Visible Light Two-Way Communication Method for Vehicle-Road Collaboration

Caipeng Gu[1], Jijing Cai[1], Meilei Lv[2], Jiefan Qiu[3], Chenzhuo Jin[4], and Kai Fang[5](✉)

[1] Hangzhou Dianzi University, Hangzhou, China
[2] Quzhou University, Quzhou, China
[3] Zhejiang University of Technology, Hangzhou, China
qiujiefan@zjut.edu.cn
[4] Anhui University, Hefei, China
A02114019@stu.ahu.edu.cn
[5] Zhejiang Agricultural and Forestry University, Hangzhou, China
kaifang@ieee.org

Abstract. With the increasing placement of sensor nodes on vehicles and Internet of Vehicles equipment, the need for online debugging of sensor nodes has grown significantly. However, current methods for online debugging of sensors heavily rely on the existing network infrastructure. In the event of communication infrastructure failure, retrieving and repairing transmission information becomes nearly impossible. In our research, we've employed the optical modules that accompany sensor nodes to establish a hybrid communication debugging system based on visible light communication (VLC). To enhance the efficiency of debugging information uploads, this article organizes and optimizes the original data, utilizing the Snappy compression algorithm to minimize empty time slots and achieve data compression during source encoding, thereby saving time. In addition, to bolster the reliability of data uploads, we've developed a frame synchronization mechanism tailored for the optical camera at the receiving end of the uplink. By employing the Transformer algorithm for frame header position prediction, we've improved the reliability of data collection.

Keywords: IoV · Visible light communication · Snappy compression · Frame synchronization · Transformer

1 Introduction

To build a more intelligent transportation system, promote the development of the transportation industry, and reduce traffic congestion, various countries are paying more and more attention to the application of the Internet of Vehicles in the transportation field. Communication technology, as the key to the Internet of Vehicles, has also developed faster and faster. Communication technology can

V. C. M. Leung et al. (Eds.): Qshine 2023, LNICST 574, pp. 484–494, 2024.
https://doi.org/10.1007/978-3-031-65123-6_36

make information such as road conditions, vehicles, and roads communicated intuitively, effectively, and clearly, achieving high-quality and low-latency effects. However, during the movement of vehicles, it is inevitable to encounter traffic jams, signal interruptions, and even accidents, which will lead to changes in the movement of vehicles. These changes will also cause unreliability and inefficiency of mobile Internet of Vehicles communications. To tackle this issue, these papers suggest a new communication technology, which uses visible light communication (VLC) [1–3] to convey information in the mobile Internet of Vehicles.

Nowadays, most vehicles are equipped with light-emitting components such as LEDs, and most Internet of Vehicles devices are equipped with ambient light sensors due to sensing tasks, which meet the basic needs of VLC equipment. This makes it possible for ordinary car networking equipment to use VLC to replace wireless communication. Once a vehicle encounters an accident while moving, VLC can be used as a supplementary communication method to wireless communication for emergency purposes. It can upload some road conditions and vehicle conditions promptly to help other vehicles receive accurate information promptly, further improving the vehicle's safety while driving overall security. Abuella H et al. proposed a hybrid debugging framework (RF/VLC) system that uses VLC's low-latency reliable communication to increase user speed and mobility, thereby optimizing the capacity and power consumption of the entire network [4]. In a VLC system based on high-order CAP modulation, Wang Y et al. achieved an aggregate data rate of 8Gb/s in one-meter indoor free-space transmission, which is the highest data rate reported so far in a VLC system. Kim Y H et al. proposed vehicle-to-vehicle (V2V) communication using VLC technology under fog conditions, which can effectively offset the damage caused by fog and has a relatively high SNR [5]. Goto Y et al. used an optical communication image sensor (OCI) and implemented 55 MBps VLC signal transmission with a faster data rate called dedicated short-range communication (DSRC) [6]. Lu I C et al. proposed and demonstrated the use of energy-saving and low-cost 682 nm VCSEL with 1 GHz modulation bandwidth to implement high-speed wireless VLC, which is regarded as an economical, energy-saving and effective communication method [7].

Therefore, this paper will conduct resource optimization research on visible light communication for vehicle-road collaboration on this basis, and establish a new mobile vehicle networking communication system framework.

