSUs are able to send and receive messages in multi-channels simultaneously [22], [23].

It is worth noting that in a practical VANETs, different vehicles may have different numbers of radios and the number of radios is normally smaller than the number of available channels. However, it is well-known that such a general multi-radio multi-channel (MR-MC) network incurs a very complex resource allocation issue for channel assignment and scheduling, which is NP-hard in general [24]–[26]. In this paper, we resort to a simplified scenario, where each vehicle has a separate radio for the coordination channel and two data channels, to better demonstrate the benefit of exploiting multiple channels for efficient cooperative data sharing. Extending our study to a more general MR-MC environment is an interesting future topic.

2) Reliable Communications: Broadcast is utilized in our framework. However, there is no acknowledgment in the broadcast. Thus, reliable communications for a certain packet can not be guaranteed. In this situation, coding methods, such as fountain codes [27], can be used to counteract the effect of packet loss in the wireless channel. In order to transmit a message which is comprised of k symbols to the receiver, the sender will encode the k original symbols to $\hat{k}$ encoded symbols and then send out these encoded symbols. As long as the receiver gets at least $(1+\epsilon)k$ encoded symbols, the original message can be recovered, where $1+\epsilon$ is the decoding inefficiency [28], [29]. If we know that the packet deliver ratio in the wireless channel is P_{pdr}, $\hat{k}$ should be at least $(1+\epsilon)k/P_{pdr}$ to guarantee the reception at the receiver side. Note that, the computation overhead that is introduced by the fountain code encoding and decoding is negligible [28].

In our framework, coding methods will be only employed in the data channels for efficient data sharing [30]. In order to be compatible with other applications, messages in the coordination channel will be transmitted without being encoded.

B. Cooperative Data Downloading Framework

1) Vehicles Classification: We classify vehicles into three types according to their roles in our framework.

Applicants are vehicles that purchase data.

Downloading vehicles are vehicles who download data from RSUs for applicants. They are assigned by the authority according to their geographic positions.

Relay vehicles are responsible for forwarding data to buyers which are more than one-hop away from downloading vehicles. Moreover, because some popular data may be applied by multiple vehicles, the existence of relay vehicles can prevent RSUs from distributing the same content repetitively.

2) Downloading and Sharing: We propose a secure cooperative data downloading framework for paid services in VANETs. The proposed framework is comprised of two major parts, secure data downloading and efficient data sharing.

In our framework, RSUs periodically broadcast hello messages in the coordination channel. Vehicles will forward the first RSU's hello message that they hear for one time by piggybacking it on the next geographic message. As shown in Fig. 2, when a vehicle, such as vehicle 1, hears the forwarded hello message from the vehicle 3, it will broadcast one-hop request messages in the coordination channel. Its neighbors who get the request message will forward it to the RSU. After receiving the request message, the authority will define some downloading vehicles and distribute a *data unit* to each of them. We define the time that a downloading vehicle needs to download a data unit as *a time unit* (δ).

When these downloading vehicles are at least $I+R$ far from the RSU, they will start to share the data one by one, where I is the interference range of RSUs and vehicles. The distance $I+R$ guarantees that all receivers in the communication

[1]To the best of our knowledge, current 802.11p WAVE communication standard does not have very clear definitions on how to incorporate new customized functions, which deserves further studies.

range of the downloading vehicle can receive data without collisions and interference from the RSU. The data sharing procedure is shown in Fig. 2. Vehicle 11 (downloading vehicle 1) will share the data unit 1 by broadcasting. After that, it will choose a relay vehicle who can further forward the message to applicants that are several hops away. In our framework, a certain vehicle may serve both as a downloading vehicle and a relay vehicle. For instance, if vehicle 11 selects vehicle 9 (downloading vehicle 2) as the relay vehicle, vehicle 9 will broadcast both data unit 1 as a relay vehicle and data unit 2 as a downloading vehicle after vehicle 11 finishes sharing. If vehicle 11 selects vehicle 8 as the next relay vehicle, vehicle 8 will broadcast data unit 1 after vehicle 9 shares the data unit 2. A *forward sharing process* is also included in our protocol to transmit data units from back to front.

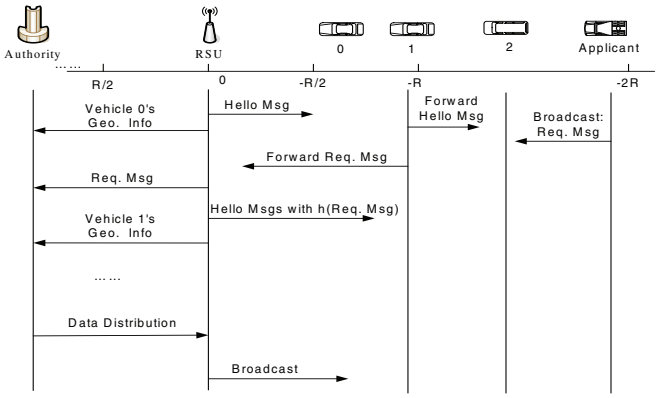


Fig. 3. Data downloading.

C. Security Model

1) Short Group Signature: We resort to the short group signature scheme [5] for privacy provision in this paper. With short group signature, members of a group sign messages under the name of the group. In a group, there is one group public key and many corresponding group private keys. A message that is signed by any group private keys can be verified with the unique group public key, and the signer's identifier will not be revealed. However, the authority holds a tracing key which can be used to retrieve the group private key from the signature. If one group private key is assigned to only one user, the signer can be identified after the authority gets its group private key.

2) Broadcast Encryption: is a security technique to encrypt broadcast content in such a way that only qualified users can decrypt it. In the broadcast encryption, unsubscription of some users will not affect the remaining users. Moreover, the system are secure against any number of colluders. In this paper, the scheme [6] that we exploit has short ciphertexts. Therefore, it is more efficient in communications.

We further assume that vehicles help each other to download and share data in a cooperative manner. In the real application, some incentive techniques [31] can be employed to stimulate the cooperation. Due to the limitation of the space, we will not discuss incentive techniques in this paper.

IV. Secure Data Downloading

The data downloading procedure is composed of two phases, request and downloading. Vehicles apply for files in the request phase. Then in the data downloading phase, the authority will define some downloading vehicles. Each downloading vehicle will download a different unit of data for applicants when it passes by an RSU. We will give detailed message formats and procedures in each phase. The data downloading flow chart is shown in Fig. 3.

A. Request Phase

RSUs periodically broadcast hello messages in the coordination channel. Each vehicle will forward the first hello message it receives from the RSU for just one time by piggybacking it on its next geographic message. With such a relay operation of the hello message, an applicant vehicle can detect the RSU when it is still out of the communication range of the RSU, and starts requesting data early (the data request can be relayed to the RSU by a predecessor vehicle to the applicant). This request mechanism allows the authority to know in advance the files that an applicant wants to buy, and can make the data ready before the applicant vehicle starts downloading in the communication range. The request message format is shown in Table I. "Request Data Names" are files that the applicant would like to buy. "Geo. Info." is vehicles' current geographic information including their current GPS locations and velocities. "T" is the current time stamp. The GPS location, velocity and the time stamp can facilitate the authority to predict applicants' approximate locations in the near future. These three items will be encrypted by authority's public key. Finally, the entire message is signed by applicant's group private key. We will discuss security techniques in details later. When applicants' neighboring vehicles get the request message, they will forward it to the RSU. Note that a random initiation scheme which was proposed in our previous work [16] can be employed to avoid collisions. If the size of the file is small and there's only one applicant who wants the data, the authority can distribute the file to the applicant directly without employing other vehicles. However, if the size of the file is big or the authority has plenty of files to distribute, the authority can select some downloading vehicles to facilitate downloading. In this paper, we focus on the latter scenario.

When the RSU gets request messages, it will verify the signature and send the fresh requests to the authority. The authority will retrieve applicants' identities from signatures by utilizing the tracing key. If applicants are qualified to purchase the data, the authority will require the data from the service provider. After the RSU forwards a fresh request message to the authority, it will calculate the hash value of the request message h(request message) and include the hash value in the hello messages. When the applicant and its neighbors hear the hash value of its request message, they know that the RSU has received the request. The applicant which does not hear the hash value of its request message within a certain period of time will send it again. In the request phase, the RSU will also forward each vehicle's geographic information to the authority for the downloading vehicles' selection.

TABLE I
MESSAGE FORMATS

Request Message Format

Encrypt (Request Data Names, Geo. Info. , T)	GSignature

Distribution Data Format

Cipher	Names & Sec. No.	Encrypted Msg.	W	T	Signature

Data Sharing Message Format

Distribution Data	Hop Number	T	Gsignature

B. Downloading Phase

In our framework, the authority selects downloading vehicles according to their positions. A downloading vehicle will occupy a data channel to download a unit of data and then release the channel to the next downloading vehicle to grab a different data unit. The downloading vehicles will forward the data to related applicants through the sharing procedure to be discussed in Section V. We design the distance between two adjacent downloading vehicles to be the communication range R. In this way, a downloading vehicle can forward its data to the next downloading vehicle in just one hop for an efficient sharing procedure. However, in the real application, it may not be always possible to pick the downloading vehicle exactly according to the distance R because the vehicle speed is generally not a constant. Therefore, we design the following protocol as an approximation.

