sensors

MDPI

Article

A Cooperative Downloading Method for VANET Using Distributed Fountain Code

Jianhang Liu [1,*], Wenbin Zhang [1], Qi Wang [2], Shibao Li [1], Haihua Chen [1], Xuerong Cui [1] and Yi Sun [2]

[1] College of Computer and Communication Engineering, China University of Petroleum, Qingdao 266555, China; audiproduct@gmail.com (W.Z.); lishibao@upc.edu.cn (S.L.); chenhaihua@upc.edu.cn (H.C.); cuixuerong@upc.edu.cn (X.C.)

[2] Institute of Computing Technology Chinese Academy of Sciences, Beijing 010190, China; wangqi08@ict.ac.cn (Q.W.); sunyi@ict.ac.cn (Y.S.)

* Correspondence: liujianhang@upc.edu.cn

Academic Editor: Paolo Bellavista
Received: 30 May 2016; Accepted: 26 September 2016; Published: 12 October 2016

Abstract: Cooperative downloading is one of the effective methods to improve the amount of downloaded data in vehicular ad hoc networking (VANET). However, the poor channel quality and short encounter time bring about a high packet loss rate, which decreases transmission efficiency and fails to satisfy the requirement of high quality of service (QoS) for some applications. Digital fountain code (DFC) can be utilized in the field of wireless communication to increase transmission efficiency. For cooperative forwarding, however, processing delay from frequent coding and decoding as well as single feedback mechanism using DFC cannot adapt to the environment of VANET. In this paper, a cooperative downloading method for VANET using concatenated DFC is proposed to solve the problems above. The source vehicle and cooperative vehicles encodes the raw data using hierarchical fountain code before they send to the client directly or indirectly. Although some packets may be lost, the client can recover the raw data, so long as it receives enough encoded packets. The method avoids data retransmission due to packet loss. Furthermore, the concatenated feedback mechanism in the method reduces the transmission delay effectively. Simulation results indicate the benefits of the proposed scheme in terms of increasing amount of downloaded data and data receiving rate.

Keywords: cooperative downloading; digital fountain code; VANET

1. Introduction

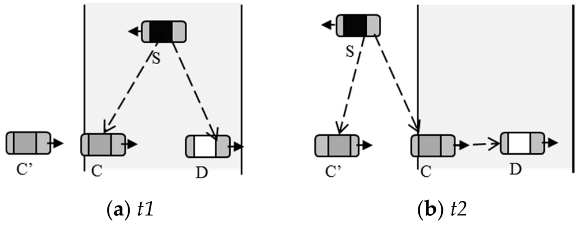
With the development of wireless communication technology, there is an increasing demand to access the Internet for commuters on vehicles. The goal can be achieved using widely available cellular systems. In some places such as highway scenarios, however, base stations are built with sparse distribution (the two BS are 2.5 km apart in highway scenarios and 400m apart in urban areas), which leads to poor performance. Furthermore, the high cost of a cellular system is another barrier to restrict the development of accessing Internet on vehicles. Wi-Fi is another technology that can be employed by VANET. The disadvantage of Wi-Fi access is the constrained communication area.

Cooperative downloading is an effective way to extend the communication area and improve the amount of downloaded data in VANET. When a client travels out of a hotspot, it can download data from encountering vehicles which carry the data it needs. The method contributes to data sharing and an improvement of the amount of downloaded data. However, due to the poor channel quality and the changing vehicle speeds frequently, the data carried by cooperative vehicles are always lost and cannot be totally transmitted to the client. The problems fail to satisfy the requirement of some applications with high quality of service (QoS). The delay is quite unacceptable when the client reports the loss data to the next hotspot. Although some compensation methods [1] can improve the downloading proportion, frequent data relay occupies the limited bandwidth of VANET.

Digital fountain code (DFC) [2], developed by John Byers, can be applied for the field of wireless transmission to improve channel utilization and transmission efficiency. Fountain code is a kind of rateless code. This means that a potentially limitless number of encoded packets can be generated from the information source. Assuming that the sender encodes the k raw data with DFC and generates encoded packets continuously, the receiver will recover the raw data successfully as long as it receives any subcollection of $k(1+\varepsilon)$ encoded packets. Not only a small decoding overhead ε but also DFC has a simple decoding method and low complexity. The biggest difference between DFC and LDPC (Low Density Parity Check Code) is that DFC has no code length. In other words, the code length multiplies towards infinity. However, DFC does not perfectly adapt to DTN (delay tolerant network) because the communication time between the sender and the client is very short, which results in not collecting enough packets to recover the raw data.

DFC is a technology of error control working on a data link layer. It is appropriate for an erasure channel. In the field of cooperative communication, the source and the destination nodes can match channel capacity self-adaptively by aid of DFC, thus improving transmission efficiency and reliability [3]. However, traditional DFC cannot adapt to the environment of VANET because the nodes move rapidly and the topology changes frequently. This is attributed to two main reasons: (1) Every relay node has to encode and decode data when receiving and sending packets. Obviously, the process greatly increases the burden of the relay nodes and the total computing complexity; (2) When receiving the data, the relay node needs to feedback to the forward node, which decides whether it continues to send the packets or not. The process is inappropriate for VANET because the communicating time among vehicles is so short that the source might leave the communication area of the client when it receives enough data. Furthermore, the mechanism of the cascade feedback brings about a long transmission delay.