The main contributions of this paper are summarized as follows:

(1) This article analyzed the transmitted debugging code and compressed it based on the characteristics of the code. Using the snappy compression method, greatly improved the transmission efficiency and reduced the volume of transmitted information.
(2) This paper presents a novel approach for enhancing the reliability of data uploads by self-powered sensor nodes through frame header prediction using a Transformer's robust reasoning capabilities.

2 System Framework

To verify the feasibility of using VLC to debug Internet of Vehicles equipment, this article builds a vehicle-road collaborative two-way communication debugging system based on visible light. The uplink and downlinks of this system are both implemented using VLC at the physical layer, but the implementation methods are different. This chapter will provide an overview of the system built. Starting from the system architecture, the overall workflow of the system is described, and the technologies and VLC modules used in debugging the uplink and downlink of the system are introduced respectively.

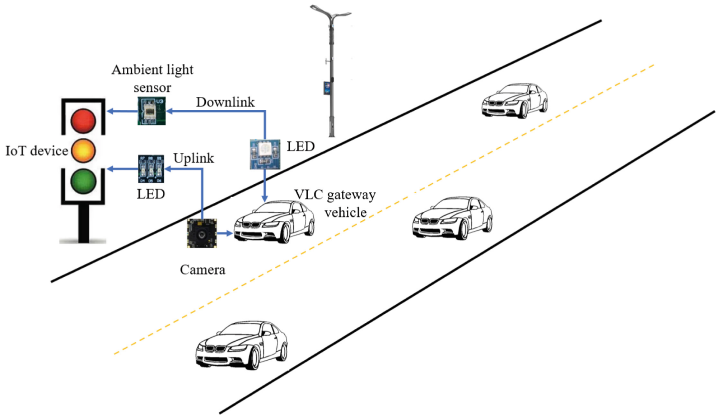

Fig. 1. System framework diagram

In the vehicle-road collaboration system, the device used as the VLC gateway is generally a vehicle equipped with general-purpose LEDs and embedded cameras. The overall system framework is shown in Fig. 1. The system has two visible light communication links, namely uplink and downlink. Since different links use different VLC technologies, the transmitters and receivers used are also different. In the uplink, the information of the IoV device is sent by its equipped LED module and received by the embedded camera module of the VLC gateway. The transmitter of the downlink is the LED module equipped with the VLC gateway, and the receiver is the ambient light sensor of the Internet of Vehicles device.

3 Our Method

This chapter introduces the algorithms used respectively to improve the rate of code transmission in the downlink of the VLC system and to improve the reliability of frame header prediction in the uplink. The overall structure diagram of the two-way communication of the vehicle-road collaboration system is shown in Fig. 2.

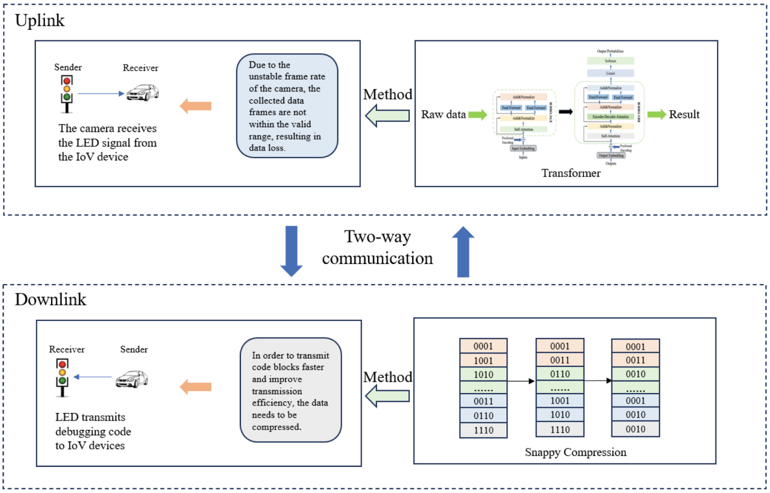

Fig. 2. Overall framework of two-way communication system

3.1 Modulation Algorithm Based on Snappy Compression

What is mainly transmitted in the downlink of the system is debugging code, which is usually a compiled code block. The code block has a large amount of binary data. Under the limited rate of the downlink, to transmit the code block faster, the data needs to be compressed. Traditional compression algorithms include classic algorithms such as Huffman and Zip [8]. Although the Huffman compression algorithm has a high compression ratio, additional data needs to be stored locally on the Internet of Vehicles device, and Zip compression has a large computational overhead and high device consumption. These algorithms are not friendly to resource-constrained IoV devices [9]. To make the built system suitable for Internet of Vehicles devices with limited resources, we need to explore a simple, low-resource consumption and high compression ratio compression method to compress code blocks.