As shown in Fig. 3, the RSU starts to broadcast data unit 1 to the first downloading vehicle of a platoon when it passes the point $-R/2$. According to our philosophy, the first downloading vehicle will spend R/V seconds (termed as a time unit δ) to download the data unit 1. When the first downloading vehicle finishes data downloading, the authority will choose the vehicle which is the closest to the point $-R/2$ as the second downloading vehicle and start to broadcast data unit 2 as soon as the second downloading vehicle is selected, so on and so forth. Note that we let RSUs distribute data when vehicles are located in $[-R/2, R/2]$ instead of $[-R, 0]$ because vehicles have comparatively higher signal to interference plus noise ratio (SINR) in this area.

After the downloading vehicle selection, the authority starts to create distribution data. As shown in Table I, "$Cipher$" is the cipher-text of the encryption key which is represented in equation (1). We will discuss it later in the security part. The names and the section numbers of the data are presented in the item "Names & Sec. No.", for instance, Movie: Titanic, section one. Next item is the encrypted data chunk. The RSU's identity W and the distribution time T are also included in the message for future index. Finally, the authority signs the message with its own private key.

When the data have been created, the authority starts to distribute data to downloading vehicles. It will send a unique index $\phi = h(\hat{K}_i \mid\mid T)$ and an expected sending time as well as the distribution data to the RSU, where $\hat{K}_i$ is vehicle i's preloaded secret key that distributed by the authority. The RSU will encode the distribution data by fountain codes and start to broadcast it in a data channel at the expected sending time with the index ϕ attached. The downloading vehicle knows it is selected by the authority when it receives ϕ.

The authority should tell each downloading vehicle how many hops away the data unit will be transmitted according to the distribution of applicants. For example, in Fig. 2, if vehicle 10 and 7 apply for the same file, the authority will estimate the number of hops (ξ) it takes from vehicle 11 to vehicle 7 and inform the value ξ to vehicle 11. The concept of ξ is similar to TTL in computer networks. A data unit will be forwarded by ξ relay vehicles and then be discarded by the last relay vehicle. In this way, the RSU only needs to distribute the data unit once, but both applicants can get this data unit through data sharing. However, in the real application, restricted by the geographic conditions, the authority may have no idea how many hops away a data unit will be transmitted when it distributes the data unit. For instance, in our example, when the authority distributes data unit 1 to vehicle 11, it does not know that vehicle 7 will apply the data. Therefore, the number of hops that each data unit will be transmitted must be updated by the authority according to the arrival of applicants. The hop information will be spread out in the coordination channel by employing message propagation protocol which will be presented in the next section. Note that, in the real application, if the density of RSUs is large enough or the density of vehicles can not reach the threshold, the authority will not trigger the cooperation by setting the value ξ to 0.

When vehicles apply for several files, for example, some vehicles apply for file A while some other vehicles request for file B, the scheduling method at the RSU side is still an open issue. We leave it to the future work. In this paper, we utilize the *default scheduling protocol* in which RSUs distribute a data unit for each requested file first. If RSUs receive multiple requests at the same time, they will send data in a first come first serve manner. Round-robin scheduling will be employed to send packets when RSUs have the ability to send more data.

C. Security in Data Downloading

1) The Concept of the Domain: In our framework, domain is a concept related to vehicles' management. Vehicles that are registered in a city may form a domain. We define A domains in our framework and B members are allowed in each domain at maximum. The broadcast encryption protocol [6] is integrated into our framework. For the convenience to the readers, in this section, we will give a brief introduction to both encryption and decryption procedures.

2) Key Generation: Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be addition and multiplicative cyclic groups with a generator g_1 and g_2 of the same prime order p, respectively. Let ψ be a computable isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$, with $\psi(g_2) = g_1$. Let e: $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ denotes a bilinear map constructed by modified Weil or Tate pairing. The authority finds $P \in \mathbb{G}_1$, $s \in \mathbb{Z}_p^*$ and computes sP. sP is the authority's public key (P_u) and s is its private key (P_r). The authority also chooses a cryptographic hash function: $h(.)$, such as SHA-1. We list some physical meanings of symbols in Table II.

Then, the authority generates a group public key G_{pub} for the system, a group private key G_i and a secret key $\hat{K}_i$ for the vehicle i [5]. After that, encryption keys for broadcasting will be created by choosing a random number $\alpha \in \mathbb{Z}_p^*$ and $Q \in \mathbb{G}_1$. It computes $Q_l = \alpha^l Q \in \mathbb{G}_1$ for l =1, 2, ...,

TABLE II
NOTATIONS AND DESCRIPTIONS

Notations	Descriptions
G_{pub} / G_i	Group public key / group private key for vehicle i
$GSig(M)$	Vehicles sign msg. M by using its group private key
P_u / P_r	The authority's public key / private key
$\hat{K}_i$	Vehicle i's preloaded secret key
$h(.)$	A hash function, such as SHA-1
A / B	The number of domains / The size of each domain
a	The domain indexes of a certain vehicle
b	The membership indexes of a certain vehicle
Y_i	The private decryption key for vehicle i
ξ	The number of hops a data unit will be transmitted
R	The communication range of RSUs and vehicles
I	The interference range of RSUs and vehicles
W	The identity of the RSU
K_d	The data encryption key which is used to encrypt data by symmetric key cryptography, such as AES

B, B+2, ..., 2B. Next, it picks random numbers $\beta_1, ..., \beta_A \in \mathbb{Z}_p^*$ and sets $V_1 = \beta_1 Q$, ..., $V_A = \beta_A Q \in \mathbb{G}_1$. Through this way, the public key for broadcast encryption Pub_{BE} = $(Q, Q_1, ..., Q_B, Q_{B+2}, ..., Q_{2B}, V_1, ..., V_A) \in \mathbb{G}_1$ is derived. For a certain user $i \in \{1, ..., N\}$ that can be written as $i = (a-1)B + b$ for $1 \leq a \leq A$ and $1 \leq b \leq B$, the corresponding private decryption key is: Y_i = $\beta_a Q_b \in \mathbb{G}_1$. Another expression of Y_i is $\alpha^b V_a$ [6]. Finally, the authority publishes (p, $\mathbb{G}_1$, $\mathbb{G}_2$, e, P, sP, Q, $h(.)$, Pub_{BE}, G_{pub}) and keeps s, $\beta_1, ..., \beta_A$ as secrets. Meanwhile, the group private key G_i, the secret key $\hat{K}_i$ and the private decryption key Y_i will be preloaded to vehicle i off-line securely.

3) Data Encryption: In the data distribution phase, the authority will start to encrypt the data after the downloading vehicle selection. The authority will choose a random number $t \in \mathbb{Z}_p^*$ and compute $K_d = e(Q_{B+1}, Q)^t \in \mathbb{G}_2$. Then, the purchased file will be encrypted by K_d with symmetric key encryption protocols, such as AES. Note that, K_d will be utilized to encrypt all sections of a certain file. In other words, once an applicant derives K_d, it can decrypt the whole file by using the key K_d.

After that, as shown in Table I, the authority will encrypt the key K_d to get the *Cipher* that only applicants can decrypt it. For each $l = 1, ..., A$, we define $S_l^{'}$ as the set that contains all users in the lth domain and S_l as the set that contains the indexes of those users relative to the beginning of the lth domain. Formally, $S_l^{'}$ and S_l can be expressed as following:

$S_l^{'} = S \cap \{(l-1)B + 1, ..., lB\}$ and

$S_l = \{x - lB + B \mid x \in S_l^{'}\} \subseteq \{1, ..., B\}$

Where S is the set for all vehicles, $S = 1, 2, ..., AB$.

Then, the authority encrypts the key K_d by computing [6]

$$\textit{Cipher} = (tQ,\ t(V_1 + \sum_{j \in S_1} Q_{B+1-j}), ..., t(V_A + \sum_{j \in S_A} Q_{B+1-j})) \quad (1)$$

The ciphertext of a domain exists in the *Cipher* only when there is at least one vehicle in this domain applies for the data. Therefore, the number of items in the *Cipher* is $z + 1$, where z is the number of domains that all applicants belong to.

4) Data Decryption: When an applicant receives the "distribution data" in the data sharing, it will try to decrypt the data by using its own private decryption key Y_i. The *Cipher* part of the message can be expressed as *Cipher* $= (C_0, C_1, ..., C_A)$ and recall that $i = (a-1)B+b$ for $1 \leq a \leq A$ and $1 \leq b \leq B$, then, the applicant will derive the key by [6]

$$K_d = e(Q_b, C_a)/e(Y_i + \sum_{j \in S_a,\ j \neq b} Q_{B+1-j+b}, C_0) \quad (2)$$

After the key K_d is derived, it can get the applied file. Note that, both encryption and decryption that are shown above are procedures to encrypt and decrypt one file. If a data unit contains several files, encryption and decryption will be implemented multi-times.