To solve the problems above, the paper proposes a cooperative downloading method based on a two-layer distributed DFC. The method makes use of DFC and the mechanism of cooperative downloading to decrease the packet loss rate and to increase communication time, respectively. Figure 1 shows the basic idea of the scheme. The vehicle S encodes the raw data into LT-encoded packets before it sends to the client D when encountering. During the process of transmitting, another vehicle, C or C', running in the communication area of S, receives the packets (Figure 1a). The C or C' encodes the packets and sends to D when S leaves the communication area of D (Figure 1b). Although some packets are likely to be lost, D can restore the raw data once it receives enough packets from S and C.



(**a**) *t1* (**b**) *t2*

Figure 1. Cooperative downloading using digital fountain code (DFC). (**a**) S sends data to D directly; (**b**) C relays data from S to D when S leaves the communication area of D.

The difference between this and other studies is that the cooperative vehicles are not employed to relay original data packets but to transmit DFC encoded packets. The client does not need to receive every packet sent by the source to recover the original data. In other words, packet loss is accepted. The characteristic adapts to VANET with a high packet loss rate. Furthermore, it is possible that the vehicle carrying the raw data leaves the communication area of the client without finishing transmission. As a result, the client cannot send feedback to the senders. Therefore, a distributed fountain code is proposed in the method to solve this problem. The cooperative vehicles do not decode the packets received from the source vehicles, but encode the encoded packets secondly using the distributed function before sending to the client. The method can increase the amount of downloaded data greatly.

2. Related Work

Cooperative downloading in vehicular networks was first introduced by Nandan et al. [4] as a part of the protocol-SPAWN for cooperative content retrieval and sharing among users aboard vehicles. MobTorrent [5] improved the amount of downloaded data. The study ignored the packet loss rate and the client had to obtain the loss packets in the next hotspot. Dongyao Jia et al. [6] proposed a retransmission scheme to reduce the packet loss rate at the cost of more retransmission time, which brought about low throughput. Yuchen Wu et al. [7] proposed to exploit trajectory prediction to improve data delivery. However, they did not discuss the solution for the loss of the packets. A compensation method was proposed in [8]. In this method, the loss packets were relayed in cooperative vehicles in order to reduce the packet loss rate. Furthermore, Liu [9] proposed the strategy of data uploading in a VANET-based mobile cloud service for enlarging the communication area and reducing the delay of remote transmission.

DFC is an ideal solution for large-scale data distribution and reliable broadcast. Some studies [10,11] have adopted independent digital fountain code to guarantee the reliability of each hop in the two hops. However, every relay node had to encode and decode data and sent an acknowledge (ACK) to the source node. Obviously, the method brought more computation complicity. Furthermore, frequent information feedback causes a longer delay. In order to solve the problem, cascade coding was proposed in [12]. The relay node did not decode the packets, but encoded the packets secondly after receiving from the source node. The problem was that the cooperative transmission based on the cascade encoding brought about more decoding complicity in the client. Meanwhile, Rui Cao et al. [13] proposed DLT (decomposed LT codes). The two links (from the source to the client and from the relay to the client) in DLT employed the same DFC to keep reliability of communication and reduced computation complicity and delay. However, the model ignored the packet loss rate among the direct link and the other links. It only gave the value span of the distribution function in the first layer, and it was difficult to fulfill. Shabbier Ahmed proposed VANETCODE [14], which divided data into packets before encoding them. Every vehicle passing AP (Access Point) downloads serval packets. They exchange the packets after leaving the communication area of AP. The method has high efficiency when the same data is requested by different users. However, it is not applicable for the requirement that different users hope to obtain different data. Similarly, Hao et al. [15] developed an application layer data sharing protocol that 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. Distributed-fountain network code (DFNC), which has low encoding, re-encoding, and decoding complexity, is proposed in [16]. However, the decoding process in intermediate vehicles will increase transmission delay.

3. The Method of Cooperative Downloading

3.1. The Strategy of Cooperative Downloading

To illustrate the method clearly, we assume that only one client requests cooperative downloading and that APs can communicate with each other via the Internet. According to the request of the client, APs choose a group of vehicles to provide cooperative downloading services. Each vehicle was equipped with a Wi-Fi interface and a DSRC protocol stack. Depending on running direction, the vehicles were divided into two groups: one moving with the same direction as the client (represented by $\overrightarrow{v}$) and the other moving in the opposite direction (represented by $\overleftarrow{v}$). When the client left an AP coverage area without finishing its downloading, the next AP chose vehicles in $\{\overleftarrow{v}\}$ that can catch up with the client and download the remaining packets to it. The principles of choosing cooperative vehicles included no overlapping collision area and a pursuit of maximum throughput.

As shown in Figure 2, the vehicles entered and left the AP communication area at different times as well as encountered the client in the dark area (DA) at different times and areas. Therefore, AP could choose a group vehicles to relay data to the client in the DA, as shown at t1–t4 in Figure 2. Due to the limited communication area of the vehicles (about 300 m using DSRC), the amount of data the cooperative vehicles carried was decided by the encountering time with the client. According to previous research [17], traffic conforms to Poisson distribution. When λ = 5, up to 20% vehicles can act as the helpers. Transmission collision will take place if more than 20% vehicles provide cooperative downloading services. Therefore, which vehicles are selected to provide cooperative downloading services is closely related to system throughput. According to the characteristic of VANET, we propose a cooperative downloading strategy based on dynamic slots. In this strategy, the DA is divided into several slots according to the encountering time between the vehicles and the client.

