entropy

Article

Efficient Communications in V2V Networks with Two-Way Lanes Based on Random Linear Network Coding

Yiqian Zhang [1,2], Tiantian Zhu [1] and Congduan Li [1,2,*]

[1] School of Electronics and Communication Engineering, Sun Yat-sen University, Shenzhen 518107, China; zhangyq75@mail2.sysu.edu.cn (Y.Z.); 18705168317@163.com (T.Z.)
[2] Shenzhen Key Laboratory of Navigation and Communication Integration, Shenzhen 518107, China
* Correspondence: licongd@mail.sysu.edu.cn

Abstract: Vehicle-to-vehicle (V2V) communication has gained significant attention in the field of intelligent transportation systems. In this paper, we focus on communication scenarios involving vehicles moving in the same and opposite directions. Specifically, we model a V2V network as a dynamic multi-source single-sink network with two-way lanes. To address rapid changes in network topology, we employ random linear network coding (RLNC), which eliminates the need for knowledge of the network topology. We begin by deriving the lower bound for the generation probability. Through simulations, we analyzed the probability distribution and cumulative probability distribution of latency under varying packet loss rates and batch sizes. Our results demonstrated that our RLNC scheme significantly reduced the communication latency, even under challenging channel conditions, when compared to the non-coding case.

Keywords: random linear network coding; vehicle-to-vehicle communication; dynamic topology; latency reduction

check for updates

Citation: Zhang, Y.; Zhu, T.; Li, C. Efficient Communications in V2V Networks with Two-Way Lanes Based on Random Linear Network Coding. *Entropy* **2023**, *25*, 1454. https://doi.org/10.3390/e25101454

Academic Editor: Syed A. Jafar

Received: 19 September 2023
Revised: 9 October 2023
Accepted: 13 October 2023
Published: 17 October 2023

1. Introduction

Vehicles serve as vital means of transportation in urban cities, necessitating increased intelligence, as the intelligence of a single car falls short of meeting the requirements for road safety, path planning, decision-making, and traffic efficiency. To address these challenges, the Internet of Vehicles (IoV) has been introduced, enabling communication and collaboration among vehicles, and playing a crucial role in slow-vehicle warnings, intersection collision warnings [1], as well as congestion alleviation, emission reduction, and time saving [2]. However, these networks face obstacles in terms of mobility and occlusion. Notably, when a vehicle is traveling at a speed of 120 km/h, a mere 1-second latency can result in a driving distance of 33 m, potentially leading to severe consequences. Initially, communication among vehicles relied on dedicated short-range communication (DSRC) technology [3], which facilitated short-range communication with low latency [4]. Nevertheless, in high-speed scenarios, vehicles often move out of the communication range, rendering DSRC inadequate. The advent of 5G technology has introduced the concept of ultra-reliable and low-latency communications (uRLLC) [5], enabling vehicles to maintain communication even during high-speed mobility scenarios. To minimize long-term content access costs in vehicle-to-vehicle (V2V) networks, ref. [6] considered a distributed multi-agent reinforcement learning (MARL)-based edge caching method and proposed a distributed MARL-based edge caching method (DMRE), where every agent adaptively learns optimal caching strategies in collaboration with others. Additionally, they integrated the advantages of deep Q-Networks into DMRE, resulting in a computationally efficient method named DeepDMRE, which utilizes neural networks to approximate Nash equilibria. Such deep Q-Networks were also considered in [7], to explore the integration of reconfigurable intelligent surfaces (RIS) with unmanned aerial vehicles (UAVs) in the downlink of non-orthogonal multiple-access (NOMA) networks. They proposed a joint optimization scheme using deep Q-networks to maximize system capacity, while considering UAV energy constraints and demonstrating significant improvements in system capacity.

Network coding [8] offers a promising solution for enhancing the performance of communication systems in V2V networks. By employing network coding techniques, intermediate nodes in the network can encode the received messages before transmitting them to the next hop, and the sink node decodes the received messages to reconstruct the original information. Ref. [9] proposed the use of XOR network coding in fault-tolerant dynamic scheduling and routing algorithms for time-sensitive in-vehicle networks (IVNs), to increase throughput, reliability, and robustness. Experimental results demonstrated that the XOR network coding scheme outperformed the frame replication and elimination for reliability (FRER) mechanism in terms of schedulability, flow, and response time, because the FRER mechanism tends to over-utilize the available bandwidth, whereas XOR network coding provides a better performance without excessive bandwidth usage. Ref. [10] expanded upon the security and privacy considerations in V2V networks as the number of vehicles accessing the network increases and proposed a comprehensive scheme that combines network coding, relay collaboration, and homomorphic encryption. The scheme ensures that the original information remains inaccessible to relay nodes, except for the intended target vehicle node. It also protects against potential collusion attacks, preventing conspiratorial attackers or multiple relay nodes from recovering the original information. Theoretically, such schemes guarantee the confidentiality, privacy protection, and anti-collusion capabilities of V2V networks. In [11], F. Ye et al. adopted network coding in vehicular ad hoc networks (VANETs) by modeling platoon vehicles driving in the same direction on a highway as a 1-D lattice network, in which a single source node aims to disseminate messages to all other vehicles. They analyzed the theoretical upper bound of the benefits achieved through network coding and conducted simulations to demonstrate the performance superiority over random broadcasting using Rayleigh fading wireless channels. F. Liu et al. [12] extended the data dissemination in VANETs in [11] to a two-way lane scenario by modeling the network as two separate 1-D lattice networks, corresponding to the two directions of traffic flow. They divided the dissemination into the *encountering* phase and the *separated* phase, determined by whether the broadcasting coverage areas of the two disseminators overlapped, which means vehicles traveling in both direction can communicate with both disseminators simultaneously. They analyzed the impact of the opposite direction over the traditional one-way lane model and showed that two disseminators traveling in opposite directions can enhance the speed of data dissemination. Ref. [13] compared three methods in a highway data mulling scenario, with vehicles from the opposite direction as data mutes to transmit large multimedia files, modeled as a *coupon collector* problem, and among which the network-coding-based strategy outperformed erasure-coding and repetition-coding strategies. The literature [14–16] shows that network coding can improve reliability and throughput, but it fails to deal with dynamic situations where the vehicle volume increases rapidly and the network structure becomes complex. Therefore, random linear network coding (RLNC) [17] has garnered significant attention, particularly for its ability to operate without prior knowledge of the network topology. RLNC involves random coding coefficient selection from a finite field and performing linear operations on the packets. As the vehicular scale increases, the random selection of RLNC encoding coefficients within a finite field obviates the need to account for variations in node quantity and network topology within this method. By receiving a sufficient number of packets with independent coefficients at the sink, the original information can be decoded at source, which enables transmitting content over wireless vehicle communications with lossy links and that are highly dynamic. Ref. [18] proposed a RLNC scheme for data transmission in a one-way lane V2V network, modeled as a multi-source multi-relay single-sink broadcasting network, to reduce latency and enhance the network robustness. In this one-way lane V2V communication scenario, the leading vehicles relay the detected road conditions and critical safety alerts to those following behind, affording them sufficient time for well-informed decision-making. This type of information, with its small data payload, facilitates swift transmission with no node departures in multi-round communication processes, as assumed in [18], and implies a static and unchanging network topology. This may not align with the evolving landscape of intelligent transportation. In particular, with the increasing demand for in-vehicle entertainment experiences, expediting the transmission of large-scale data from nearby vehicles has become essential. Given the significant data volume involved, this study explores the utilization of vehicles in the opposite lane to establish a framework for bidirectional V2V large-scale data transmission over an extended period. During prolonged communication sessions for large-scale data transmission, nodes at high speeds tend to exit the communicable range of receiving vehicles, leading to dynamic changes in the network topology over multiple rounds. In this extended two-way lane large-scale data transmission scenario, the network is modeled as a multi-source single-sink network with a dynamic topology, where cars may enter or leave the communication range, resulting in a varying number of sources each round. The destination car node receives information from cars traveling in both the same and opposite directions. By utilizing RLNC in this dynamic two-way lane model, the proposed scheme enhances throughput and robustness, without relying on a specific network topology.

The main contributions of this paper are as follows:

- it extends the one-way lane model proposed in [18] to incorporate two-way lanes, thereby creating a dynamic network model;
- the paper provides a lower bound on the generation probability, demonstrating the feasibility and effectiveness of the RLNC scheme;
- it evaluates the performance of the RLNC scheme under frequently changing network conditions and poor channel conditions. The results demonstrate that RLNC significantly reduces latency compared to non-coding schemes.