D. Security Protocol Overhead

In this section, we will analyze the overheads induced by security protocols. Moreover, the comparison between the broadcast encryption protocol and the intuitive method will also be given. We let p have 256 bits, which is equivalent to the security level of 128 bits AES protocol.

1) Communication Overhead: The dynamic part in the distribution data is the $Cipher$ item. Therefore, we focus on the length of this item. Let m be the number of applicants in a sharing group. q_a is the probability of vehicles from foreign domains rather than the local domain. The worst case happens when each of the vehicle from foreign domains is from a different domain. In this scenario, according to equation (1), the size of $Cipher$ is

$$\begin{aligned} |Cipher|_{BE} &= |Q| + |Q| + mq_a|Q| \\ &= 33 + 33 + 33mq_a \end{aligned} \quad (3)$$

where $|Q|$ is the size of the point Q. In our protocol, $|Q|$ equals to 33 bytes. The Fig. 4(a) shows the communication overhead of $Cipher$ item in our protocol. From the figure, we can see that, with one hundred applicants and q_a equals to five percent, the size of $Cipher$ is less than 250 bytes.

An intuitive way to achieve paid services is that each vehicle reports a public key to the authority in the request message. When the authority distributes data, it will encrypt the message by using key K_d, and then encrypt the data encryption key K_d by the reported public key. If ElGamal on elliptic curves is employed, the cipher text is 66 bytes for each applicant. Therefore, the overhead ratio between intuitive method and broadcast encryption is

$$Ratio = \frac{66 + 33mq_a}{66m} = \frac{2 + mq_a}{2m} \quad (4)$$

The $Cipher$ size in our protocol is explicitly small compared with that in the intuitive method. In Fig. 4(b), it is clearly shown that the more vehicles applying for the data, the bigger advantages can be gained.

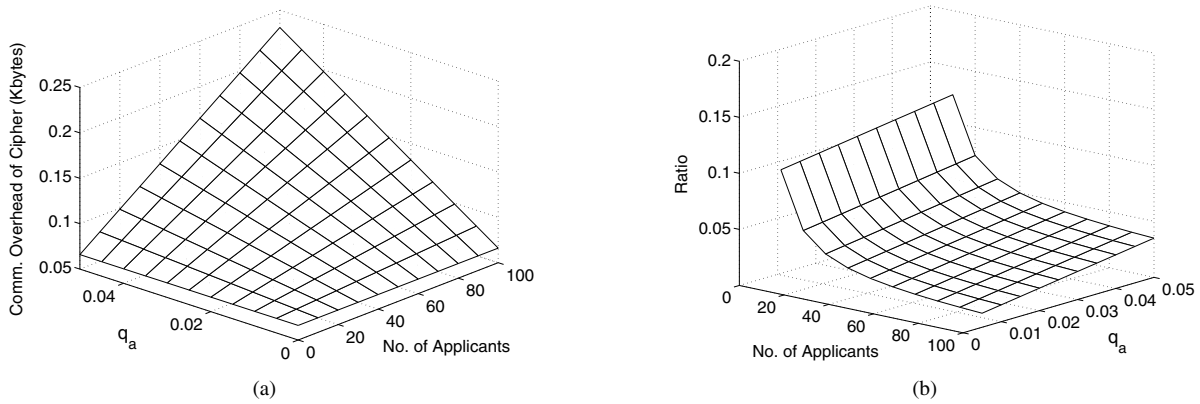


Fig. 4. Communication overhead of cipher. (a) Our protocol. (b) Comparison between our protocol and intuitive protocol.

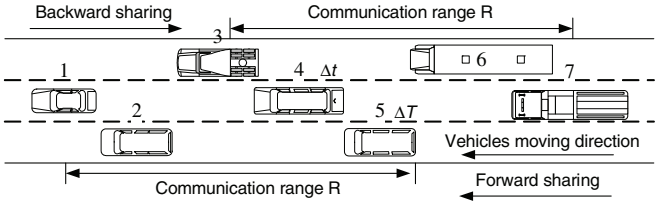


Fig. 5. Message propagation protocol.

2) OBU Storage Overhead: When the total number of vehicles (the size of S) is held constant, the number of domains (A) and the size of each domain (B) is a tradeoff between communication overhead and storage. In the broadcast encryption, each vehicle has to preload A+2B-1 public keys. For example, if there are one hundred thousand domains and one hundred thousand users in each domain. In a vehicle, the public key size

$$|Pub_{BE}| = 2 \times 10^5 \times |Q| + 10^5 \times |Q| \approx 10Mbytes \qquad (5)$$

V. Efficient Data Sharing

We propose a location based data sharing protocol in which downloading vehicles share their data units downloaded from RSUs with applicants. Relay vehicles will be employed to facilitate data sharing by forwarding data units to applicants who are several hops away from downloading vehicles. In our protocol, downloading vehicles and relay vehicles share data one by one in sequence according to their positions. Therefore, collisions can be avoided. Our data sharing protocol is comprised of two parts, backward sharing and forward sharing. As shown in Fig. 5, we define message propagation with a direction opposite to vehicles' moving direction as backward sharing. Otherwise, we call it as forward sharing. Backward sharing and forward sharing are independent each of which occupies a dedicated data channel.

A. Message Propagation Protocol

Before we go to details of our protocol, we would like to present a message propagation protocol which will be used in the data sharing. It is based on the assumption that vehicles know each other's geographic information.

Let us consider a scenario that a source node needs to broadcast a message to a destination node which is several hops far away from it. After the source node broadcasts a message with its geographic information attached to it, all the receiving nodes will check their current location from the GPS. The node which is the farthest one to the source node with the distance between them less than R will broadcast within a certain time ΔT. The value of ΔT is a heuristic value in which most vehicles can finish message forwarding and it varies in different applications. For example, in our protocol, ΔT is 15ms which includes signature verification time(11ms [16]), message transmission time and so on. As shown in Fig. 5, vehicle 1 needs to broadcast a message to vehicle 7. After vehicle 1 sends, its neighbors will figure out who should send next according to sender's geographic information. Based on our protocol, vehicle 5 should send. If vehicle 5 sends the message, other vehicles will keep silence. However, in the case vehicle 5 does not broadcast in time, the vehicle next to it which is closer to vehicle 1 will broadcast. Except for the farthest one, all the rest vehicles have a time slot of Δt to send out the message until a vehicle broadcasts. Δt equals to ΔT minus signature verification time. If the source vehicle does not hear any broadcast messages in a certain period of time, it will rebroadcast the message with updated geographic information. In the figure, finally, vehicle 3 transmits the message to vehicle 7 because vehicle 4 and vehicle 5 did not broadcast in time.

B. Relay Vehicle Selection

Relay vehicle selection is the foundation of the data sharing protocol. Thus, we discuss it before we introduce the protocol in details. If a data unit needs to be shared with applicants that are ξ hops away from a downloading vehicle, ξ-1 relay vehicles will be employed. Before the downloading vehicle (or the i_ath relay vehicle) finishes broadcasting, it will start to select the first relay vehicle (or the (i_a+1)th relay vehicle) in the *coordination channel*. The downloading vehicle will draw an arc which is R meters behind it. Several vehicles which are closest to the arc will be chosen to form a candidate set. One vehicle in the set will be eventually appointed. Note that R meters is the largest sharing step size, so it takes the minimum number of relay vehicles to transmit a data unit.

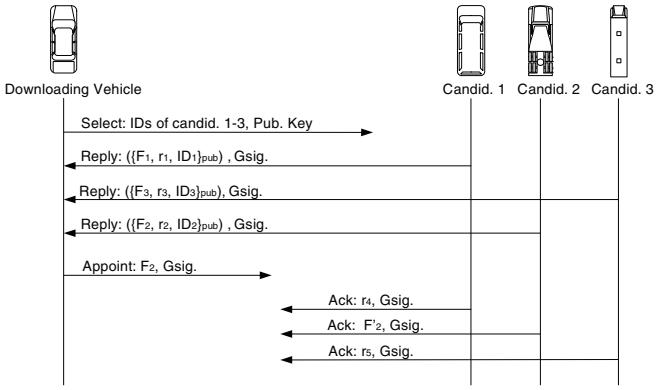


Fig. 6. Relay vehicle selection.

In the relay vehicle selection, as shown in Fig. 6, the downloading vehicle will list several (e.g. three) relay vehicle candidates in the select message and broadcast the message in the *coordination channel*. Although vehicles' privacy is under protection, identities (IDs) of vehicles can be represented by something that is unique, for example, hash values of vehicles' latest safety related messages. A public key will also be included in the request message. If a candidate collects enough packets to decode the message at time T_α, it will create a reply message. A pseudonym F_i will be generated and included in the reply message. We define $F_i = h(\hat{K}_i||r_i)$, where $\hat{K}_i$ is vehicle i's preloaded secret key and r_i is a random number. The identity (ID_i) of each candidate vehicle will also be presented in the reply message. Then, the candidate vehicle will encrypt the information by using the public key included in the request message and then sign the message by using its group private key. After this, it will choose a random time point between T_α and T_β and send the reply message to the downloading vehicle at this time point, where T_β is the estimated broadcasting finish time.