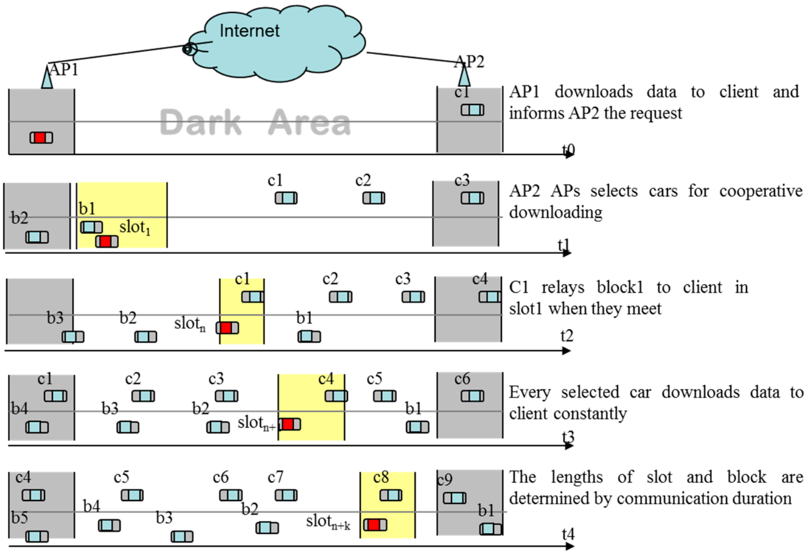

Internet
AP1
AP2
Dark Area
c1
t0
AP1 downloads data to client and informs AP2 the request
b2
b1
slot$_1$
c1
c2
c3
t1
AP2 APs selects cars for cooperative downloading
b3
b2
slot$_n$
c1
c2
c3
c4
b1
t2
C1 relays block1 to client in slot1 when they meet
c1
b4
c2
b3
c3
b2
slot$_{n+}$
c4
c5
b1
c6
t3
Every selected car downloads data to client constantly
c4
b5
c5
b4
b3
c6
b2
c7
slot$_{n+k}$
c8
c9
b1
t4
The lengths of slot and block are determined by communication duration

Figure 2. Cooperative downloading based on dynamic slots.

While entering an AP coverage area, the vehicle n registers its ID, the speed v_n, and the present time t_n. Every AP maintains a list including all vehicles in its coverage area, which is represented by $List = \{(id_0, v_0, t_0), \ldots, (id_n, v_n, t_n)\}$. The list changes along with vehicles entering or leaving the communication area of the AP. *AP1* and *AP2* exchange their *Lists* every 30 s and the relevant items will be removed when an AP finds that the vehicle with a newer t is in the list of other APs.

When *AP1* receives a downloading request from a client, it searches for the content from the Internet before it downloads the data to the client. The size of the downloading data is determined by its running time in the communication area, which can be calculated according to the information in the *List*. If the downloading is unfinished, *AP1* informs *AP2* of cooperative downloading.

When *AP2* receives the task, it picks out a group of vehicles that will meet the client in different time slots of the DA and put them in the collection $M = \{(id_0, v_0, t_0, B_0, E_0, T_0), \ldots, (id_n, v_n, t_n, B_n, E_n, T_n)\}$ in items of meeting time. $(id_n, v_n, t_n, B_n, E_n, T_n)$ represent the vehicle ID, its average speed, its register time, the begin time of encountering, the end time of encountering, and the selecting time, respectively. The $slot_n = (E_n - B_n)$ is the entire duration of communication when the client meets the helper C_n. The B_n and E_n can be calculated by Equations (1) and (2).

$$B_n = T_n + \frac{D + 2L - (T_n - t_s) \times v_s - (T_n - t_n) \times v_n - d}{v_s + v_n} = \frac{D + 2L - d + v_s t_s + v_n t_n}{v_s + v_n} \quad (1)$$

$$E_n = B_n + \frac{2d}{v_s + v_n} = \frac{D + 2L + d + v_s t_s + v_n t_n}{v_s + v_n} \tag{2}$$

As shown in Equations (1) and (2), T_n has nothing to do with B_n and E_n. If the DA is regarded as a linear space on a time axis, then the $slot_n$ becomes a communication slot in which a cooperative vehicle communicates with the client. The strategy guarantees no overlap of the slots for improving channel utilization.

Assuming that the bandwidth between vehicles is W, the cooperative vehicle n should carry the data of $W \times (E_n - B_n)$ in theory. However, due to the characteristics of bad channel quality and high packet loss in the environment of VANET, cooperative vehicles have to carry fewer data than the theoretical value to compensate packet loss.

The traditional DFC can be utilized to solve packet loss, but VANET cooperative communication consists of serval links and communication time is short between the source and the destination. Therefore, it is difficult to converge the communicating process though feedback from receiver to source. This problem will result in a longer transmission delay. To solve the problem, we propose distributed DFC to relay data among the client and the helpers. Since the relative vehicle speeds running at the same direction is slower and the communicating time is longer, in this method, we only make use of SDCD (same direction cooperative downloading) and employ DFC with a hierarchical feedback mechanism to decrease transmission delay.