The rest of this paper is organized as follows: Section 2 provides a brief overview of RLNC and compares our work with the related literature. Section 3 presents a system model, detailing the two-way lane RLNC transmission scheme and conducting an analysis of the generation probability and time delay. In Section 4, we analyze the simulation performance for communication delays under different packet loss rates and batch sizes, and then compare the coding and non-coding schemes. Section 5 concludes the paper.

2. Related Work

In this section, we give details about RLNC and introduce the reasons why RLNC is used. Then, the recent related literature for RLNC in V2V networks is compared with our work.

2.1. Brief of RLNC

We first give a brief introduction to RLNC. Li et al. proposed linear network coding (LNC) in [19], where they allowed intermediate nodes in the network to perform operations on the incoming packets, combining them linearly before forwarding. At the receiving end, the nodes can then decode the received combinations, to retrieve the original information. Ho et al. then proposed LNC in a randomized setting [20], where the coding coefficients are randomly chosen in a fixed-size finite field. Figure 1 illustrates a straightforward application of RLNC in a butterfly network. Source node s is tasked with sending messages X_1 and X_2 to sinks t_1 and t_2. Each channel can transmit only one message during a given time slot. Node s sends the linearly encoded X_1 and X_2 with the randomly selected coefficients (ξ_1, ξ_2), resulting in $\xi_1 X_1 + \xi_2 X_2$, to node 1. This information is then forwarded to nodes 3 and t_1. Similar operations occur at node 2, with randomly chosen coefficients (ξ_3, ξ_4). Given that node 3 receives two messages but can only utilize one channel to communicate with node 4, it becomes imperative to perform linear network coding at node 3 using randomly chosen coefficients (ξ_5, ξ_6). Subsequently, node 4 forwards the encoded message to both sinks. The messages received at t_1 are denoted as Y_{11} and Y_{12}, which is

$$\begin{bmatrix} Y_{11} \\ Y_{12} \end{bmatrix} = \begin{bmatrix} \xi_1 & \xi_2 \\ \xi_5\xi_1 + \xi_6\xi_3 & \xi_5\xi_2 + \xi_6\xi_4 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}, \tag{1}$$

and the messages received at t_2 are denoted as Y_{21} and Y_{22}, which is

$$\begin{bmatrix} Y_{21} \\ Y_{22} \end{bmatrix} = \begin{bmatrix} \xi_3 & \xi_4 \\ \xi_5\xi_1 + \xi_6\xi_3 & \xi_5\xi_2 + \xi_6\xi_4 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}, \tag{2}$$

With invertible coefficient matrices, the original X_1 and X_2 can be decoded.

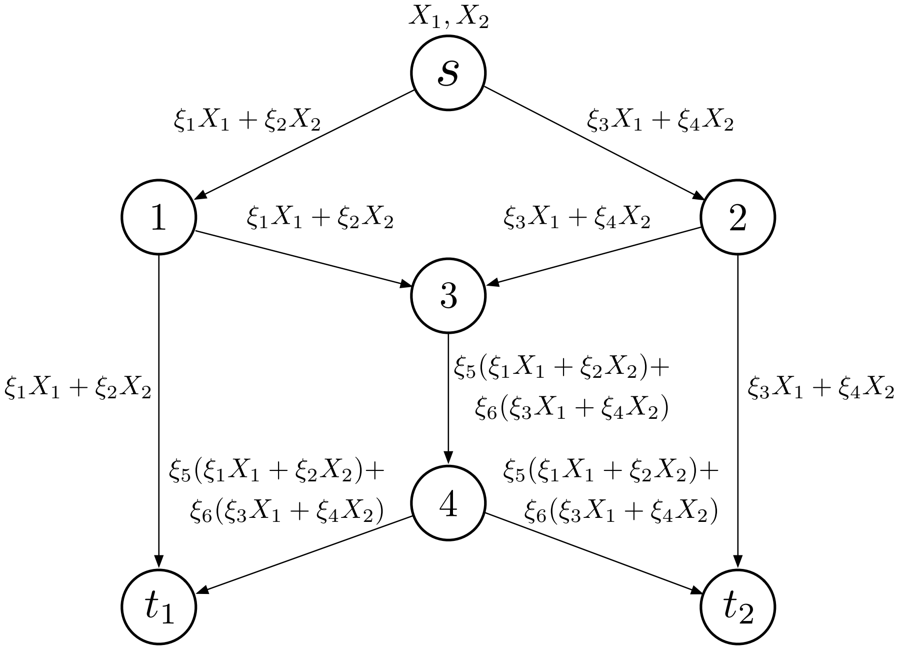
X_1, X_2
s
$\xi_1 X_1 + \xi_2 X_2$
$\xi_3 X_1 + \xi_4 X_2$
1
$\xi_1 X_1 + \xi_2 X_2$
$\xi_3 X_1 + \xi_4 X_2$
2
3
$\xi_1 X_1 + \xi_2 X_2$
$\xi_5(\xi_1 X_1 + \xi_2 X_2)+$
$\xi_6(\xi_3 X_1 + \xi_4 X_2)$
$\xi_3 X_1 + \xi_4 X_2$
$\xi_5(\xi_1 X_1 + \xi_2 X_2)+$
$\xi_6(\xi_3 X_1 + \xi_4 X_2)$
4
$\xi_5(\xi_1 X_1 + \xi_2 X_2)+$
$\xi_6(\xi_3 X_1 + \xi_4 X_2)$
t_1
t_2

Figure 1. RLNC model in a butterfly network.

2.2. RLNC in V2V

Many works have introduced RLNC to V2V communication scenarios. Considering massive gigabit content transmission in millimeter-wave networks, ref. [21] applied symbol-level network coding (SLNC); that is, RLNC at the symbol scale, and utilized a cooperative concurrent distribution strategy in the scenario of highway network topology, where roadside units (RSU) encode the original packets and then forward them to vehicles. The proposed scheme enables collaborative V2V and vehicle-to-infrastructure (V2I) mmWave communications through a greedy network coding strategy based on a graph-theoretic approach. The scheme achieves a low latency, high efficiency, error resilience, and reliability. In ref. [22], E. Tasdemir et al. implemented a dynamic systematic sliding window RLNC scheme for end-to-end communication in vehicle platooning scenarios, where the platooning leader generates packets that are transmitted hop-by-hop to the platooning members. The coding process only involves packets within the dynamically sliding window, which moves forward to include new packets and is closed through feedback, and these packets are combined linearly to generate coded packets using RLNC techniques. This coding scheme was shown through simulation to provide resilience and low latency. To address challenges like transmission collisions and channel fading, ref. [23] proposed a hybrid medium access control (MAC) protocol for basic safety messages (BSMs) dissemination within the DSRC framework. Additionally, this protocol, with three sessions, a MAC setup session, CSMA session, and PNC session integrating physical-layer network coding and RLNC, further enhances the reliability and efficiency of BSM dissemination.

Ref. [24] further analyzed packet delivery ratio performance theoretically and through a comprehensive simulation. Our proposed method is compared with the recent literature works in a comparative table, as Table 1.

Table 1. Comparative table of our proposed method and recent literature.

	Ours	[21]	[22]	[23]	[24]
With the help of RSU	✗	✓	✗	✓	✓
The level of RLNC	Packet level	Symbol level	Packet level	Packet level	Packet level
One/two-way lane	Two-way lane	One-way lane	One-way lane	Two-way lane	Two-way lane and intersection scenario
Data scale	Large scale	Large scale	Large scale	BSM	BSM

3. System Model and RLNC Algorithm

In this section, we give the system model and introduce the RLNC algorithm.

3.1. System Model

First, we build the two-way lane V2V model based on real vehicle road scenarios, as illustrated in Figure 2. In the model, the car receiving messages, denoted as R and travels at a speed of v_R. We have m cars, denoted as $A_1, A_2, \ldots, A_m$, traveling in the same direction as R at constant speeds of $v_{A_1}, v_{A_2}, \ldots, v_{A_m}$, respectively. Additionally, there are w cars, denoted as $B_1, B_2, \ldots, B_w$, traveling in the opposite direction at constant speeds of $v_{B_1}, v_{B_2}, \ldots, v_{B_w}$, respectively. For each $i \in 1, 2, \ldots, m$ and $j \in 1, 2, \ldots, w$, cars A_i and B_j store M identical raw packets to be transmitted. These M raw packets collectively form a *generation*. Once the sink node R receives (or decodes) all M raw packets, the raw packets are updated to transmit the next generation.