When the downloading vehicle finishes sharing, it will designate a relay vehicle among the candidate vehicles who give replies in the *coordination channel* by broadcasting an appointment message. The appointment message includes the pseudonym F_i of the selected relay vehicle and the group signature. The relay vehicle candidate will identify that it is chosen when it "sees" its F_i. Once the relay vehicle is designated, all the candidate vehicles who replied the request should broadcast an acknowledgement according to the sequence defined in the request message no matter whether they are selected. The selected relay vehicle should include $F_i' = h(F_i||r_i)$ in the acknowledgement message, while other vehicles just need to send a random number. According to the F_i', the downloading vehicle will know that the selected vehicle has received the appointment. In general, we call the downloading vehicle as *predecessor* and the selected vehicle as *successor*. If the downloading vehicle does not receive any replies when it finishes sharing, it will continue its broadcasting in the data channel for some extra time, such as 200ms. Meanwhile, it will send the select message again in the coordination channel when the *extra sending* starts. As long as a candidate gets enough packets, it will reply to the downloading vehicle.

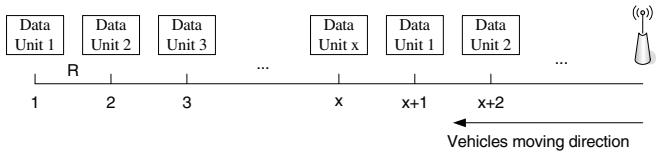


Fig. 7. Periodically data distribution.

After the downloading vehicle identifies the relay vehicle, it will abnegate the sending right by broadcasting a finish message in the *data channel*. Then, vehicles will start to send notification messages one by one in the data channel to find the next sharing vehicle. A vehicle who does not have anything to share will send a "NULL" in the data channel. Otherwise, it will send an "occupy" message. Vehicles that send neither "NULL" nor "occupy" message within Δt will be ignored.

C. Efficient Data Sharing

In the data sharing process, we focus on the case that files will be shared with all vehicles in a platoon. The number of data units (x) that vehicles can get in a drive through is defined by situations in both backward sharing and forward sharing. When we claim a vehicle can get x_b data units through backward sharing, it means that all data units (from data unit 1 to data unit x_b) will be transmitted to the (x_b+1)th downloading vehicle before the x_bth downloading vehicle reaches the next RSU. Similarly, when we claim that vehicles can download x_f data units through forward sharing, it means that all data units (from data unit 1 to data unit x_f) will be transmitted to the first downloading vehicle before it reaches the next RSU. In our framework, $x = min(x_b, x_f)$. Based on this analysis result, the authority will distribute data units with a period of x as shown in Fig. 7. Note that numbers in the figure represent downloading vehicles. For example, "1" denotes the first downloading vehicle of the platoon. In our framework, x is determined by the distance between two adjacent RSUs. It is a constant as long as the distance between two adjacent RSUs is fixed. Downloading vehicle 1 and all vehicles between downloading vehicle 1 and downloading vehicle $x+1$ form a *sharing group* in which we define downloading vehicle 1 as the *head* of the group. Repeatedly, the platoon can be divided into multiple sharing groups. Because data sharing follows the same procedure in each group, we only analyze the first one.

As we have discussed before, the distance between two adjacent downloading vehicles is around R and a downloading vehicle tends to select the vehicle which is about R meters behind it as its next relay vehicle. Thus, it is possible that downloading vehicle i appoints downloading vehicle $i+1$ as its next relay vehicle. Even if downloading vehicle i does not choose downloading vehicle $i+1$ as its next relay vehicle, the relay vehicle that it selects will not be far from downloading vehicle $i+1$. To better calculate the number of data units that vehicles can get in a drive through, we assume that downloading vehicle i will always select downloading vehicle $i+1$ as its next relay vehicle. In the next section, we use simulation to demonstrate that our analysis based on this assumption is a good approximation to the real application.

1) Backward Sharing: If a downloading vehicle is the head of a sharing group, it will start to broadcast data when it is $I + R$ far from the RSU. We term this point which is $I + R$ far from the RSU as *starting point*. Let $\hat{L}$ be the real distance between two adjacent RSUs, the effective distance L between these two adjacent RSUs is defined as

$$L = \hat{L} - 2(I + R) \tag{6}$$

This means that vehicles can share data with no collisions and interference from both RSUs for L meters. Before the downloading vehicle shares the data, it will broadcast an advertising message in the coordination channel to inform others by using the message propagation protocol. The advertising message includes the "Names & Sec. No." of the data, the vehicle's current geographic information, current time, estimated transmission time and a signature which is signed by its group private key. If the transmission time of the sharing data is more than one second, the downloading vehicle will send an advertising message every second as a heart beat. We need to emphasize that the advertising message should be disseminated far enough to prevent collisions and interferences in the concurrent transmission which will be discussed later.

After sending the advertising message, the downloading vehicle shares data with others by broadcasting data sharing message, as shown in Table I, in the data channel. In our protocol, coding methods, such as fountain code, will be employed to encode the "distribution data". When an applicant receives enough pieces of the encoded "distribution data" in the data sharing, it will decode the data and then try to decrypt them as shown in equation (2).

Then, we estimate the number of data units that vehicles can download through backward sharing. In the backward sharing, when the x_bth downloading vehicle finishes data sharing, the distance between the x_bth downloading vehicle and the starting point should be less than L. As shown in Fig. 8, after downloading vehicle 1 shares data unit 1, downloading vehicle 2 will share both data unit 1 and data unit 2, so on and so forth. Obviously, downloading vehicle i has i data units to share. Therefore, when downloading vehicle x_b finishes transmission, the total time that vehicles spend for sharing is $\sum_{i=1}^{x_b} i\delta$ seconds. During this period, the whole platoon moves $\sum_{i=1}^{x_b} iV\delta$ meters ahead. On the other hand, the distance between the first downloading vehicle and downloading vehicle x_b is $(x_b - 1)R$ meters. Thus, the max number of data units that vehicles can download (x_b) should satisfy following inequality, where x_b is an integer.

$$\sum_{i=1}^{x_b} iV\delta - (x_b - 1)R \ < \ L \tag{7}$$

The left side of the inequality shows the distance between the x_bth downloading vehicle and the starting point when the x_bth downloading vehicle finishes data sharing. Note that, as we discussed in the data distribution, although R numerically equals to $V\delta$, R represents distances between two adjacent downloading vehicles. $V\delta$ indicates the distance that a vehicle travels during a time unit (δ).

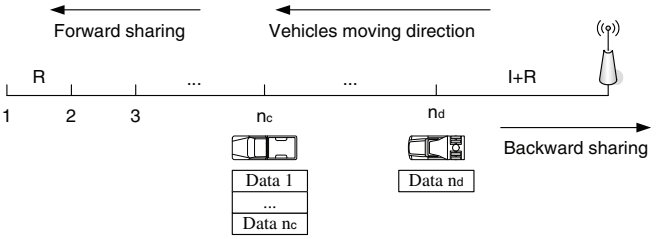


Fig. 8. Concurrent transmission.

2) Concurrent Transmission: In our framework, vehicles share data periodically. However, the head vehicle of a sharing group may not be able to start sharing when it reaches the starting point because it is not far enough from the current sending vehicle in front of it. For instance, in Fig. 7, if $x = 4$, when the head vehicle of the second sharing group (downloading vehicle 5) reaches the starting point, downloading vehicle 3 is sending. Therefore, downloading vehicle 5 can only share its data after downloading vehicle 3 and downloading vehicle 4 finish transmission. This will reduce the total sharing time of the second sharing group. In other words, the second sharing group may not have enough time to share all x data units. In this subsection, we estimate the first downloading vehicle in a platoon that can start concurrent transmission.