To improve the transmission rate of the downlink of the vehicle-road cooperative system, this paper optimizes VLC from the aspect of source coding. In source coding, the Snappy method is used to compress the transmitted data, reduce the volume of transmitted information, and increase the transmission speed.

The goal of Snappy compression is very high speed and reasonable compression. For example, compared to Zip's fastest mode, Snappy is an order of magnitude faster for most inputs. Snappy compression is a dictionary in data sliding window compression algorithm, which means that it does not store the dictionary separately from the compressed data stream, but instead uses back-references to indicate repeated sequences of data. This algorithm was chosen due to its broad use and portability [10]. An open-source C++ implementation of Google's Snappy algorithm is used to guide the development of the hardware platform.

Unlike other compression algorithms like Huffman coding, Snappy doesn't rely on bit-based compression in its process and doesn't involve entropy values. In Snappy, the initial byte in the byte stream denotes the length of the uncompressed data, stored in little-endian format.

Flexible length coding, according to coding theory, maps input data or symbols into a varying number of bits. This approach allows for data compression and decompression without any loss, ensuring lossless data compression. In the fast algorithm, 2 bits are used for the element type in the compression process, as opposed to the primary 4-bit byte tag that represents uncompressed data. The Snappy algorithm's data structure, as illustrated in Fig. 3, includes a representation where "00" signifies text elements, representing uncondensed data, with the inaugural 6 bits allocated for data length storage. [11].

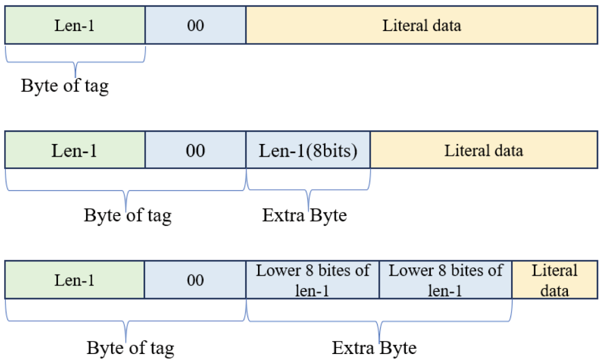

Fig. 3. Data structure of snappy algorithm

3.2 Frame Header Prediction Based on Transformer Algorithm

The Image is divided into three sections based on the ERA (Effective Reception Area) as depicted in Fig. 4. ERA represents the portion of the image containing valid data, where this data is encoded as alternating light and dark stripes. The receiver utilizes ERA to demodulate and retrieve the original data. However, because the camera's frame rate is not consistent, the data frame might initiate before or conclude after ERA. If any part of the frame header extends beyond the acceptable range within ERA, either situation can lead to data loss.

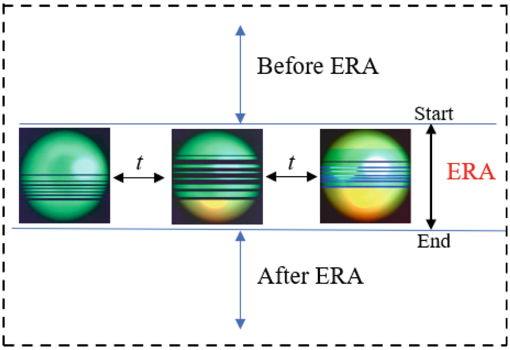

Fig. 4. ERA framework diagram

Through a set of experiments, we identified a pattern in the time-related movement of the frame header position. This led us to frame header prediction, which can be effectively framed as a time series prediction problem. Consequently, we explored the application of the Transformer algorithm to predict the frame header position.