3.2. The Algorithm of Distributed Digital Foundation Code

To illustrate the scheme clearer, we turn the two processes in Figure 1 into a three-node communication model. As shown in Figure 3, the original cooperative vehicle S encounters the client D at its slot in DA and sends encoding packets to D. The vehicle C running at the same direction with D receives these packets and sends data to the client D when S leaves out the communication area of the client. The source (S) and the cooperative vehicle (C) stop transmitting packets when they leave out the communication area of the destination (D).

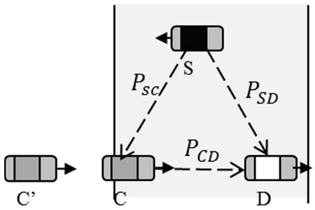


Figure 3. Three-node communication model.

Assuming that the channels between nodes are erasure channels, the erasure probabilities are P_{SC}, P_{SC}, and P_{CD}, respectively. S and C encode the receiving packets into LT-encoded packets. S utilizes the degree distributed function $f_s(x)$ to encode the raw data before it broadcasts to C and D. When C receives the encoded packets, instead of decoding, it utilizes another degree distributed function $f_c(x)$ to encode the packets secondly before sending to D. D employs BP (Belief Propagation) algorithm to decode those encoded packets from S and C. If C leaves the area of D, another cooperative vehicle C' continues to relay the encoded packets. The process does not stop until D receives enough packets to restore the raw data. Unlike traditional digital fountain code, D does not need to feedback to the original cooperative vehicle because it might have already left the communication area of S without receiving enough packets to restore the data. In this method, therefore, D only sends feedback to the last senders when finishing transmission. In order to obtain better decoding performance, the key challenge is how to design the degree distributed functions $f_s(x)$ and $[f_c(x), \ldots, f_{c^n}(x)]$ so that D can easily decode the encoded packets using the degree distributed function $\theta(x)$. An algorithm of distributed digital fountain code is given as follows.

Assuming that the source node *S* will transmit *n* packets, it encodes the raw data using DFC with the distributed function $f_s\,(x)$ before broadcasting the encoded packets to *C* and *D*. The probability of which the packets are received correctly through the link *S*-*D* is $(1 - P_{SD})$, while the probability of which the packets arrive at *C* through the link *S*-*C* is $(1 - P_{SC})$. *C* does not decode the packets, but encodes them secondly using the distributed function $f_c\,(x)$ before it sends the packets to *D*, and the probability of delivering is $(1 - P_{CD})$. Therefore, via *C*, the probability of which packets are received correctly by *D* is $(1 - P_{SC})\,(1 - P_{CD})$. In this model, there are two ways that the packets are transmitted from *S* to *D*. Assuming that the probability of which *D* receives encoded packets from *C* is *P1*, while the probability of receiving from *S* is *P2*, then

$$P_1 = \frac{(1 - P_{SC})\,(1 - P_{CD})}{(1 - P_{SD}) + (1 - P_{SC})\,(1 - P_{CD})}; \tag{3}$$

$$P_2 = \frac{(1 - P_{SD})}{(1 - P_{SD}) + (1 - P_{SC})\,(1 - P_{CD})}. \tag{4}$$

Assuming that the degree distribution function of *S*, *C*, and *D* are $f_s\,(\mathrm{x}) = \sum_{i=1}^{D_s} f_i x^i$, $f_c\,(\mathrm{x}) = \sum_{i=1}^{D_c} f_i x^i$, $\theta\,(x) = \sum_{i=1}^{k} \theta_i x^i$. Obviously, $D_s{\cdot}D_c = k$ and all probability of the degree distributed functions are supposed to satisfy $f_s\,(x) = f_c\,(x) = \theta\,(x) = 1,\ f_s, f_c, \theta \in [0,1)$. If the degree distributed function of the packets received by *D* approximately equals the degree distribution function of DFC $\theta\,(x)$, then

$$P_1{\cdot}\,(f_c\,(f_s\,(x))) + P_2{\cdot}\,(f_s\,(x)) = \theta\,(x). \tag{5}$$

To analyze easily, we employ a matrix formula to represent (5)

$$\delta{\cdot}\omega = \theta. \tag{6}$$

In Equation (6), $\theta = [\theta_1, \theta_2, ..., \theta_k]^T$, $\omega = [P_1 f_{c_1} + P_2, P_1 f_{c_2} + P_2, \ldots,\ P_1 f_{c_k} + P_2]^T$, $\delta = \left(\delta_{i,j}\right)_{1 \le i \le k, 1 \le j \le D_C}$, then

$$\delta_{i,j} = \sum \frac{j!}{i_1! \ldots i_{D_{fs}}!} f_{s_1}^{\,i_1} f_{s_2}^{\,i_2} \ldots f_{s_k}^{\,i_k}. \tag{7}$$

Because the equation set is nonlinear, it is difficult to obtain an algebraic solution. If we can get a suitable δ, ω can be determined by linear equation set uniquely. The coefficient of a distributed function of LT coding attenuates with the increase of the degree of distributed function. Therefore, δ can be represented by a $D_{f_c} \times D_{f_c}$ matrix.