A downloading vehicle should satisfy two conditions if it wants to share data concurrently. First, it should be at least $(I + R)$ meters far from both the current sending vehicle and the RSU. Second, it should be able to finish transmission before the front sending vehicle. Thus, if the concurrent transmission happens when downloading vehicle n_c is sharing data, the distance between downloading vehicle n_c and the starting point should be larger than $(I+R)$ before downloading vehicle n_c starts to broadcast its last data unit. Based on the discussion in the backward sharing, as shown in Fig. 8, we can easily figure out that the distance between the n_cth downloading vehicle and the starting point can be expressed as $\sum_{i=1}^{n_c-1} iV\delta-(n_c-1)R$, when downloading vehicle n_c starts to transmit its first data unit. Therefore, the following inequality should be satisfied.

$$\sum_{i=1}^{n_c-1} iV\delta - (n_c - 1)R + (n_c - 1)R > I + R \tag{8}$$

The left side of the inequality illustrates the distance between downloading vehicle n_c and the starting point when the downloading vehicle starts to share its last data unit. The equation (8) can be reduced to

$$\sum_{i=1}^{n_c-1} iV\delta > I + R \tag{9}$$

Thus, we can illustrate downloading vehicle n_d which starts the concurrent transmission as,

$$n_d = n_c + \lceil \frac{R + I}{R} \rceil = n_c + \lceil \frac{I}{R} \rceil + 1 \tag{10}$$

In our framework, the distance between two RSUs should be large enough to let vehicles in a sharing group to download at least $n_d - 1$ data units. Namely, according to equation

(7), the distance between two adjacent RSUs should be at least $\sum_{i=1}^{n_d-1} iV\delta - (n_d-2)R$. Then, the head of the next sharing group can start the concurrent transmission as soon as it reaches the starting point. We would like to emphasize that this is not a disadvantage of our protocol because a condition to use our protocol is that the density of RSUs is low. If the density of RSUs is high, vehicles can get data from RSUs directly rather than by sharing through multi-hop communications. We term downloading vehicles who initiate a sharing as *initial downloading vehicles*, such as downloading vehicle 1 and downloading vehicle n_d. In the backward concurrent sharing, collisions and interference can be avoided automatically. As shown in Fig. 8, downloading vehicle n_d starts concurrent transmission when downloading vehicle n_c is sending. Obviously, as the time goes on, the distance between them will be enlarged if data are transmitted backward because the front downloading vehicle has more data to transmit than the rear downloading vehicle. However, in the forward concurrent sharing, data are transmitted from right to left, the distance between two sending vehicles will be shorten in the process of data sharing. Therefore, collisions happen. We will employ the authority to control the data transmission in the forward concurrent sharing.

3) Forward Sharing: Messages will be transmitted not only backward but also forward. Otherwise, the applicants at the head of the sharing group can only get a part of the data and they will apply for downloaded data again from the next RSU, which is a waste of resources. In our framework, the second data channel will be assigned for forward data sharing. The basic procedures of forward data sharing is similar with the backward data sharing. However, we implement it in a different way. Thus, collisions and interference can be avoided.

In the forward sharing, the authority will add a forward hop number and an initial bit to the "distribution data". With the forward hop number, vehicles know the number of hops that a piece of data will be sent forward. If the initial bit is "1", the downloading vehicle will start to share the data when it is $I+R$ far from the RSU. Otherwise, the downloading vehicle will wait until it is triggered. If the rear sending vehicle realizes the distance between itself and the front sending vehicle is less than a threshold in the sharing process, it will stop transmission. Meanwhile, "holding" messages will be sent in the coordination channel every second. According to the "estimated transmission time" from the front vehicle, an "estimated holding time" will be included in the holding message. When the front sending vehicle finishes sharing, a vehicle in front of it who has data to share will take over the sending right. The "holding" vehicle will resume transmission if the front sending vehicle is far enough.

Initial downloading vehicles will be chosen uniformly by the authority at the interval of ω. The value of ω is a tradeoff between performance and the probability to have collisions. For example, if a sharing group can get x_f data units through forward sharing, the x_fth, $(x_f-\omega)$th, $(x_f-2\omega)$th, *etc*, downloading vehicles in a sharing group will be selected as initial downloading vehicles.

Theorem 1. *In the forward sharing, the data sharing process will move to a stable status when downloading vehicles have ω data units to transmit and the distance between two adjacent concurrent sending vehicles is $(\omega+1)R$.*

Proof: Let T_u be the time that an initial downloading vehicle n_r starts to share its data, T_v be the time that initial downloading vehicle $n_r-\omega$ receives data units $n_r-\omega+1$, $n_r-\omega+2$, ..., n_r, and $T_v^{'}$ be the time that downloading vehicle n_r+1 receives data units n_r+2, n_r+3, ..., $n_r+\omega$. At time T_u, the initial downloading vehicle n_r starts to share the data unit n_r. It will totally take $\omega\delta$ to transmit data unit n_r to the downloading vehicle $n_r-\omega$. We can easily infer that it takes $i\delta$ to transmit data unit $n_r-\omega+i$ to the downloading vehicle $n_r-\omega$, where i = 1,2,.., ω. Therefore,

$$T_v = T_u + \sum_{i=1}^{\omega} i\delta;$$

The initial downloading vehicle $n_r+\omega$ will start the concurrent sharing $\omega\delta$ seconds after the T_u. When the initial downloading vehicle $n_r+\omega$ starts to transmit, it totally takes $\sum_{i=1}^{\omega-1} i\delta$ to share data units n_r+2, n_r+3, ..., $n_r+\omega$ to downloading vehicle n_r+1.

$$T_v^{'} = T_u + \omega\delta + \sum_{i=1}^{\omega-1} i\delta = T_v;$$

Because vehicle n_r+1 is not the initial downloading vehicle, so it also has to broadcast its own data unit n_r+1. We can see that both downloading vehicle $n_r-\omega$ and downloading vehicle n_r+1 has ω data units to share and they will start to transmit at the same time T_v. The distance between them is

$$S = (n_r+1) - (n_r-\omega) = \omega+1;$$

■

Specifically, in order to avoid collisions and interference, the following inequality should be satisfied.

$$(\omega+1)R \geq I+R; \tag{11}$$

An example. Fig. 9 is a special case of the theorem when $\omega=3$, $x_f=9$ and $n_r=6$. In this case, downloading vehicles 3, 6, 9 will be assigned as initial downloading vehicles. Two vehicles above the line demonstrate a scenario at time $T_u+3\delta$, while two vehicles below the line represent another scenario at time T_v. Downloading vehicle 6 starts to transmit at time T_u. It takes three time units for vehicles to share both data unit 6 and data unit 5 to downloading vehicle 4. Meanwhile, downloading vehicle 9 will start to transmit at this time. During the time that data unit 9 and data unit 8 are sent to location 7, data unit 4-6 will reach downloading vehicle 3. At the time T_v, both downloading vehicle 7 and downloading vehicle 3 have three data units and they will start to share data units at the same time with a distance of $4R$ between them.

In the forward sharing, we only need to focus on the x_fth data unit because it is the furthest one to the head. When the x_fth downloading vehicle starts to share, the distance between it and the first downloading vehicle is $(x_f-1)R$. The x_fth, (x_f-1)th, ..., $(x_f-\omega+2)$th downloading vehicles in the forward sharing will transmit their data units to the $(x_f-\omega+1)$th downloading vehicle by spending $\sum_{i=1}^{\omega-1} i\delta$ seconds in total.

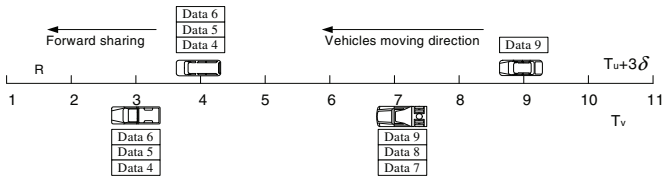


Fig. 9. Forward data sharing.

All the rest downloading vehicles, except these first $\omega - 1$ downloading vehicles, have ω data units to share. The total number of hops from the x_fth downloading vehicle to the first downloading vehicle is $x_f - 1$. Thus, the number of groups that have ω data units to share is $\tau = x_f - 1 - (\omega - 1) = x_f - \omega$. When these vehicles send data to the first downloading vehicle, the platoon will move forward $\tau\omega V\delta$. Therefore, the number of data units vehicles can get through forward sharing is

$$(x_f - 1)R + \sum_{i=1}^{\omega-1} iV\delta + \tau\omega V\delta < L \tag{12}$$

By solving equations (7) and (12), we can get the values of x_b, x_f and $x = min(x_b, x_f)$.

We summarize the data sharing procedure in this paragraph. In the backward sharing procedure, after vehicles download data from an RSU, they will constantly check the distance between them and the RSU. When a vehicle reaches the starting point and it is the head of a sharing group, the vehicle will start to share data. Otherwise, it will wait for the sending right from others. First of all, the head vehicle will send an advertising message in the coordination channel by employing message propagation method. Then, it starts to share data in a data channel. Before the head vehicle finishes data sharing, it will utilize relay vehicle selection method and GPS information to identify and appoint the next relay vehicle. Finally, the head vehicle will abnegate sending right to the next relay vehicle. The forward sharing procedure has a similar process. Due to the limitation of space, we omit details here.

VI. Security and Privacy Performance

A. Exclusive Data Access

Our framework guarantees applicants' exclusive access to the data that they apply. Before the authority disseminates data, it will encrypt the data by using a session key K_d derived from the broadcast encryption technique, as shown in equation (1). The encryption ensures only applicants who pay for the service are able to get the key K_d. In other words, the applicants have exclusive access to the data.