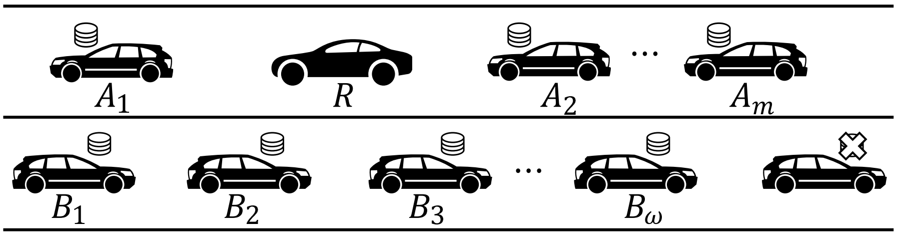

Figure 2. Vehicle road model.

R only communicates with cars within its communication range d. Specifically, R and A_i establish contact only when the distance between them, denoted as $d_{(A_i,R)}$, satisfies $d_{(A_i,R)} < d$. In the case where A_i is positioned ahead of R, the communication between R and A_i can be maintained for

$$t = \frac{d + d_{(A_i,R)}}{v_R - v_{A_i}}; \tag{3}$$

When $v_{A_i} > v_R$, the communication between R and A_i can be maintained for

$$t = \frac{d - d_{(A_i,R)}}{v_{A_i} - v_R}. \tag{4}$$

If $v_{A_i} = v_R$, they can always communicate with each other.

In addition, when A_i is positioned behind R, when $v_{A_i} < v_R$, the communication between R and A_i can be maintained for

$$t = \frac{d - d_{(A_i,R)}}{v_R - v_{A_i}}; \tag{5}$$

When $v_{A_i} > v_R$, the communication between R and A_i can be maintained for

$$t = \frac{d + d_{(A_i,R)}}{v_{A_i} - v_R}. \tag{6}$$

If $v_{A_i} = v_R$, they can always communicate with each other.

Regarding the opposite lane, if car B_i is moving towards R, then

$$t = \frac{d - d_{(B_i,R)}}{v_R + v_{B_i}}; \tag{7}$$

If car B_i is traveling in the opposite direction and is positioned behind car R, R and B_i can keep in touch for

$$t = \frac{d + d_{(B_i,R)}}{v_R + v_{B_i}}. \tag{8}$$

We further extract the model as a multi-source single-sink network, as shown in Figure 3. In this model, the sink node is denoted R. The cars traveling in the same direction are A_{α_t}, where $\alpha_t = 0, 1, 2, \ldots, m$.Similarly, the cars traveling in the opposite direction are denoted as B_{β_t}, with $\beta_t = 0, 1, 2, \ldots, w$. Both the cars in the same direction and those in the opposite direction possess identical sets of M raw data packets, collectively referred to as a generation. These packets are organized into batches to be transmitted. It is important to note that the number of source nodes, denoted by α_t and β_t, may vary in each round.

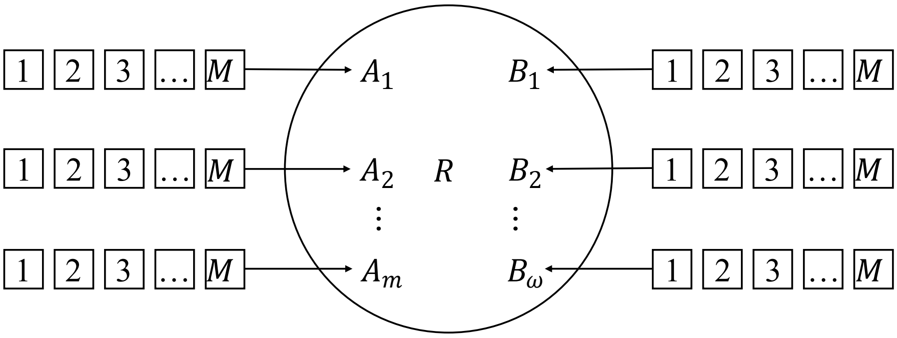

Figure 3. A sketch of a two-way lane model.

3.2. RLNC Algorithm

We now implement the RLNC algorithm, analyze the probability of generation, and present the corresponding algorithms. To implement RLNC, we select encoding coefficients from the finite field $GF(q)$, where the size of the finite field is denoted q. Consequently, we obtain the encoded data packet $\mathbf{\Gamma}_{A_i}$ transmitted by the source node A_i from the same direction as

$$\mathbf{\Gamma}_{A_i} = a_i^1 \boldsymbol{r}_1 + a_i^2 \boldsymbol{r}_2 + a_i^3 \boldsymbol{r}_3 + \cdots + a_i^M \boldsymbol{r}_M, \tag{9}$$

where $\boldsymbol{r}_k$ represents the kth data packet, and $a_i^k \in GF(q)$ denotes the encoding coefficient associated with $\boldsymbol{r}_k$ in the encoded data packet $\mathbf{\Gamma}_{A_i}$. Here, i ranges from 0 to α_t (the number of source nodes in the same direction), and k ranges from 1 to M (the total number of data packets in a generation). Similarly, the encoded data packet $\mathbf{\Gamma}_{B_j}$ from the opposite source B_j is

$$\mathbf{\Gamma}_{B_j} = b_j^1 \boldsymbol{r}_1 + b_j^2 \boldsymbol{r}_2 + b_j^3 \boldsymbol{r}_3 + \cdots + b_j^M \boldsymbol{r}_M, \tag{10}$$

where $b_j^k \in GF(q)$ is the encoding coefficient of $\boldsymbol{r}_k$ in $\mathbf{\Gamma}_{B_i}$, $j = 0, 1, 2, \ldots, \beta_t$, $k = 1, 2, \ldots, M$.

In each time slot, the source nodes collectively transmit the $\alpha_t + \beta_t$ packets that have been encoded using RLNC, which can be represented in matrix form as follows:

$$\begin{bmatrix} \mathbf{\Gamma}_{A_1} \\ \mathbf{\Gamma}_{A_2} \\ \cdots \\ \mathbf{\Gamma}_{A_{\alpha_t}} \\ \mathbf{\Gamma}_{B_1} \\ \mathbf{\Gamma}_{B_2} \\ \cdots \\ \mathbf{\Gamma}_{B_{\beta_t}} \end{bmatrix} = \begin{bmatrix} a_2^1 & a_2^2 & \cdots & a_2^M \\ a_1^1 & a_1^2 & \cdots & a_1^M \\ \cdots & & & \cdots \\ a_{\alpha_t}^1 & a_{\alpha_t}^3 & \cdots & a_{\alpha_t}^M \\ b_1^1 & b_1^2 & \cdots & b_1^M \\ b_2^1 & b_2^2 & \cdots & b_2^M \\ \cdots & & & \cdots \\ b_{\beta_t}^1 & b_{\beta_t}^2 & \cdots & b_{\beta_t}^M \end{bmatrix} \begin{bmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \\ \boldsymbol{r}_3 \\ \cdots \\ \boldsymbol{r}_{M-1} \\ \boldsymbol{r}_M \end{bmatrix} \tag{11}$$

$$= \boldsymbol{C}_{(\alpha_t+\beta_t)\times M} \boldsymbol{R}_{M\times 1}, \tag{12}$$

where $\boldsymbol{C}_{(\alpha_t+\beta_t)\times M}$ is the coefficient matrix and $\boldsymbol{R}_{M\times 1}$ is the raw packet matrix.

3.2.1. Generation Probability

In order to decode M raw packets, the sink node R needs to receive M linearly independent encoded packets. If the encoding coefficient vector of a packet is linearly independent of the encoding data packets previously received, then this packet contributes to the decoding process. We define the number of linearly independent packets received by the sink node as the "sink's state", denoted as S_R. The generation probability, which represents the probability that the encoding packets are linearly independent, takes different forms based on the sink's state. Specifically, it depends on whether S_R is greater than or less than $M - \alpha_t - \beta_t$. To address these scenarios, we give Lemmas 1 and 2.

Lemma 1. *When the sink's state $S_R \leq M - \alpha_t - \beta_t$, generation probability is of the form*

$$P_{ge=\alpha_t+\beta_t} = \prod_{l=i}^{i+\alpha_t+\beta_t-1} \left(1 - \frac{1}{q^{M-l}}\right), \tag{13}$$

where q is the Galois field size, M is the number of raw packets in each batch, and i is the current state of the sink node, $\alpha_t = 0, 1, 2, \cdots, m$, $\beta_t = 0, 1, 2, \cdots, w$, $i = 0, 1, 2, \cdots, M - \alpha_t - \beta_t$.