According to the fundamental theorem of linear algebra, the necessary and sufficient condition for which ω has real solutions is that the rank of δ equals the rank of $(\delta\ \theta)$. According to the characteristic of the lower triangular matrix, the rank of δ is D_{f_c}. Therefore, the rank of $(\delta\ \theta)$ is also D_{f_c}, namely,

$$\det \begin{bmatrix} \delta_{i,1} & \delta_{i,2} & \cdots & \delta_{i,D_{fc}} & \theta_{i_1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \delta_{i_{D_{fc}},1} & \delta_{i_{D_{fc}},2} & \cdots & \delta_{i_{D_{fc}},D_{fc}} & \theta_{i_{D_{fc}}} \end{bmatrix} = 0. \tag{8}$$

We deduce the relations further to simplify the formula. Defining $\varnothing_m = (\delta_{m1}\ \delta_{m2}\ \ldots \delta_{mD_{fc}}, \theta_m)$ $= i_{D_{fc}+1}$. If $\sum_{1 \le m \le D_{fD}} i_m \le D_{fc} - 1$, then

$$\textstyle\sum_{1 \le m \le D_{fD}} m{\cdot}i_m \le D_{fs}{\cdot}(D_{fc} - 1). \tag{9}$$

Combing Equation (5) with (7), we can deduce Equation (10):

$$\delta_{mD_{fc}} \varnothing_k - \delta_{kD_{fc}} \varnothing_m = (0\ \ldots 0\ 0\ \delta_{mD_{fc}} \theta_k - \delta_{kD_{fc}} \theta_m). \tag{10}$$

According to Equations (6) and (8), we can deduce Equation (11).

$$\delta_{mD_{fc}} \theta_k - f_{s_{D_{fs}}}^{\ D_{fc}} \theta_m = 0. \tag{11}$$

Assuming that

$$P_h = \sum \frac{D_{fc}!}{i_{h+1}! \ldots i_{D_{fs}}!} f{s_{h+1}}^{i_{h+1}} f{s_{h+2}}^{i_{h+2}} \ldots f{s_{D_{fs}}}^{i_{D_{fs}}}, \tag{12}$$

$1 \leq \text{h} \leq (D_{fs} - 1)$. According to Equations (7) and (12), we can deduce Equation (13).

$$\begin{aligned} \delta_{D_{fs}\cdot(D_{fc}-1)+h, D_{fc}} &= D_{fc} f{s_{D_{fs}}}^{D_{fc}-1} fs_h + P_h \\ \rightarrow \quad \frac{f{s_{D_{fs}}}^{D_{fc}} \cdot \theta_{D_{fs}(D_{fc}-1)+h}}{\theta_k} &= D_{fc} f{s_{D_{fs}}}^{D_{fc}-1} fs_h + P_h \\ \rightarrow \quad fs_h &= \frac{f{s_{D_{fs}}}^{D_{fc}} \cdot \theta_{D_{fs}(D_{fc}-1)+h}}{\theta_k} - \frac{P_h}{D_{fc} \cdot f{s_{D_{fs}}}^{D_{fc}-1}}. \end{aligned} \tag{13}$$

Due to $\sum_{h=1}^{D_{fs}} fs_h = 1$, Equation (14) is deduced from (13).

$$fs_{D_{fs}} = \frac{1}{1 + \sum_{h=1}^{D_{fs}-1} \frac{\theta_{k-1}}{D_{fc} - \theta_k}}. \tag{14}$$

Therefore, fs_h can be obtained by recursion in sequence, while it is easy to solute fc_h using the result of fs_h and θ_i.

In order to compare the delay using DDFC with traditional DFC to download the file in theory, we used the two methods to download the same file. The file was divided into n packets and the length of every packet is l. The overhead of the decoding of the client was ε and source node and the relay node uses degree distributed function $f_s(x)$ and $f_c(x)$ to encode the packets, respectively (there is one relay node in this example). The relay node encodes packets directly before sending to the client. Because the relay node does not need to decode data, we can get the transmission time using DDFC as Equation (15):

$$t = \frac{ak(1+\varepsilon)\left(\frac{W_n}{l}\right) + t_{ack}}{k(1-P_{SD})(1-P_{CD})}. \tag{15}$$

The a is defined as a repeated index. The source node and relay node might send the same packet to the client, so the value of a is $1 \geq a \geq 2$; t_{ack} is the ACK time which was returned to the relay node by the client when the data were decoded correctly.

Accordingly, using traditional DFC, the relay node has to decode the data received from the source node before re-encoding the data and sending to the client. Furthermore, it needs cascading feedback to the source. These processes increase transmission delay. The delay is as follows:

$$t = \frac{ak(1+\varepsilon)\left(\frac{W_n}{l}\right)(1 + P_{SD} - P_{SC} - P_{CD}) + 2t_{ack}}{k(1-P_{SD})(1-P_{CD})}. \tag{16}$$

Obviously, as shown at Equation (16), although using traditional DFC can reduce the influence of retransmission due to packet loss, frequent encoding and decoding increase processing delay. Therefore, traditional DFC does not adapt to VANET, since the nodes move fast and topology change frequently. Compared with that, DDFC using distributed encoding method reduces the procedure of the encoding and decoding of relay nodes so that the transmission efficiency of cooperative downloading of VANET is greatly improved.

4. Performance Evaluation

4.1. Simulation Methodology

In this section, we employ a simulation experiment to evaluate the performance of the proposed method. The simulator VanetMobiSim [18] based on JAVA is utilized to conduct the experiments.

Our objectives in conducting this evaluation include a data receiving rate of different traffic arrivals, vehicle speed and the changing speed rate, the amount of downloaded data, and the delay of downloading different size files.