B. Privacy Provision

Our framework allows vehicles to download data with their privacy under protection. Eavesdroppers are not able to link any two messages sent by the same vehicle. Applicants only need to prove their legitimacy to RSUs and the authority in the request phase. In this procedure, applicants' privacy is guaranteed by the short group signature protocol. After the request phase, applicants do not need to send any messages. Therefore, it is impossible for eavesdroppers to compromise their privacy. In the data distribution phase, the authority attempts to define downloading vehicles according to their positions. Downloading vehicles know it is selected from the index $\phi = h(\hat{K}_i \mid\mid T)$. Because only vehicles themselves have their secret keys, eavesdroppers can not know who is the downloading vehicle. In the data sharing phase, relay vehicles are chosen randomly from candidate sets. After the data sharing, similar to downloading vehicles' assignment, random index F_i and F_i' are used to appoint relay vehicles. Later, when the next relay vehicle starts to share a certain data unit, eavesdroppers can only be sure that this relay vehicle was a member in its predecessor's the candidate set. However, they can not exactly identify which member it was. We would like to emphasize that in the relay vehicle selection process, except the downloading vehicle and the selected relay vehicle, even vehicles in the candidate set do not know who is selected. All they know is they are not selected. Thus, in our framework, it is impossible for eavesdroppers to link any two messages from the same vehicle.

TABLE III
Simulation Parameter Settings

Parameter	Value
Forward initial downloading vehicles (ω)	3
Interference range(I)	$3R$
Communication range(R)	270m
System communication rate	6Mbps (QPSK)
Propagation mode	Nakagami m = 3
Decoding inefficiency(1+ϵ)	1.1

VII. Simulation Results

In this section, we use NS-2.34 to evaluate the performance of the proposed cooperate data downloading protocol. We examine the number of data units that a vehicle can download in a drive through with different effective distances between RSUs. Comparisons between our protocol and an existing cooperative downloading protocol "VC-MAC" [4] will also be given. After that, the time that vehicles spend to download all the data and real time throughput will be exhibited. Finally, we evaluate the throughput of our protocol when multiple files are requested. Moreover, Comparisons between our protocol and a simple sharing protocol are presented.

In order to produce realistic simulations, modified PHY and MAC [32] which provide cumulative signal to noise plus interference ratio, header and frame body capture, structured and modular MAC procedures are employed. We consider a highway scenario in which vehicles enter the highway according to a Poisson distribution with traveling speeds of 30 $\pm$ 5m/s (roughly equivalent to the range of 56 $\sim$ 80 miles/hour). The traffic density in our simulation is 100 vehicles per kilometer (50 vehicles per kilometer on each side). Given a certain transmission power, we define communication range as the range up to which 95 percent of packets can be correctly received. Some parameters that will be used in our simulations are listed in Table III.

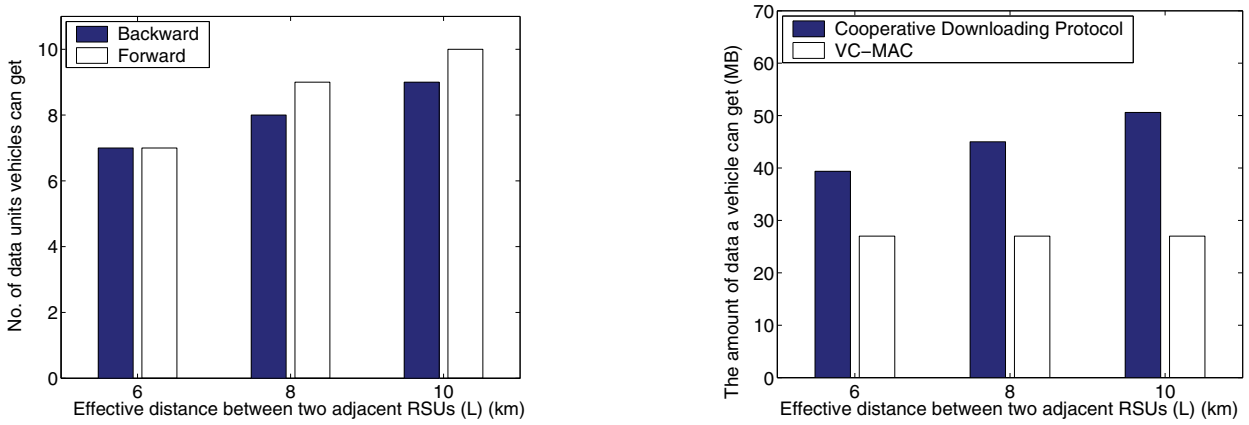


Fig. 10. Downloaded data vs. distances between RSUs. (a) Cooperative downloading protocol. (b) Comparison between the cooperative downloading protocol and the VC-MAC.

A. Data Volume Through Cooperative Downloading

We employ coding methods to counteract the packet loss in our protocol. Considering both decoding inefficiency and packet loss, in our simulation, RSUs and vehicles send 20 percent extra packets in the data distribution and sharing procedure. For example, if the original data unit have k symbols, the total number of symbols that RSUs and vehicles broadcast will be $1.2k$. If a receiver can get $1.1k$ encoded symbols, we consider that it could decode the data unit successfully.

The number of data units that vehicles can get in a drive through directly influences the downloading efficiency. As shown in Table III, when the interference range I equals to $3R$ and ω equals to 3, we can find that inequality (11) is satisfied. According to equations (9) and (10), downloading vehicle 8 will start the backward concurrent transmission. According to equation (7), the effective distance between two adjacent RSUs should be at least 6km. Given a certain L, we can get the theoretical values of x_b and x_f by employing equations (7) and (12). Due to the limitation of the space, we ignore the detailed calculation procedures. The number of data units that vehicles can download in simulations are presented in Fig. 10(a). The simulation results match our theoretical calculations except in the forward sharing when L is 10km. The mismatch is caused by the marginal effect which can be avoided in the real application if some redundancies are reserved.

Our relay vehicle selection protocol can facilitate data sharing. In fact, applicants can collect encoded packets for a certain data unit from both front and back relay vehicles. The critical issue is whether relay vehicles can get enough encoded packets from the previous relay vehicle because there is no acknowledgement in the broadcast communication. Moreover, the previous relay vehicle is the only source for them to collect encoded packets. If relay vehicles can not get enough packets to decode, the sharing chain will be broken. For example, as illustrated in Fig. 2, if vehicle 10 is an applicant and vehicle 11 appoints vehicle 9 its next relay vehicle, vehicle 10 can collect encoded packets of data unit 1 through both vehicle 11's and vehicle 9's broadcasting. However, as the next relay vehicle, vehicle 9 can only obtain encoded packets of data unit 1 through vehicle 11's broadcasting. If it can not get enough packets, no one can continue to share data unit 1. In our relay vehicle selection protocol, several relay vehicle candidates are selected to guarantee the relay vehicle selection. Meanwhile, "extra" packets sending mechanism is a good supplementary in case no relay vehicle candidates can get enough packets.

A cooperative MAC protocol [4] is proposed to augment data downloading. In the protocol, RSUs periodically broadcast packets with the maximum cycle of $2R/V$, where R is RSUs' communication range and V is the platoon velocity. Vehicles receive these packets when they pass by. However, due to the unavoidable wireless packet loss, vehicles will miss some data. Therefore, vehicles complete the data downloading by sharing data with each other when they travel out of RSUs' communication range. In the VC-MAC, the best case is that vehicles get all the packets through sharing. The amount of data that a vehicle can download from one channel is 6Mbps $\times$ 2R/V. Therefore, in their protocol, vehicles can download 27 megabytes if they utilize two channels.

However, in our protocol, vehicles can download different data units when they pass by an RSU. And then, share these data units when they travel out of the RSU's communication range. In this way, the total data that a vehicle can download in a drive through will not be bounded by the duration it has direct connection with the RSU. In our protocol, the total amount of data that vehicles can download in a drive through is

$$Total\ Data = \frac{6R}{V}\ x\ \frac{1}{8 \times 1.2}\ (MB)$$

A comparison between our protocol and "VC-MAC" is shown in Fig. 10(b). From the figure, we can see that the amount of data our protocol can download increases when the effective distance between RSUs increases. In the scenario that the effective distance between RSUs reaches 10 kilometers, vehicles using our protocol can get 87.4% more data than vehicles employing "VC-MAC".

Another advantage of our protocol is that it only uses one channel when RSUs distribute data to vehicles although two channels are occupied for data sharing. In other words, our protocol only needs to employ the second channel in the area out of RSUs' interference range. Thus, some applications

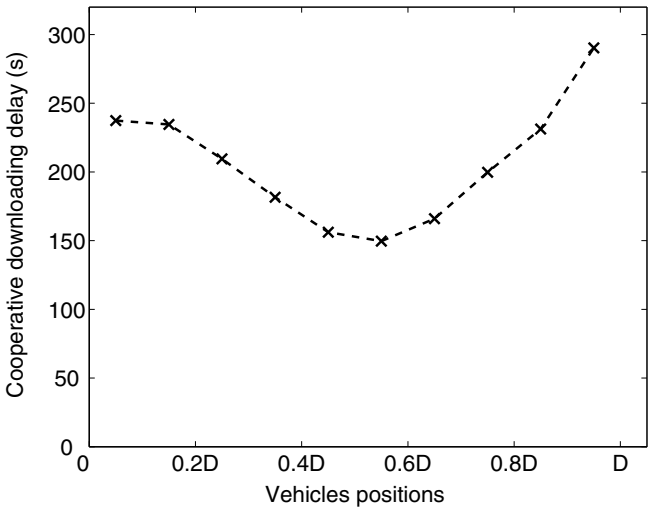


Fig. 11. The time to finish data downloading.