Proof. Similarly to the proof of Theorem 1 in [18] but with $n = \alpha_t + \beta_t$, source nodes send $\alpha_t + \beta_t$ data packets in a time slot and the n packets are regarded as a group. n sources send n data packets $\mathbf{\Gamma}_{Y_\eta}$ in one time slot. There are $q_M - 1$ choices, resulting in

$$C_{(q^M-1)+\alpha_t+\beta_t-1}^{\alpha_t+\beta_t} = C_{q^M+\alpha_t+\beta_t-2}^{\alpha_t+\beta_t} \tag{14}$$

kinds of combination.

If all the n packets are linearly independent, then there are

$$\frac{\prod_{j=0}^{\alpha_t+\beta_t-1} C_{q^M-q^{i+j}}^1}{A_{\alpha_t+\beta_t}^{\alpha_t+\beta_t}} \tag{15}$$

kinds of combination.

Therefore, the generation probability is

$$P_{ge=\alpha_t+\beta_t} = \frac{\dfrac{\prod_{j=0}^{\alpha_t+\beta_t-1} C_{q^M-q^{i+j}}^1}{A_{\alpha_t+\beta_t}^{\alpha_t+\beta_t}}}{C_{q^M+\alpha_t+\beta_t-2}^{\alpha_t+\beta_t}} = \frac{\prod_{j=0}^{\alpha_t+\beta_t-1} C_{q^M-q^{i+j}}^1}{A_{q^M+\alpha_t+\beta_t-2}^{\alpha_t+\beta_t}}. \tag{16}$$

After simplifying, we can prove this. □

Lemma 2. *When the sink's state is $M > S_R > M - \alpha_t - \beta_t$, that is, $S_R = M - \alpha_t - \beta_t + 1, M - \alpha_t - \beta_t + 2, \ldots, M - 1$, the generation probability is of the form:*

$$P_{ge=M-i} = \prod_{l=i}^{M-1} (1 - \frac{1}{q^{M-l}}), \tag{17}$$

where q is the Galois field size, M is the number of raw packets in each batch, and i is the current sink's state, $i = M - \alpha_t - \beta_t + 1, M - \alpha_t - \beta_t + 2, \ldots, M - 1$.

Proof. Similarly to the proof of Theorem 2 in [18] but with $n = \alpha_t + \beta_t$, the generation probability is

$$P_{ge=M-i} = \frac{\frac{\prod\limits_{j=0}^{M-i-1} C^1_{q^M - q^{i+j}}}{A^{M-i}_{M-i}}}{C^{M-i}_{q^M + M-i-2}} = \frac{\prod\limits_{j=0}^{M-i-1} C^1_{q^M - q^{i+j}}}{A^{M-i}_{q^M + M-i-2}}. \tag{18}$$

After simplifying, we can prove this. □

The generation probability in this context exhibits similarities to the generation probability discussed in [18]. However, in the context of dynamic topology, the generation probability in each round is influenced, not only by the sink's state, but also by the number of currently communicable source nodes. Specifically, the sink's state in time slot t can be expressed as $S_R > M - \alpha_t - \beta_t$, indicating that the generation probability aligns with the conditions stated in Lemma 2. Conversely, prior to the initiation of the $(t+1)$th round of communication, due to multiple vehicles departing the communicable range, in time slot $t+1$, we have $S_R \leq M - \alpha_{t+1} - \beta_{t+1}$. In such cases, the generation probability adheres to the conditions specified in Lemma 1. This distinction arises from the changes in the sink's state and the varying number of communicable source nodes as a consequence of the dynamic topology in the network.

We focus on determining the lower bound of the generation probability. The lower bound is acquired when the sink's state is $M - m - w$ and sources send $m + w$ data packets. According to Appendix B in [18], let $n = m + w$, and we establish the lower bound of generation probability

$$\min P_{ge} = \prod_{l=M-m-w}^{M-1} (1 - \frac{1}{q^{M-l}}), \tag{19}$$

which is equivalent to

$$\min P_{ge} = \prod_{\mu=1}^{m+w} (1 - \frac{1}{q^{\mu}}), \tag{20}$$

Thus, the lower bound of the generation probability depends on the total number of sources at the beginning of communication $m + w$ and the Galois field size q.

Figure 4 illustrates the lower bound of P_{ge}, as given by Equation (20), where n represents the total number of sources ($n = m + w$). As the finite field size increases, the lower bound of generation probability also increases. For example, when considering a finite field size of $GF(256)$, the minimum generation probability exceeds 0.996. Consequently, with a sufficiently large finite field size, it is reasonable to assume that every packet transmitted to the sink is valid, and the generation probability approaches 1. Assuming that, after ζ rounds of communication, the sink node has received M data packets, the decoding probability can be expressed as

$$P_d > \prod_{\mu=1}^{m+w} (1 - \frac{1}{q^{\mu}})^{\zeta}. \tag{21}$$

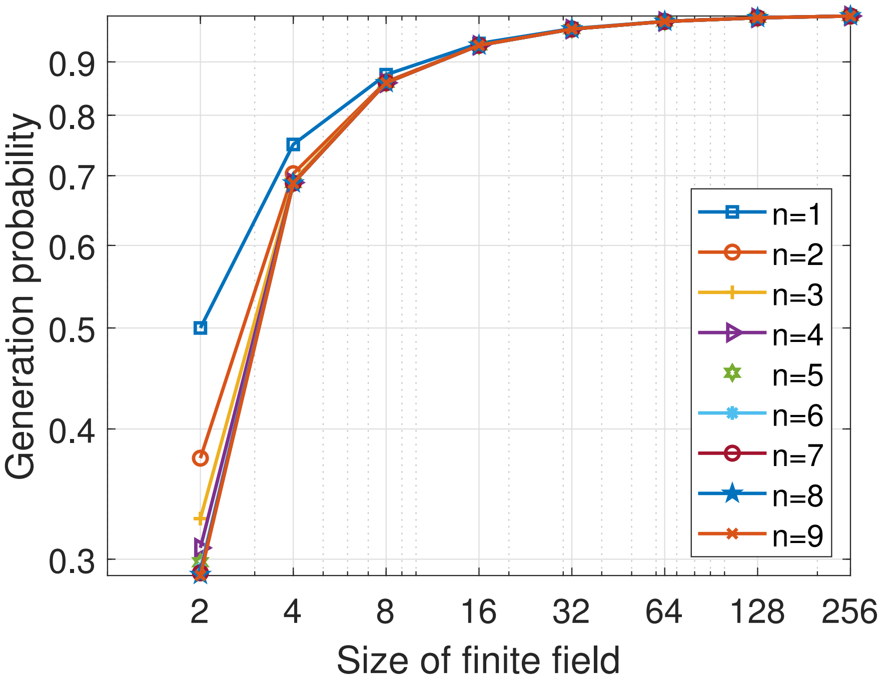

Figure 4. Lower bound of the generation probability with different sources.

3.2.2. Time Delay Analysis

In our analysis of the time delay, we consider the dynamic nature of the participating sources in each round, which differs from the one-way lane scenario described in [18]. To address a two-way lane scenario, we first determine the number of source nodes within the communication range of the sink during each round. This is performed based on the position, speed, and initial distance to the sink, as outlined in Algorithm 1. In Algorithm 1, we utilize an indicator variable **f**. When $\mathbf{f} = 1$, this indicates that the source node is initially positioned ahead of the sink node. Conversely, when $\mathbf{f} = 2$, this indicates that the source node is initially located behind the sink node. The algorithm utilizes this indicator to determine the number of source nodes present in each round, considering their relative positions with respect to the sink node. We tally the number of same direction sources engaged in each communication round. This is contingent upon whether the source is positioned ahead or behind the destination, as well as the relative speeds of the source and destination vehicles, and the relative distance between them. As for the count of counter-directional sources, this hinges on whether the source is located ahead or behind the destination, along with the relative distance between them.

Based on the number of sources participating in each round, $\alpha_t + \beta_t$, we obtain the binomial distribution for the state transition of the sink node. When the sink's state is $S_R = 0, 1, 2, \ldots, M - \alpha_t - \beta_t$, $\alpha_t + \beta_t$ source nodes collectively send $\alpha_t + \beta_t$ valid data packets. The probability of the sink node receiving k valid data packets in this round, which corresponds to a transition to k states in time slot t, can be calculated as

$$P_{mov}(t,k) = C^{k}_{\alpha_t+\beta_t}(1-p_e)^k p_e^{\alpha_t+\beta_t-k}, \tag{22}$$

where p_e denotes the packet loss rate.