Transformer is essentially an encoder-decoder architecture. The overall framework is shown in Fig. 5.

In the Transformer model, the Self-Attention mechanism replaces the traditional RNN network commonly used in NLP tasks. Its main advantage over RNNs is the ability to perform parallel computations. In the Transformer, each encoder consists of two sub-layers: Self-Attention and Position-wise Feed Forward Network (FFN). These encoders share the same structure but use different weight parameters. Every sub-layer (Self-Attention and FFN) within each encoder includes a residual connection, followed by layer normalization. The overall computation process can be summarized as follows:

$$sub\_layer\_output = LayerNorm(x + SubLayer(x)) \tag{1}$$

The encoder input initially passes through the Self-Attention layer.

The encoder uses the Self-Attention mechanism to integrate date from other words in the input sentence when encoding a particular word. Later on, we'll explore the inner workings of Self-Attention. The Self-Attention layer's output is then directed to the feedforward network.

The decoder also incorporates these two layers from the encoder. However, there's an additional attention layer, known as Encoder-Decoder Attention, positioned between them. This layer assists the decoder in concentrating on pertinent sections of the input sentence. To support residual connections, it's essential to ensure that the output dimensions of all sub-layers and embedding layers in both the encoder and decoder remain consistent.

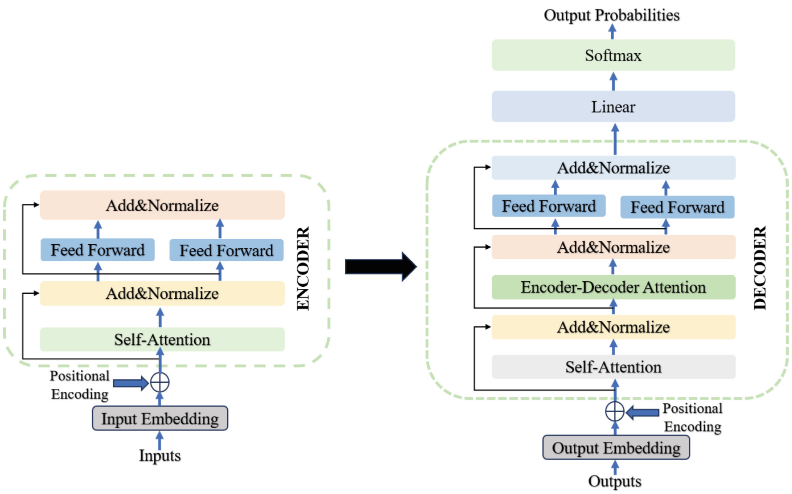

Fig. 5. Transformer architecture diagram

To enhance upload reliability, we've introduced a Transformer-based frame header prediction scheme. Within this solution, we've developed a real-time analysis segment to determine the frame header position. The segment retains the header positions from five successive images as a dataset for the subsequent frame header prediction. Subsequently, this stored data is fed as input to the Transformer model, which then produces the predicted frame header position. If any of these predictions exceeds 30% of the ERA, it indicates that the complete frame data cannot be received, triggering an immediate resynchronization process.

4 Experimental Analysis

In our experiments, we utilized a Linux-based system along with standard sensor nodes. The smart vehicle was equipped with an optical camera as the receiver, and a Raspberry Pi-based Internet of Vehicles device served as the emitter. The LED signal, affected by the rolling shutter mechanism, could be acquired by the camera, resulting in alternating light and shadow lines within the ERA. The smart vehicle efficiently demodulated the ERA for uplink communication. Additionally, Jeston Nano device acted as the computational center for the smart vehicle, executing the Transformer model. The hardware setup for this experiment is depicted in Fig. 6.