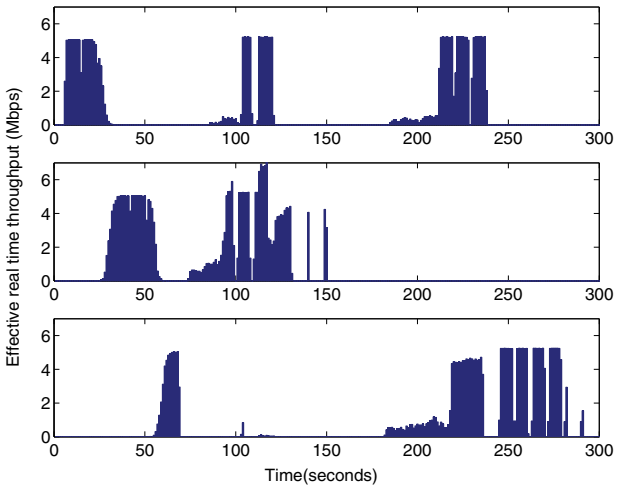


Fig. 12. Realtime throughput.

which will not utilize multi-hop communications can be implemented in the second channel when vehicles pass by RSUs.

B. Downloading Delay and Real Time Throughput

We study the downloading delay and the real time throughput of our protocol in the scenario that the effective distance between RSUs is 6km. In this setting, there are approximately 95 vehicles in a sharing group.

1) Cooperative Downloading Delay: The time point that the first vehicle enters the RSU's communication range is termed as zero second (T_0). If a certain vehicle downloads all the data units at time T_{finish}, the cooperative downloading delay for this vehicle is $T_{finish} - T_0$. Fig. 11 illustrates the cooperative downloading delay for vehicles at different parts of a sharing group where D is the length of a sharing group.

From the figure, we can see that vehicles in the middle area of the sharing group collect all seven data units earlier than vehicles at two sides. It is reasonable because backward sharing and forward sharing data meet in the middle. The simulation results show that the quickest vehicle get all data units at about 150th second. However, vehicles at two sides finish downloading much later.

2) Real Time Throughput: The effective throughput is vehicle's real throughput minus the fountain codes' overhead. Quantitatively, the effective throughput equals to vehicles' throughput divided by the decoding inefficiency. We choose vehicles from the front, middle and rear of the sharing group and plot their effective realtime throughputs at the top, middle and bottom in Fig. 12 respectively.

First of all, we compare the cooperative downloading delay in Fig. 11 and Fig. 12 which shows they agree with each other. Then, we observe these figures in details. In our protocol, RSUs distribute data to downloading vehicles through broadcasting. Therefore, all vehicles locate within RSUs' communication range will benefit from it. As illustrated in Fig. 12, the first data block in each plot is the amount of data that vehicles can get from the RSU. The middle vehicle obviously can download more data from the RSU than the other two vehicles. This is due to the longer contact time it has. Because the authority prefers to utilize the area with higher signal to interference plus noise ratio, the front vehicle starts to receive data when it is located at $-R/2$, as shown in Fig. 3. Therefore, its throughput suddenly increases to the maximum value at a certain time and then decreases to zero gradually when it leaves the RSU. Meanwhile, the rear vehicle has no data to receive from the RSU after it reaches the $R/2$ point. This is the reason why its throughput goes to zero rapidly at about the 70th second.

In the figure, although two vehicles receive data at the same time, their throughputs may be caused by different data units. For example, between 200th second and the 225th second, both front vehicle and the rear vehicle receive data. However, our trace shows that, during this period, the front vehicle is receiving data units 5, 6 and 7 through forward sharing. And the rear vehicle is receiving data units 3, 4 and 5 through backward sharing. Moreover, we can also see that data downloading is burst based in our protocol. It only ensures vehicles to get all the data units before they reach the next RSU. The data downloading sequence can not be guaranteed.

An interesting observation of our simulation is that at about 120th second, the middle vehicle's effective throughput reaches 7Mbps. This is because backward sharing data and forward sharing data meet in the middle. At a certain time, the vehicle can receive fresh encoded data through both backward sharing and forward sharing. Our trace shows that the receiver locates between downloading vehicle 4 and 5. At this time, both downloading vehicle 4 and 6 are sending. The receiver is more than one hop away from a sender. This is the reason why the effective throughput only reaches 7Mbps.

C. Throughput with Multiple Files

In this subsection, we simulate the scenario that multiple files are requested by vehicles. The effective distance between two RSUs is 6km. Within the stream of vehicles, 25 applicants are randomly generated and each applicant randomly requests one data file from a set of 7 candidate files. Moreover, we compare the cooperative downloading protocol with a simple

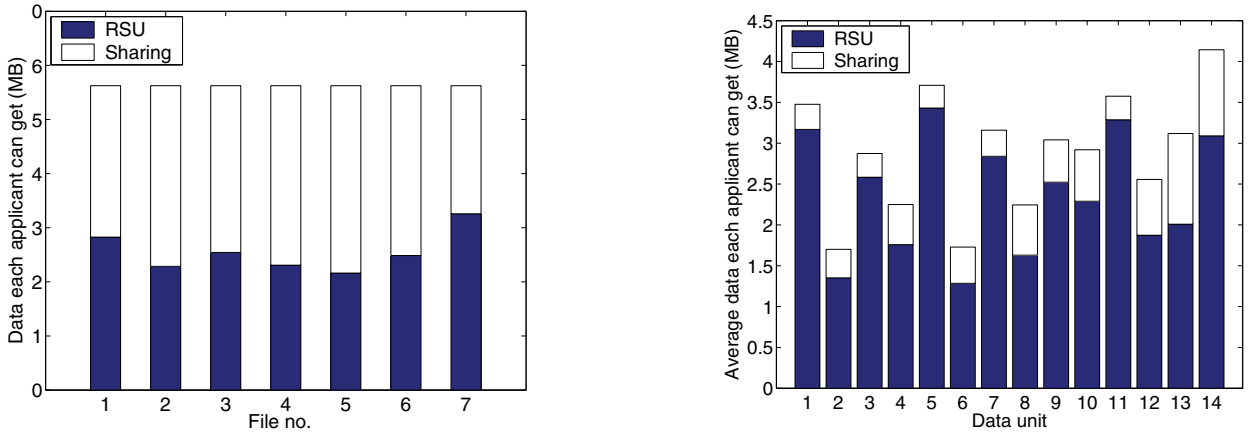


Fig. 13. Amount of downloaded data (multiple files) (a) Cooperative downloading protocol. (b) Simple sharing protocol.

sharing protocol (SSP). In the SSP, when a stream of vehicles pass an RSU, vehicles will send their request messages to the RSU if any. After receiving request messages, the RSU will distribute a data unit for each requested file first without using any coding techniques. If the RSU receives multiple requests at the same time, it will send data units based on a first come first serve policy. When the RSU has extra capacity, it will continue to send requested files in a round-robin manner. Vehicles acquire some data which are requested either by themselves or by other vehicles when they pass by the RSU. When vehicles move out of the RSU coverage area, they will start to share all the received packets through the default CSMA/CA MAC protocol until they reach the next RSU. Note that, in the simple sharing protocol, two 10MHz data channels will be combined to a 20MHz channel with a communication rate of 12Mbps.

Under the cooperative downloading protocol, the average amount of data that applicants of each file can get is shown in Fig. 13(a). First, we compare the results in Fig. 13(a) with that in Fig. 10(b). The total amount of data of seven downloaded files equals to the total amount of downloaded data in Fig. 10(b). In the simulation, our protocol can guarantee that all vehicles in the sharing group can get their requested data distributed by the RSU. Moreover, from Fig. 13(a), we can also see that our sharing protocol is efficient. Applicants in our cooperative downloading protocol can get approximately 50% data through sharing. Last but not least, our protocol ensures a fair data distribution to each file.

In the SSP, because the RSU uses a 20MHz channel to distribute data, the total amount of data that the RSU can disseminate will be two times of that in our cooperative downloading protocol. As shown in Fig. 13(b), data units 1 and 2 are data for file 1. Data units 3 and 4 are data for file 2 ..., and so on. Obviously, the sharing process in the SSP is less efficient than that in our protocol. In the SSP, compare with our protocol shown in Fig. 13(a), the amount of data that vehicles get through sharing is much smaller.