Algorithm 1 # of source nodes within communication range in time slot t

```
Input: # of source nodes from the same direction $m$
    # of source nodes from the reverse direction $w$
    communication range $d_b$
    speed of sink $v_b$
    position of source nodes in the same direction $\mathbf{f}$, speed $\mathbf{v}$, distance $\mathbf{d}$
    position of source nodes in the reverse direction $\mathbf{fo}$, speed $\mathbf{vo}$, distance $\mathbf{do}$
    rounds of communication $N$
Initialize # of same direction nodes $\mathbf{Q} = 0$
$\mathbf{Q}(0) = m$
for $i = 1 \to N$ do
    count=0
    for $j = 0 \to m - 1$ do
        if $\mathbf{f}(j) = 1\&\&\mathbf{v}(j) < v_b\&\&j(v_b - \mathbf{v}(j)) - (d_b + \mathbf{d}(j)) > 0$ then
            count=count-1
        end if
        if $\mathbf{f}(j) = 1\&\&\mathbf{v}(j) > v_b\&\&j(\mathbf{v}(j) - v_b) - (d_b - \mathbf{d}(j)) > 0$ then
            count=count-1
        end if
        if $\mathbf{f}(j) = 2\&\&\mathbf{v}(j) < v_b\&\&j(v_b - \mathbf{v}(j)) - (d_b - \mathbf{d}(j)) > 0$ then
            count=count-1
        end if
        if $\mathbf{f}(j) = 2\&\&\mathbf{v}(j) > v_b\&\&j(\mathbf{v}(j) - v_b) - (d_b + \mathbf{d}(j)) > 0$ then
            count=count-1
        end if
    end for
    $\mathbf{Q}(i) = m$+count
end for
Initialize # of reverse direction nodes $\mathbf{Qo} = \mathbf{0}$
$\mathbf{Qo}(0) = w$
for $i = 1 \to N$ do
    count=0
    for $j = 0 \to w - 1$ do
        if $\mathbf{fo}(j) = 1\&\&j(v_b + \mathbf{v}(j)) - (d_b + \mathbf{d}(j)) > 0$ then
            count=count-1
        end if
        if $\mathbf{f}(j) = 2\&\&j(v_b + \mathbf{v}(j)) - (d_b - \mathbf{d}(j)) > 0$ then
            count=count-1
        end if
    end for
    $\mathbf{Qo}(i) = w$+count
end for
total number of the source nodes $\mathbb{Q} = \mathbf{Q} + \mathbf{Qo}$
Output: # of source nodes within communication range
```

When the state of the sink is $S_R = M - \alpha_t - \beta_t + 1, M - \alpha_t - \beta_t + 2, \cdots, M - 1$, the $M - S_R$ source nodes will send $M - S_R$ valid data packets. The probability of sink node transits k states in time slot t is

$$P_{mov}(t,k) = C^k_{M-S_R}(1 - p_e)^k p_e^{M-S_R-k}. \tag{23}$$

The state matrix of the sink in the first time slot is

$$\begin{aligned} \mathbb{S}_1 &= [B_{n(1)}(0)\ B_{n(1)}(1) \cdots\ B_{n(1)}(n(1))] \quad (24)\\ &= [P^1_{0,0}\ P^1_{0,1} \cdots\ P^1_{0,n(1)}], \quad (25) \end{aligned}$$

where the binomial distribution $B_{n(t)}(\alpha)$ represents the probability of receiving α valid data packets out of the data packets sent by $n(t)$ source nodes in time slot t. Here, $n(t)$ represents the total number of source nodes in time slot t, which is initially $m + w$. $P_{i,j}^{k}$ denotes the probability of the sink state transitioning from i to j in the kth time slot. To determine the state matrix of the sink node after time slot t, denoted as $\mathbb{S}t$, we need to first solve for the state matrix after time slot $t - 1$. The state matrix after time slot t is denoted $\mathbb{S}t$ and is represented by Equation (26).

The probability $P_t(M)$, which represents the sink node receiving M valid data packets after t time slots, can be calculated by summing the probabilities $P_{i,M}^{t}$ over all possible states i in the state matrix $\mathbb{S}t$. Mathematically, this can be expressed as $P_t(M) = \sum P_{i,M}^{t}$. The solution for the state matrix $\mathbb{S}_t$ depends on the state matrix $\mathbb{S}_{t-1}$ from the previous time slot and the number of source nodes $\mathbb{Q}(t)$ derived from Algorithm 1. Algorithm 2 provides a solution for calculating the probability P_t, which represents the sink node being in different states after time slot t. This probability is dependent on the values of P_{t-1} and $\mathbb{Q}(t-1)$. Specifically, $P_t(M)$ represents the completion probability of time slot t, which is the probability that sink node receives M data packets after t time slots. The recursive relationship between P_t and P_{t-1} is expressed as $P_t = \Phi(P_{t-1}, B_{\mathbb{Q}(t-1)})$, where $B_{\mathbb{Q}(t-1)}$ represents the binomial distribution of the number of received packets in time slot t when $\mathbb{Q}(t-1)$ data packets are sent. Such a solution for the recursive relation is provided in Algorithm 3. In Algorithm 3, the first step is to determine the number of source nodes participating in each round of communication. If this is larger than the needed number of packets M, then only M nodes will participate in the communication. Otherwise, all the nodes will participate. For each sink state i, the probability distribution at time slot t is computed by calculating the probability of receiving $k = i - j$ messages correctly after having received j messages (see line 19 in Algorithm 3). By utilizing Algorithm 2 and Algorithm 3, we can calculate the completion probability $P_t(M)$ of the sink node after t time slots. Subsequently, we will conduct an analysis of the delay probability distribution, taking into consideration varying packet loss rates p_e and packet batch sizes M.

$$\mathbb{S}_t = \begin{bmatrix} B_{n(t)}(0)\sum P_{\eta,0}^{t-1} & B_{n(t)}(1)\sum P_{\eta,0}^{t-1} & B_{n(t)}(2)\sum P_{\eta,0}^{t-1} & \cdots & B_{n(t)}(n(t))\sum P_{\eta,0}^{t-1} \\ B_{n(t)}(0)\sum P_{\eta,1}^{t-1} & B_{n(t)}(1)\sum P_{\eta,1}^{t-1} & B_{n(t)}(2)\sum P_{\eta,1}^{t-1} & \cdots & B_{n(t)}(n(t))\sum P_{\eta,1}^{t-1} \\ \cdots & & & & \\ B_{n(t)}(0)\sum P_{\eta,M-2}^{t-1} & B_{n(t)}(1)\sum P_{\eta,M-2}^{t-1} & B_{n(t)}(2)\sum P_{\eta,M-2}^{t-1} & \cdots & 0 \\ B_{n(t)}(0)\sum P_{\eta,M-1}^{t-1} & B_{n(t)}(1)\sum P_{\eta,M-1}^{t-1} & 0 & \cdots & 0 \end{bmatrix}$$
$$= \begin{bmatrix} P_{0,0}^{t} & P_{0,1}^{t} & P_{0,2}^{t} & \cdots & P_{0,n}^{t} \\ P_{1,1}^{t} & P_{1,2}^{t} & P_{1,3}^{t} & \cdots & P_{1,1+n}^{t} \\ \cdots & & & & \\ P_{M-2,M-2}^{t} & P_{M-2,M-1}^{t} & P_{M-2,M}^{t} & \cdots & 0 \\ P_{M-1,M-1}^{t} & P_{M-1,M}^{t} & 0 & \cdots & 0 \end{bmatrix}. \tag{26}$$

Algorithm 2 Completion probability of sink at time slot t

```
Input: # of raw data packets $M$, packet loss rate $p_e$, communication times $N$, # of source
    nodes $\mathbb{Q}$
1: $B_n(i) = C_n^i(1 - p_e)^i p_e^{n-i}$
2: for $i = 0 \to \mathbb{Q}(0)$ do
3:     $P_1(i) = B_{\mathbb{Q}(0)}(i)$
4: end for
5: for $t = 2 \to N$ do
6:     $P_t = \Phi(P_{t-1}, B_{\mathbb{Q}_{(t-1)}})$
7: end for
Output: Completion probability of sink at time slot $t$: $P_t(M)$
```