Compared with traditional methods, the DDFC can improve the data receiving rate (DRR) greatly to increase the amount of downloaded data and reduce the delay. Therefore, we regard the DRR as an import index to evaluate the performance of DDFC. The DRR is formally defined as following:

$$\text{DRR} = \frac{\sum \textit{data packets received by the destination}}{\sum \textit{packets generated by the source}} \tag{17}$$

The experiment scenario is shown in Figure 4. The coverage area of AP is set to 800 m, while the communication radius of vehicles is set to 300 m according to [19]. We roughly assume 1 s as overhead to set up connection, 150 Kbytes/s, 200 Kbytes/s, and 50 Kbytes/s as the downloading speed of the APs, and SDCD (same direction cooperative downloading) and ODCD (opposite direction cooperative downloading) driving vehicles, respectively [20]. The range of vehicle speed is set from 60 km/h to 120 km/h. p is defined as the changing speed rate in accordance with log-normal distribution [17].

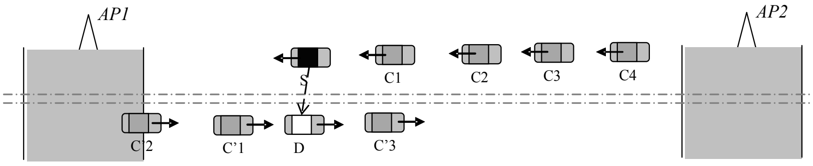


Figure 4. The experiment scenario.

4.2. Simulation Results

Firstly, we compared packet loss rates (PLRs) using four different methods (DDFC, DFC-LT, DFC-Relay, and DSRelay). DSRelay [1] does not employ digital fountain code. DFC-LT [13] is the method of transmitting packets from the source to the destination directly using DFC. DFC-Relay transmits packets by aid of cooperative vehicles using traditional DFC. DDFC is the approach proposed in this paper.

Figure 5 illustrates the result of when the speeds of vehicles are from 60 km/h to 120 km/h and p = 20%, assuming that the traffic arrival follows a Poisson distribution with a rate of one vehicle per λ seconds (λ = 10). Obviously, PLRs of the four methods rise with the increase in the vehicle speed. DSRelay does not employ DFC. Therefore, the PLR is higher compared with the other methods. Interestingly, when the vehicle speed is 120 km/h, the PLR arrives at 25%. In contrast, the other methods using DFC have a lower rate. Their PLRs keep below 5% when the speed changes from 60 to 120 km/h.

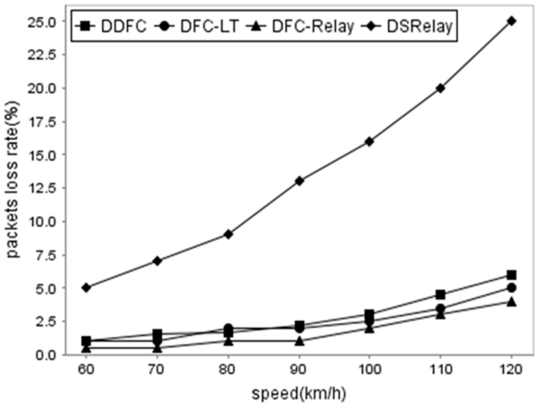


Figure 5. Comparison of packet loss rates using the DDFC, DFC-LT, DFC-Relay, and DSRelay methods.

Although there are no obvious differences in PLR using the three methods with DFC, DDFC has better performance in throughput. Figure 6 shows the throughputs using the four methods to download a 4-Mbytes file. The client can download nearly all packets using DDFC. Due to requiring feedback, the amount of data downloaded by the client using DFC-LT and DFC-Relay are about 20% lower than DDFC. In contrast, the amount of data using DSRelay is higher than DFC-LT and DFC-Relay when the vehicle speed is lower than 100 km/h because of cooperative downloading, while the high PLR brings about poor performance when the speed of the client not using DFC is higher than 100 km/h.

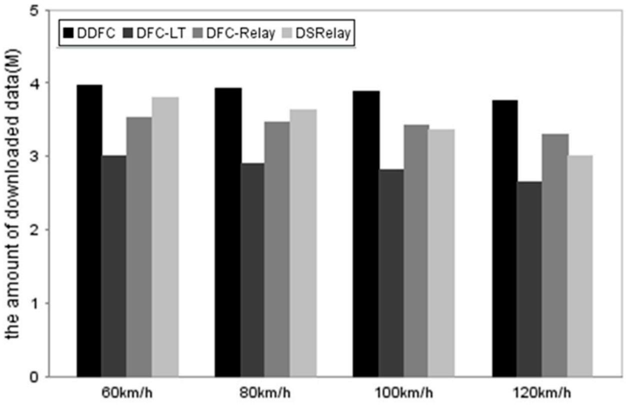


Figure 6. Comparison of the amount of downloaded data using the DDFC, DFC-LT, DFC-Relay, ODCD, and DSRelay methods.

The results indicate that, although the PLR of DDFC is a little higher than those of DFC-LT and DFC-Relay, the client can download more data using DDFC compared with the two methods.