Fig. 14 demonstrates the receipt of each data unit through SSP. In Fig. 14, each data unit is associated with three bars indicating the proportion of packets that are received by none of the applicants, some of the applicants, and all the applicants,

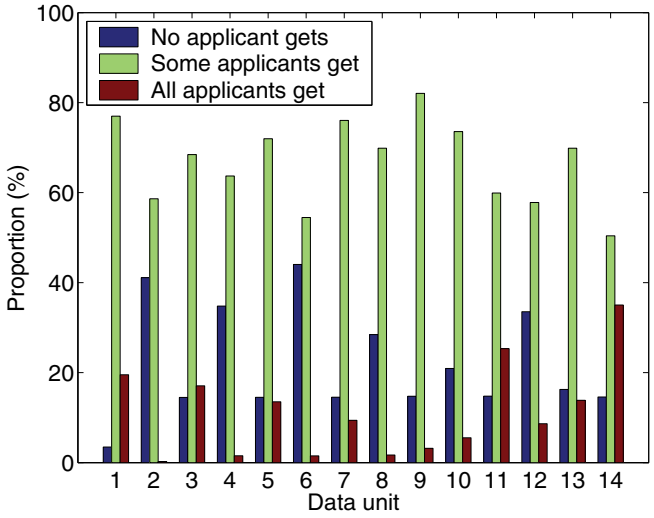


Fig. 14. Proportion of successful downloading of each data unit.

respectively. We can observe that every data unit can only be partially received by its applicants. Because there is no guaranteed data reception in the SSP, the authority has to distribute the same data units multiple times if it wants to ensure all the applicants get their requested data. Thus, the overall performance would be poor.

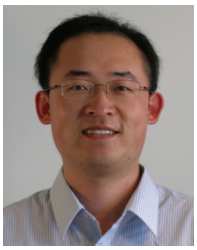
VIII. Conclusions

In this paper, we propose a secure cooperative data downloading framework for paid services in VANETs. Security and privacy protocol as well as a location based sharing protocol are proposed to secure and improve the efficiency of our framework. Extensive NS2 simulations are presented to evaluate the performance of our protocols. In the future work, we will research on RSU scheduling protocols and low vehicle density scenarios.

References

[1] T. H. Luan, X. Lin, and X. Shen, "MAC in motion: Impact of mobility on the MAC of drive-thru internet," *IEEE Trans. Mobile Comput.*, vol. 11, no. 2, pp. 305–319, 2012.

[2] A. Nandan, S. Das, and M. Gerla, "Cooperative downloading in vehicular ad-hoc wireless networks," in *Proc. WONS*, Jan. 2005.
[3] S. Ahmed and S. S. Kanhere, "VANETCODE: Network coding to enhance cooperative downloading in vehicular ad hoc networks," in *Proc. IWCMC*, 2006.
[4] J. Zhang, Q. Zhang, and W. Jia, "VC-MAC: A cooperative MAC protocol in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 3, pp. 1561–1571, 2009.
[5] X. Lin, X. Sun, P.-H. Ho, and X. Shen, "GSIS: A secure and privacy preserving protocol for vehicular communications," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3442–3456, 2007.
[6] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Proc. Crypto, LNCS 3621*, 2005, pp. 258-275.
[7] A. Studer, E. Shi, F. Bai, and A. Perrig, "TACKing together efficient authentication, revocation, and privacy in VANETs," in *Proc. IEEE SECON*, 2009.
[8] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A media access protocol for wireless LANs," in *Proc. ACM SIGCOMM*, 1994, pp. 212–225.
[9] S. Tabbane and p. Godlewski "Performance evaluation of the RBTMA protocol in a distributed mobile radio network context," *IEEE Trans. Veh. Technol.*, vol. 41, no. 1, pp. 24–34, Feb. 1992.
[10] S. Makido, N. Suzuki, T. Harada, and J. Muramatsu "Decentralized TDMA protocol for real-time vehicle-to-vehicle communications," *IPSJ J.*, vol. 48, no. 7, pp. 2257–2266. July 2007.
[11] S. Yuan, C. Zhang, and P.-H. Ho, "A secure business framework for file purchasing in vehicular networks," *Security Commun. Netw.*, vol. 1, no. 3, pp. 259–268, 2008.
[12] K. E. Shin, H. K. Choi, and J. Jeong, "A practical security framework for a VANET-based entertainment service," in *Proc. ACM PM2HW2N*, 2009.
[13] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *J. Computer Security*, vol. 15, no. 1, pp. 39–68, 2007.
[14] K. Sampigethava, M. Li, L. Huang, and R. Poovendran, "AMOEBA: Robust location privacy scheme for VANET," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp. 1569–1589, 2007.
[15] Y. Hao, J. Tang, Y. Cheng, and C. Zhou, "Secure data downloading with privacy preservation in vehicular ad hoc networks," in *Proc. IEEE ICC*, May 2010.
[16] Y. Hao, Y. Cheng, C. Zhou, and W. Song, "A distributed key management framework with cooperative message authentication in VANETs," *IEEE J. Sel. Areas Commun.*, vol. 29 , no. 3, pp. 616–629, 2011.
[17] J. Jeong, S. Guo, Y. Gu, T. He, and D. Du, "Trajectory-based statistical forwarding for multi-hop infrastructure-to-vehicle data delivery," *IEEE Trans. Mobile Comput.*, Aug. 2011.
[18] J. Huang and H. S. Tan, "A low-order DGPS-based vehicle positioning system under urban environment," *IEEE Trans. Mechatron.*, vol. 11, no. 5, pp. 567–575, Oct. 2006.
[19] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, and O. Tonguz, "Routing in sparse vehicular ad hoc wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp. 1538–1556, 2007.
[20] "IEEE 1609.3-2007 WAVE networking service," 2007.
[21] "FCC report and order 06-110: Amendment of the commission's rules regarding dedicated short-range communication services in the 5.850-5.925 GHz band," July 2006.
[22] S. M. Abd El-atty, "Vehicular communications framework for efficient multihop connectivity in AHVN," in *Proc. IEEE VTC*, 2011.
[23] S. Kaul, K. Ramachandran, P. Shankar, S. Oh, M. Gruteser, I. Seskar, and T. Nadeem, "Effect of antenna placement and diversity on vehicular network communications," in *Proc. IEEE SECON*, 2007.
[24] M. Alicherry, R. Bhatia, and L. Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks," in *Proc. ACM MobiCom*, Aug. 2005, pp. 58–72.
[25] H. Li, Y. Cheng, C. Zhou, and P. Wan, "Multi-dimensional conflict graph based computing for optimal capacity in MR-MC wireless networks," in *Proc. IEEE ICDCS*, June 2010.
[26] Y. Cheng, H. Li, and P. Wan, "A theoretical framework for optimal cooperative networking in multi-radio multi-channel wireless networks," *IEEE Wireless Commun. Mag.*, vol. 19, no. 2, pp. 66–73, 2012.
[27] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.
[28] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," in *Proc. ACM SIGCOMM*, 1998, pp. 56–678.
[29] M. Li, Z. Yang, and W. Lou, "CodeOn: Cooperative popular content distribution for vehicular networks using symbol level network coding," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 1, pp. 223–235, 2011.
[30] F. Ye, S. Roy, and H. Wang, "Efficient data dissemination in vehicular ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 4, pp. 769–779, May 2012.
[31] S. Lee, G. Pan, J. Park, M. Gerla, and S. Lu. "Secure incentives for commercial ad dissemination in vehicular networks," in *Proc. ACM MobiHoc*, 2007.
[32] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi, and H. Hartenstein, "Overhaul of IEEE 802.11 modeling and simulation in NS-2," in *Proc. ACM MSWiM*, 2007.



Yong Hao received the B.E. and M.E. degrees in electrical engineering from Huazhong University of Science and Technology, Wuhan, Hubei, China, in 2003 and 2007, respectively, and the Ph.D. degree in computer engineering from the Illinois Institute of Technology, Chicago, IL, USA, in 2012. He is now with Juniper Networks. His current research interests include wireless networking, network security, and vehicular ad hoc networks.

Jin Tang (S'10-M'13) received the B.S. degree in computer science from Fudan University, Shanghai, China, in 2004, the master's degree in information technology and management from the Illinois Institute of Technology, USA, in 2007, and the Ph.D. degree in computer engineering from the Illinois Institute of Technology, USA, in 2012. He is now with AT&T Labs. His research interests include wireless network security, intrusion detection, and security in VoIP applications. He received a Best Paper Award from IEEE ICC 2011.

Yu Cheng (S'01-M'04-SM'09) received the B.E. and M.E. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1995 and 1998, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, Ontario, Canada, in 2003. From September 2004 to July 2006, he was a postdoctoral research fellow in the Department of Electrical and Computer Engineering, University of Toronto, Ontario, Canada. Since August 2006, he has been with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, Illinois, USA, and is now an Associate Professor. His research interests include next-generation Internet architectures and management, wireless network performance analysis, network security, and wireless/wireline interworking. He received a Postdoctoral Fellowship Award from the Natural Sciences and Engineering Research Council of Canada (NSERC) in 2004, and a Best Paper Award from the conferences QShine 2007 and ICC 2011. He received the National Science Foundation (NSF) CAREER Award in 2011. He served as Co-Chair for the Wireless Networking Symposium of IEEE ICC 2009, Co-Chair for the Communications QoS, Reliability, and Modeling Symposium of IEEE GLOBECOM 2011, Co-Chair for the Signal Processing for Communications Symposium of IEEE ICC 2012, and as Technical Program Committee (TPC) Co-Chair for WASA 2011. He is an Associated Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and the new book *Multimedia Column Editor* for IEEE NETWORK.