Algorithm 3 The state distribution probability of sink after time slot t:$P_t = \Phi(P_{t-1}, B_{\mathbb{Q}(t)})$

```
Input: The distribution probability of sink state last time slot $P_{t-1}$,# of current source nodes $\mathbb{Q}(t-1)$
sumQ(t)=0
for $i = 0 \to t-1$ do
    sumQ(t)=sumQ(t)+$\mathbb{Q}(i)$
end for
if sumQ(t) $> M$ then
    maxNum = $M$
else
    maxNum=sumQ(t)
end if
sumQ(t-1)=0
for $i = 0 \to t-2$ do
    sumQ(t-1)=sumQ(t-1)+$\mathbb{Q}(i)$
end for
for $i = 0 \to$ maxNum do
    $P_t(i) = 0$
    for $j = 0 \to$ sumQ(t-1) do
        for $k = 0 \to \mathbb{Q}(t-1)$ do
            if $j + k = i\&\&j! = M$ then
                $P_t(i) = P_t(i) + P_t(j) \times B_{\mathbb{Q}(t-1)}(k)$
            end if
        end for
    end for
end for
Output: The state probability distribution $P_t$ of sink after time slot $t$
```

4. Simulation Performance

In this section, we present a comprehensive analysis of the performance of the proposed scheme through Matlab simulations. Our main focus was on minimizing the latency, which was quantified by the number of time slots required to complete the transmission. We paid particular attention to two key factors that impact the latency: the packet loss rate, and the batch size. In addition, we conducted an analysis on the impact of varying vehicle communication ranges and different arrival rates following a stochastic arrival process, the Poisson process. Furthermore, we compared the coding and non-coding schemes. By comparing their performance, we could evaluate the effectiveness of the coding scheme in reducing the latency and improving the overall efficiency of the system.

4.1. Packet Loss Rate

To analyze the completion probability distribution under varying packet loss rates, we set up the following parameters:

- The initial distance of the source node to the sink node was randomly generated within the range of 0 to 150 m. This was because the current communication range of intelligent cars is 150–300 m. We stipulated that the communication range of the sink node was 150 m in front and behind; that is, the sink node could communicate with vehicles within a distance of 150 m:
- The speed of each node was randomly assigned within a range of 60 to 120 km/h considering the highway scenario;
- The initial position of each car was randomly generated, either in front of or behind the sink node;
- Each time slot was set to a duration of 100 ms, resulting in 10 rounds of communication per second.

With these parameters in place, we calculated the completion probability of the sink node at time slot t, which represents the likelihood of receiving M packets after t time slots.

Figure 5 shows the results of the simulation conducted with varying packet loss rates of $m = 2$, $w = 3$, and $M = 100$. In Figure 5a, we can observe that as the packet loss rate increased, the sink node required more time to receive M data packets, resulting in a more dispersed probability distribution of completion delay. This was because of the decrease in the number of data packets received by the sink node during each round, as a result of the high packet loss rate.Additionally, as the time slots progress, more sources may move out of the sink's communication range, resulting in fewer sources participating in the communication process and increasing the delay. Figure 5b demonstrates the correlation between the packet loss rate and the slope of the cumulative completion probability distribution curve. As the packet loss rate decreases, the curve becomes steeper, indicating a higher probability of timely completion. This implies that a lower packet loss rate leads to more efficient and reliable completion of the transmission process.

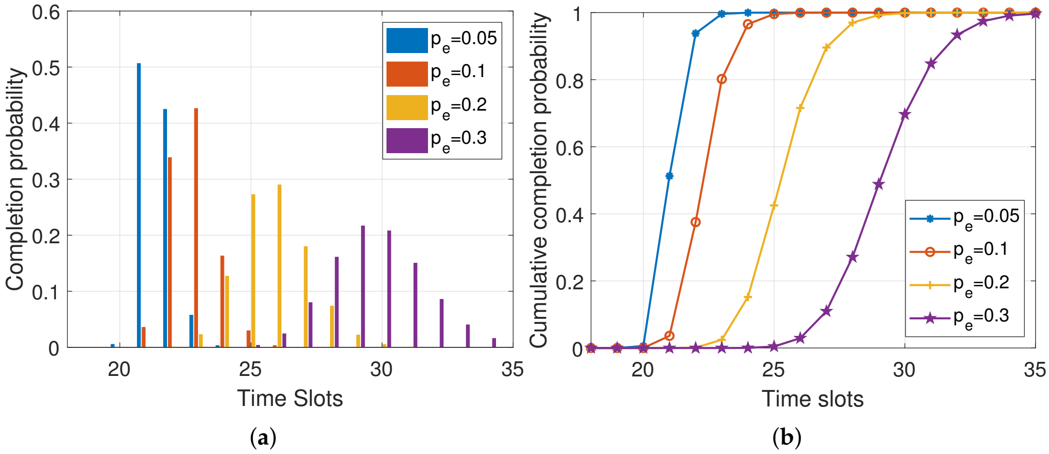

Figure 5. Completion probability distribution and cumulative completion probability distribution of the delay under different packet loss rates ($m = 2$, $w = 3$, $M = 100$). (**a**) Completion probability distribution; (**b**) cumulative completion probability distribution.

4.2. Batch Size

The delay also depends on the batch size. Figure 6 presents the simulation results under different batch sizes, with parameters set as $m = 3$, $w = 4$, $p_e = 0.1$. In Figure 6a, we can observe the completion probability distribution, and in Figure 6b, we can observe the cumulative distribution of the probability of delay. As the batch size increased, the number of time slots required to transmit a batch also increased. This resulted in a broader probability distribution of completion delay, indicating that larger batch sizes require more time to complete the transmission process. The increased delay was attributed to the larger number of data packets that had to be transmitted within each batch.

Table 2 presents the average delay and unit delay of the first batch of packets transmitted under varying batch sizes. It can be observed that as the batch size increased, the average delay consistently increased. However, contrary to the findings in the one-way lane scenario [18], the unit delay did not always decrease in the two-way lanes scenario. In the two-way lane scenario, after applying RLNC to M raw data packets and transmitting them to the sink node, the sink node needed to receive M valid data packets to perform decoding. Therefore, with larger batch sizes, more rounds of transmission were required to complete the transmission process, even with the same number of initial sources. Consequently, the duration of the communication process was prolonged, leading to a higher likelihood of source nodes moving out of the communication range of the sink node. Thus, the subsequent rounds witnessed a decrease in the number of sources and the number of packets received in each time slot. It is worth noting that for smaller network sizes, the communication delay became larger, which was due to the limited number of sources available for transmission.

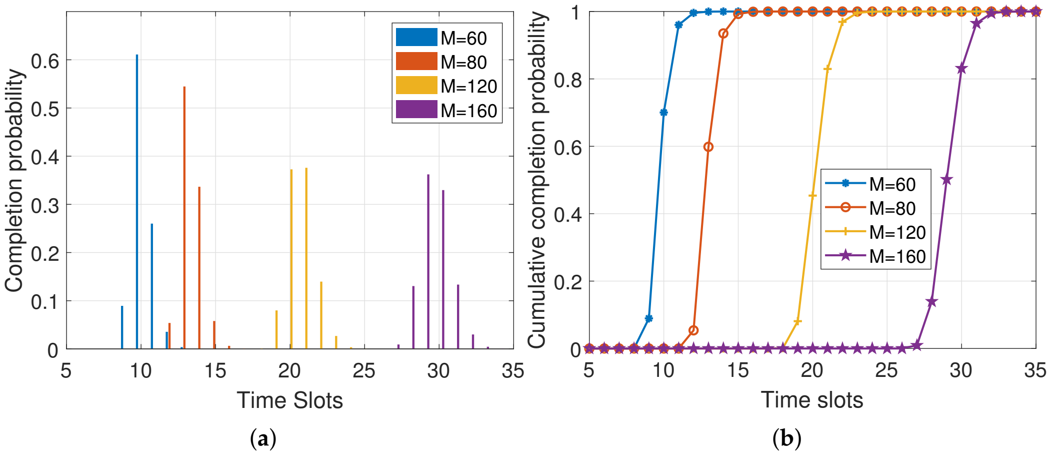

Figure 6. Completion probability distribution and cumulative completion probability distribution of delay under different batch size ($m = 3$, $w = 4$, $p_e = 0.1$). (**a**) Completion probability distribution; (**b**) cumulative completion probability distribution.