Furthermore, we used the four methods to download three files—an 8.1-MBytes file, a 55.3-MBytes video, and a 379.8-MBytes film files. Table 1 shows that the time lengths of downloading the three files are 231 s, 1284 s, and 7980 s using DSRelay. By comparison, DFC-LT and DFC-Relay using the technology of DFC have no obvious advantage to downloading small size files, while the time length drops dramatically when downloading big size files. This is because DFC improves the transmission efficiency and reduces the delay from packet loss. DDFC using cascading feedback displays better performance. The result shows that it only spends 6570 s downloading 379- MBytes files. The transmission efficiency rises by about 20%.

Table 1. Comparison of delay downloading files using the four methods.

Length	DSRelay	DFC-LT	DFC-Relay	DDFC
8.1 MB	231 s	243 s	228 s	225 s
55.3 MB	1248 s	1225 s	1185 s	1108 s
379 MB	7980 s	7690 s	7108 s	6570 s

In order to validate the performance of the proposed method, we compare the data receiving rate of the vehicles using DDFC in different traffic flow, the changing speed rate, and the different vehicle speeds, respectively. Figure 7 shows that the traffic flow has an influence on DRR because more cooperative vehicles means more opportunities for the client to obtain packets. The DRR is lower when the vehicle speed is faster. Figures 8 and 9 illustrate the influence from the changing speed rate. Obviously, the DRR is higher than 97% when the vehicle speed remains invariable, while the rate drops slightly when p = 20%~50% and the vehicle speed is slower than 90 km/h. The DRR decreases rapidly to about 6.5% when the vehicle speed arrives at 120 km/h.

The results indicate that the speed is still an important factor to DRR because of the Doppler effect. Furthermore, traffic flow also influences DRR because more cooperative vehicles result in more communication collision. Besides, the possibility of speed change and DRR are in the inverse ratio according to the results.

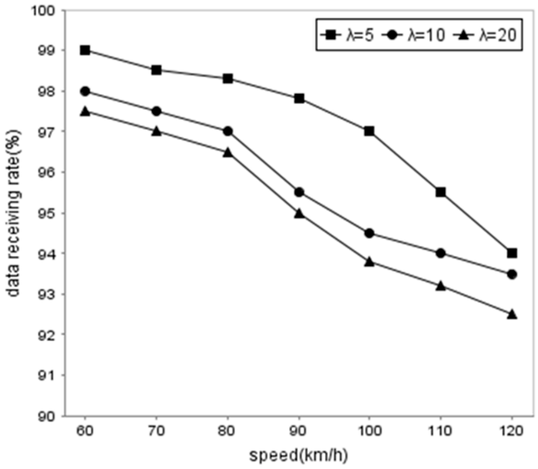


Figure 7. Comparison of data receiving rate generate in different traffic flow, average speed.

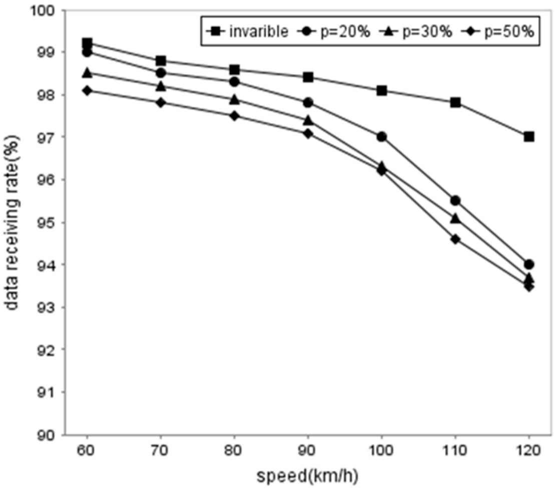


Figure 8. Comparison of data receiving rate generate in different *p* and average speed.

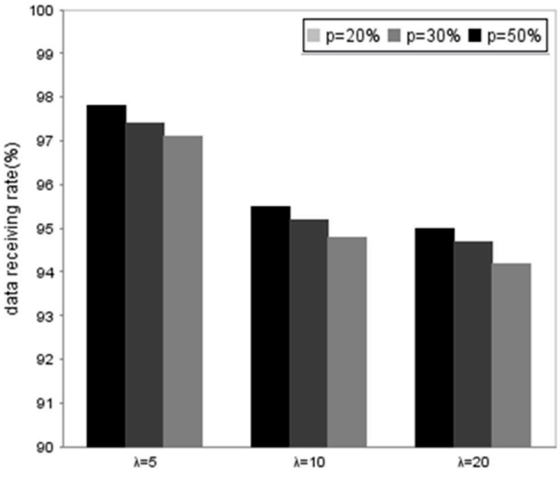


Figure 9. Comparison of data receiving rate generate in different traffic flow and *p*.

5. Conclusions

In this paper, we propose the cooperative downloading method for VANET using DFC to increase the amount of downloaded data and enable the transmission to be more robust in a vehicular environment. The requested data is not only directly transferred between an access point and a passing vehicle, but further vehicles running in DA are being used as relays. By making use of DFC, the client can recover the raw data once it receives enough encoded packets from the source vehicle or cooperative vehicles. Besides, the cascading encoding mechanism avoids encoding and decoding processes in relay nodes, which reduces the compute complexity. Simulation results indicate that DDFC can reduce packet loss rate efficiently, while the amount of downloaded data increases by about 20% compared with other DFC methods. It implies that the approach is more robust to occasional packet losses, especially in a highway scenario with low coverage and a high communication cost of cellular systems. In future work, we will continue to research the influence of multi-clients cooperative downloading in multi-channels using distributed DFC.