The dynamic nature of the network topology needs to be considered when analyzing the impact of batch size. Merely focusing on the average and unit delay for the first batch is not sufficient. This is because, as subsequent batches are sent, the number of source nodes changes over time, influencing the overall communication delay. Therefore, we analyzed the total delay for sending a total of Q data packets, while varying the batch size M, under the condition that the total quantity of data packets Q remained constant. Table 3 presents the cumulative delay incurred during the transmission of all packets, under varying batch sizes. It can be observed that, as the batch size increased, the total delay exhibited a decreasing trend. This was because, with a smaller batch size, it took more rounds of communication to send the subsequent batches, and the source nodes may have left the communication range, resulting in more time slots. This implies that increasing the batch size can mitigate communication delays and improve the overall efficiency of the transmission process.To reduce the communication delay in two-way lane V2V communication using RLNC, increasing the batch size within the storage and computing capabilities of the source and sink nodes can effectively minimize the communication delay.

Table 2. Unit delay under different batch sizes (in time slots).

$M(m = 3, w = 4, p_e = 0.1)$	E_T	$\overline{T}$
60	10.2545	0.1709
80	13.4206	0.1678
120	20.6697	0.1722
160	29.5586	0.1847

Table 3. Total delay under different batch sizes (in time slots).

$M(Q = 480, m = 3, w = 4, p_e = 0.1)$	E_T	$M(Q = 600, m = 4, w = 5, p_e = 0.1)$	E_T
60	258.0652	75	138.5758
80	255.6062	100	136.7833
120	252.2668	120	136.0340
160	250.0318	150	135.4871

4.3. Vehicle Range

Next, we examined the influence of varying the vehicle ranges on the time slots required for transmission, as illustrated in Figure 7. Figure 7a shows the completion probability distribution, and Figure 7b shows the cumulative completion probability distribution. It is evident that with a smaller communication range, more time slots were needed to complete the transmission. However, as the communication range increased, there was minimal impact on the transmission process. This was attributed to the fact that with a sufficiently large communication range, vehicles remained within the communication range until the transmission was complete.

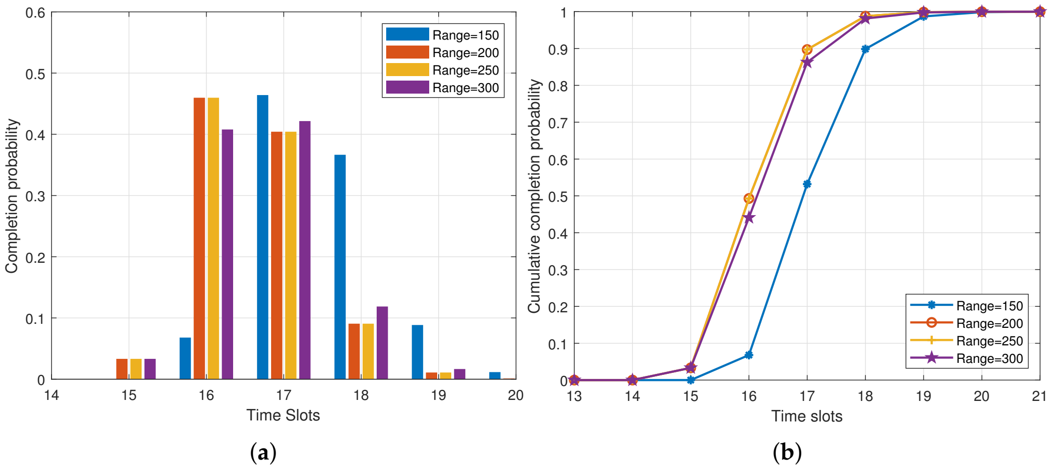

Figure 7. Completion probability distribution and cumulative completion probability distribution of delay under different range of vehicles ($m = 2$, $w = 3$, $p_e = 0.1$, $M = 100$). (**a**) Completion probability distribution; (**b**) cumulative completion probability distribution.

4.4. Poisson Arrival Process

The Poisson process approximates the stochastic arrival process of vehicles well. In the following analysis, we delved into the influence of varying arrival rates on the transmission, utilizing a Poisson process model to simulate vehicle arrivals. As depicted in the Figure 8, we initiated the network with three vehicles in the same direction and four in the opposite direction, assuming a packet loss rate of 0.1. λ represents the Poisson parameter, indicating the arrival rate of vehicles. For each task involving the transmission of 100 data packets, we determined the time slots required for 50 cases. At the beginning of the transmission, there was a period during which the model with $\lambda = 1$ (represented by blue triangles) required a significantly large number of time slots. This was because the vehicles transmitting initially gradually leave, but due to the relatively low arrival of vehicles, more time slots were needed to complete the transmission.

Notably, after multiple cases, we observed a stabilizing trend in the time required for each task. Once it reached a steady state, a higher vehicle arrival rate (signified by a larger λ) corresponded to a reduced number of time slots being required to complete a transmission. This was because, when there were more vehicles within communication range, more vehicles could participate in the transmission task, enabling faster reception of a sufficient number of decoded data packets.

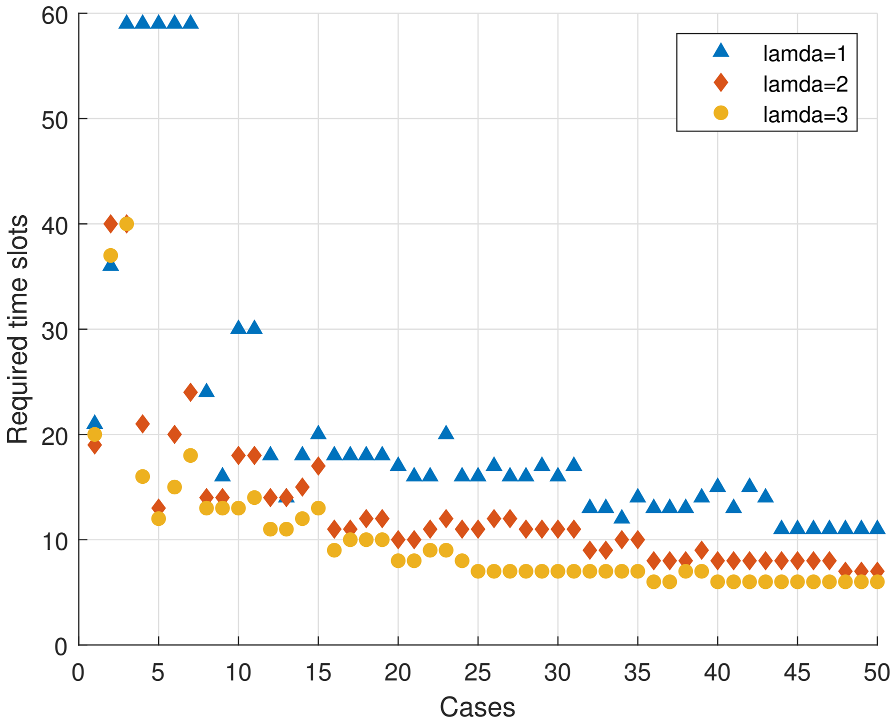

Figure 8. The time slots required with different λ in the Poisson process ($m = 2$, $w = 3$, $p_e = 0.1$, $M = 100$).

4.5. Coding vs. Non-Coding

By employing RLNC, M raw data packets were encoded in batches at the source nodes and transmitted to the sink node, which required the sink node to receive M valid data packets for decoding. Through multiple batches, the sink node could recover Q original data packets. Without coding, each source node randomly selected and sent one packet to the sink node in each round, until all packets had been received. However, this approach does not guarantee that the randomly selected data packets from different sources will be distinct, potentially leading to the sink node receiving duplicate data packets. Table 4 provides a comparison between the coding and non-coding schemes. It shows that as the packet loss rate increased, the communication latency also increased. However, it is worth noting that doubling the packet loss rate did not result in a significant increase in communication delay. The adoption of RLNC technology improved the communication robustness, enhancing the resistance against channel degradation and reducing communication delay.

In terms of RLNC coding overheads, this hinged on both the finite field size q and the number of original packets N involved in the coding process. Storing the coefficients in a single coded packet necessitated $(N-1) \cdot \log_2 q$ bits. It is evident that, with a small packet size, the performance was significantly hindered due to this substantial overhead. An approach involving the attachment of the seed of the random coefficients generator to the coded packets was employed in [14,15] for network coding, effectively reducing the overheads to $\log_2 q$, regardless of the number of combined packets. This idea, initially proposed in [25], is worth considering for adoption to reduce overheads in future research.

Table 4. Comparison of coding and non-coding schemes (in time slots).

Case	E_T $(Q = 400,\ M = 50,\ m = 2,\ w = 3)$	E_T $(Q = 600,\ M = 60,\ m = 4,\ w = 5)$
Coding $(p_e = 0.1)$	270.9284	180.3904
Coding $(p_e = 0.2)$	327.5296	225.0866
Coding $(p_e = 0.3)$	399.8219	282.0798
Non-coding $(p_e = 0)$	514.86	474.73

5. Conclusions

In this paper, we proposed an RLNC scheme for efficient and low-latency transmission of large-scale data in V2V communication with two-way lanes. We introduced a dynamic multi-source single-sink model specifically designed for the two-way lane V2V communication scenario and derived the lower bound of the generation probability for the RLNC scheme. Our analysis revealed that reducing the packet loss rate and increasing the sending batch size properly can effectively decrease the communication delay. By conducting a comparative analysis with a non-RLNC scheme, we demonstrated the superior performance of the RLNC scheme in reducing the communication delay and enhancing network robustness in V2V networks with a dynamic topology with two-way lanes.

Author Contributions: Conceptualization, T.Z. and Y.Z.; methodology, T.Z.; software, T.Z. and Y.Z.; validation, Y.Z., T.Z. and C.L.; formal analysis, Y.Z.; investigation, C.L.; writing—original draft preparation, Y.Z.; writing—review and editing, Y.Z. and C.L.; visualization, Y.Z.; supervision, C.L.; project administration, C.L.; funding acquisition, C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Science Foundation of China (NSFC) with grant Nos. 62271514 and the Science, Technology and Innovation Commission of Shenzhen Municipality with grant Nos. JCYJ20210324120002007, and ZDSYS20210623091807023.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Papadimitratos, P.; Fortelle, A.L.; Evenssen, K.; Brignolo, R.; Cosenza, S. Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation. *IEEE Commun. Mag.* **2009**, *47*, 84–95. [CrossRef]
2. Rios-Torres, J.; Malikopoulos, A.A. A Survey on the Coordination of Connected and Automated Vehicles at Intersections and Merging at Highway On-Ramps. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1066–1077. [CrossRef]
3. Abboud, K.; Omar, H.A.; Zhuang, W. Interworking of DSRC and Cellular Network Technologies for V2X Communications: A Survey. *IEEE Trans. Veh. Technol.* **2016**, *65*, 9457–9470. [CrossRef]
4. Kenney, J.B. Dedicated Short-Range Communications (DSRC) Standards in the United States. *Proc. IEEE* **2011**, *99*, 1162–1182. [CrossRef]
5. Popovski, P.; Nielsen, J.J.; Stefanovic, C.; de Carvalho, E.; Strom, E.; Trillingsgaard, K.F.; Bana, A.S.; Kim, D.M.; Kotaba, R.; Park, J.; et al. Wireless Access for Ultra-Reliable Low-Latency Communication: Principles and Building Blocks. *IEEE Netw.* **2018**, *32*, 16–23. [CrossRef]
6. Zhou, H.; Jiang, K.; He, S.; Min, G.; Wu, J. Distributed Deep Multi-Agent Reinforcement Learning for Cooperative Edge Caching in Internet-of-Vehicles. *IEEE Trans. Wirel. Commun.* **2023**, 1. [CrossRef]
7. Zhang, H.; Huang, M.; Zhou, H.; Wang, X.; Wang, N.; Long, K. Capacity Maximization in RIS-UAV Networks: A DDQN-Based Trajectory and Phase Shift Optimization Approach. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 2583–2591. [CrossRef]
8. Ahlswede, R.; Cai, N.; Li, S.Y.; Yeung, R. Network information flow. *IEEE Trans. Inf. Theory* **2000**, *46*, 1204–1216. [CrossRef]
9. Syed, A.A.; Ayaz, S.; Leinmüller, T.; Chandra, M. Network Coding Based Fault-Tolerant Dynamic Scheduling and Routing for In-Vehicle Networks. *J. Netw. Syst. Manag.* **2023**, *31*. [CrossRef]
10. Sun, Y.; Yin, L.; Ma, Y.; Wang, C. IoV-SDCM: An IoV Secure Data Communication Model Based on Network Encoding and Relay Collaboration. *Secur. Commun. Netw.* **2022**, *2022*, 6546004. [CrossRef]
11. Ye, F.; Roy, S.; Wang, H. Efficient Data Dissemination in Vehicular Ad Hoc Networks. *IEEE J. Sel. Areas Commun.* **2012**, *30*, 769–779. [CrossRef]
12. Liu, F.; Chen, Z.; Xia, B. Data Dissemination With Network Coding in Two-Way Vehicle-to-Vehicle Networks. *IEEE Trans. Veh. Technol.* **2016**, *65*, 2445–2456. [CrossRef]
13. Park, J.S.; Lee, U.; Oh, S.; Gerla, M.; Lun, D.; Ro, W.W.; Park, J. Delay Analysis of Car-to-Car Reliable Data Delivery Strategies Based on Data Mulling with Network Coding. *IEICE Trans. Inf. Syst.* **2008**, *E91-D*, 2524–2527. [CrossRef]
14. Hassanabadi, B.; Valaee, S. Reliable Periodic Safety Message Broadcasting in VANETs Using Network Coding. *IEEE Trans. Wirel. Commun.* **2014**, *13*, 1284–1297. [CrossRef]
15. Gao, Y.; Ali, G.G.M.N.; Chong, P.H.J.; Guan, Y.L. Network Coding Based BSM Broadcasting at Road Intersection in V2V Communication. In Proceedings of the 2016 IEEE 84th Vehicular Technology Conference (VTC-Fall), Montreal, QC, Canada, 18–21 September 2016; pp. 1–5. [CrossRef]

16. Gao, Y.; Chong, P.H.J.; Guan, Y.L. BSM dissemination with network coded relaying in VANETs at NLOS intersections. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–6. [CrossRef]
17. Ho, T.; Medard, M.; Koetter, R.; Karger, D.; Effros, M.; Shi, J.; Leong, B. A Random Linear Network Coding Approach to Multicast. *IEEE Trans. Inf. Theory* **2006**, *52*, 4413–4430. [CrossRef]
18. Zhu, T.; Li, C.; Tang, Y.; Luo, Z. On latency reductions in vehicle-to-vehicle networks by random linear network coding. *China Commun.* **2021**, *18*, 24–38. [CrossRef]
19. Li, S.Y.; Yeung, R.; Cai, N. Linear network coding. *IEEE Trans. Inf. Theory* **2003**, *49*, 371–381. [CrossRef]
20. Ho, T.; Koetter, R.; Medard, M.; Karger, D.; Effros, M. The benefits of coding over routing in a randomized setting. In Proceedings of the IEEE International Symposium on Information Theory, 2003. Proceedings, Yokohama, Japan, 29 June 2003–4 July 2003. [CrossRef]
21. Pan, S.; Zhang, X.M. Cooperative Gigabit Content Distribution with Network Coding for mmWave Vehicular Networks. *IEEE Trans. Mob. Comput.* **2023**. [CrossRef]
22. Tasdemir, E.; Lehmann, C.; Nophut, D.; Gabriel, F.; Fitzek, F.H.P. Vehicle Platooning: Sliding Window RLNC for Low Latency and High Resilience. In Proceedings of the 2020 IEEE International Conference on Communications Workshops (ICC Workshops), Dublin, Ireland, 7–11 June 2020. [CrossRef]
23. Zhang, M.; Chong, P.H.J.; Seet, B.C.; Rehman, S.U.; Kumar, A. Integrating PNC and RLNC for BSM dissemination in VANETs. In Proceedings of the 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, Canada, 8–13 October 2017; pp. 1–5. [CrossRef]
24. Zhang, M.; Ali, G.G.M.N.; Chong, P.H.J.; Seet, B.C.; Kumar, A. A Novel Hybrid MAC Protocol for Basic Safety Message Broadcasting in Vehicular Networks. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 4269–4282. [CrossRef]
25. Liu, Z.; Wu, C.; Li, B.; Zhao, S. UUSee: Large-Scale Operational On-Demand Streaming with Random Network Coding. In Proceedings of the 2010 Proceedings IEEE INFOCOM, San Diego, CA, USA, 14–19 March 2010; pp. 1–9. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.