Acknowledgments: This work has been supported by the Shandong Provincial Natural Science Foundation, China, No. ZR2014FM017; Fundamental Research Funds for the Central University, China, No. 15CX05025A; the Nature Science Foundation of China under Grant No. 61301139; and the China Scholarship Council No. 201606455011.

Author Contributions: Haihua Chen and Jianhang Liu conceived and designed the experiments; Qi Wang performed the experiments; Wenbin Zhang analyzed the data; Yi Sun and Xuerong Cui contributed reagents/materials/analysis tools; Jianhang Liu and Shibao Li. wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, J.; Bi, J.; Ge, Y.; Cui, X.; Ding, S.; Li, Z. A Compensation Model of Cooperative Downloading for Vehicular Network. *Trans. Emerg. Telecommun. Technol.* **2013**, *24*, 532–543. [CrossRef]
2. Byers, J.W.; Luby, M.; Mitzenmacher, M. A digital fountain approach to reliable distribution of bulk data. In Proceedings of the IEEE/ACM Proceedings of SIGCOMM, Vancouver, BC, Canada, 1998; pp. 56–67.
3. Moczar, Z.; Molnar, S. On the Dynamic Behavior of Digital Fountain Based Communication. In Proceedings of the IEEE Global Communications Conference, San Diego, CA, USA, 6–10 December 2015; pp. 1–6.
4. Ott, J.; Kutscher, D. A Disconnection-Tolerant Transport for Drive-thru Internet Environments. In Proceedings of the IEEE Infocom, Miami, FL, USA, 2005; Volume 3, pp. 1849–1862.
5. Chen, B.B.; Chan, M.C. MobTorrent: A FrameWork for Mobile Internet Access from Vehicles. In Proceedings of the IEEE Infocom, Barcelona, Spain, 19–25 April 2009.
6. Jia, D.; Zhang, R.; Lu, K. Improving the Uplink Performance of Drive-Thru Internet via Platoon-Based Cooperative Retransmission. *IEEE Trans. Veh. Technol.* **2014**, *63*, 4536–4545. [CrossRef]
7. Wu, Y.; Zhu, Y.; Li, B. Ingrastructure-Assisted Routing in Vehicular Networks. In Proceedings of the IEEE Infocom, Turin, Italy, 14–19 April 2013; pp. 1485–1493.
8. Liu, J.; Bi, J.; Xu, P. A Compensation Model of Cooperative Downloading Improving System Throughput. *Chin. J. Comput.* **2012**, *35*, 1390–1398. [CrossRef]
9. Liu, B.; Wu, L.; Jia, D. Data Uplink Strategy in Mobile Cloud Service Based Vehicular Ad Hoc Network. *J. Comput. Res. Dev.* **2016**, *53*, 811–823. (In Chinese)
10. Razavi, R.; Claussen, H. Digital Fountain Codes with Reduced Latency, Complexity and Buffer Requirements for Wireless Communications. In Proceedings of the IEEE 79th Vehicular Technology Conference, Seoul, Korea, 18–21 May 2014; pp. 1–5.
11. Lei, W.-J.; Xie, X.-Z.; Li, G.-J. The scheme and performance of wireless cooperative relay system using digital fountain codes. *Acta Electron. Sin.* **2010**, *38*, 228–233.
12. Mao, Y.; Huang, J. A new iterative LT decoding algorithm for binary and nonbinary Galois fields. *J. Commun. Netw.* **2013**, *15*, 411–421. [CrossRef]
13. Cao, R.; Yang, L. Decomposed LT codes for cooperative relay communications. *IEEE J. Sel. Areas Commun.* **2012**, *30*, 407–414. [CrossRef]

14. Ahmed, S.; Kanhere, S.S. VANETCODE: Network coding to enhance cooperative downloading in vehicular ad-hoc networks. In Proceedings of the International Conference on Wireless Communications and Mobiling Computing, Vancouver, BC, Canada, 3–6 July 2006; pp. 527–532.
15. Hao, Y.; Tang, J.; Cheng, Y. Secure Cooperative Data Downloading in Vehicular Ad Hoc Networks. *IEEE J. Sel. Areas Commun.* **2013**, *31*, 523–537.
16. Li, C.; Jose, J.; Wu, X. Distributed-fountain network code (DFNC) for content delivery in vehicular networks. In Proceeding of the VANET Workshop on Vehicular Inter-Networking, Systems, and Applications, New York, NY, USA, 6 January 2013; pp. 31–40.
17. Gerlough, D.L. *Use of Poisson Distribution in Highway Traffic*; Eno Foundation: Westport, CT, USA, 1955.
18. VanetMobiSim. Available online: https://sourceforge.net/projects/vanetmobisim/ (accessed on 20 December 2015).
19. Ott, J.; Kutscher, D. Drive-thru Internet: IEEE 802.11b for "Automobile" Users. In Proceedings of the IEEE Infocom, Hong Kong, China, 7–11 March 2004; pp. 362–373.
20. Bai, F.; Stancil, D.; Krishnan, H. Toward Understanding Characteristics of Dedicated Short Range Communications (DSRC) from a Perspective of Vehicular Network Engineers. In Proceedings of the 16th Annual International Conference on Mobile Computing and Networking, Chicago, IL, USA, 20–24 September 2010.