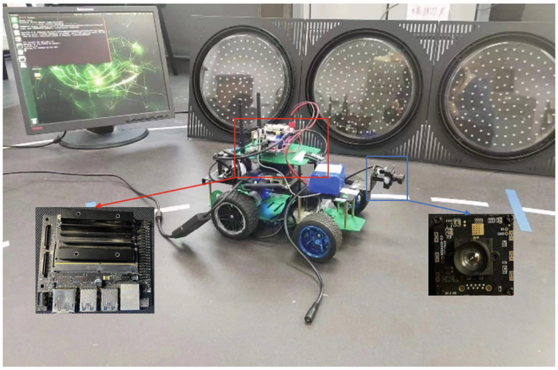

Fig. 6. Experimental hardware configuration

4.1 Performance Analysis Based on Snappy Compression

To test the performance of the snappy compression method, this article performs compression tests on five different code blocks and analyzes the experimental results. These commonly used code blocks for IoT devices are stored as binary files, including the control function Timer that controls the timer of the IoT device and the ambient light sensor control function LS-Ctrl. There are also classic algorithms such as the bubble sort function, search The Dijkstra function of the shortest path, and the Horspool function of the character matching function. This experiment uses the Snappy-based compression method to compress these code blocks and uses compression time to measure performance. The results are shown in Table 1.

Table 1. Compressed information

Compression method	Function name	Original file size (bit)	Compressed size (bit)	Compression time (s)
Snappy	Timer Fun	1170	352	0.00029
	Bubble Sort Fun	266	128	0.000143
	Dijkstra Fun	2195	758	0.00029
	Ls-Ctrl Fun	2125	608	0.000276
	Horspool Fun	1567	571	0.000309
Dictionary	Timer Fun	1170	764	0.47
	Bubble Sort Fun	266	209	0.21
	Dijkstra Fun	2195	1996	0.76
	Ls-Ctrl Fun	2125	1867	0.84
	Horspool Fun	1567	1443	0.56

The table also records the compressed file size and compression time of each code block. It can be seen from the table that the files compressed by Snappy are significantly smaller than those compressed by traditional dictionaries, and the compression time is significantly improved.

Simultaneously, this paper also employs the traditional Huffman compression method to compress individual code blocks and contrasts it with the compression approach introduced in this article. The results are depicted in Fig. 7.

The results demonstrate that the Snappy compression method employed in this article outperforms the Huffman compression method in terms of code compression. Furthermore, the Huffman compression method necessitates the maintenance of a dictionary on the Internet of Vehicles device, which consumes valuable memory space. In cases where the Internet of Vehicles device lacks a specific dictionary, additional dictionaries must be transmitted through the downlink, thereby diminishing transmission efficiency.

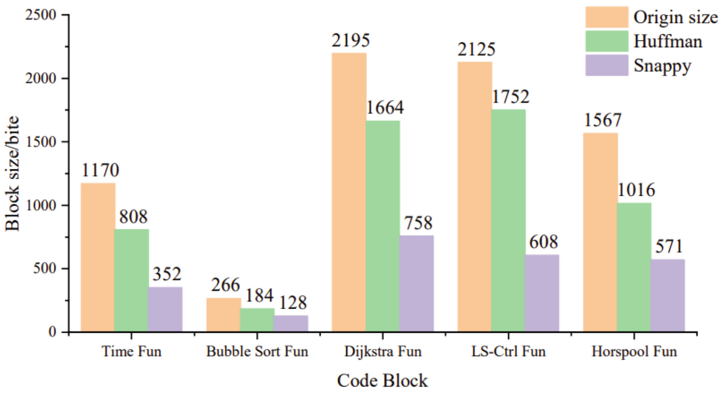

Fig. 7. Comparison of code size after compression

4.2 Transformer-Based Frame Header Prediction Performance Analysis

We selected a sample of 100 consecutive frames to determine their respective frame header positions. As depicted in Fig. 8, our Transformer-based frame header prediction system is primarily utilized when there is a loss of data frames. It's evident that the actual frame header position initially diverges from the predicted value. This discrepancy is attributed to the inherent instability of the embedded camera's frame rate, causing a shift in the position of the LED-generated light and shadow lines. Furthermore, variations in the camera's frame rate during the sampling process can result in synchronization issues between the transmitter and receiver. When the camera's frame rate changes, the sampling interval on the receiving end is affected, leading to incorrect signal sampling, transmission errors, and a rapid decline in the frame header position value, particularly when it approaches a threshold.To ensure the smooth upload of debugging information, we proactively predict and synchronize the frame header, anticipating situations where it may exceed the threshold. However, in cases like frame 45, which are challenging to predict, we can only discard such frames to handle exceptional situations.

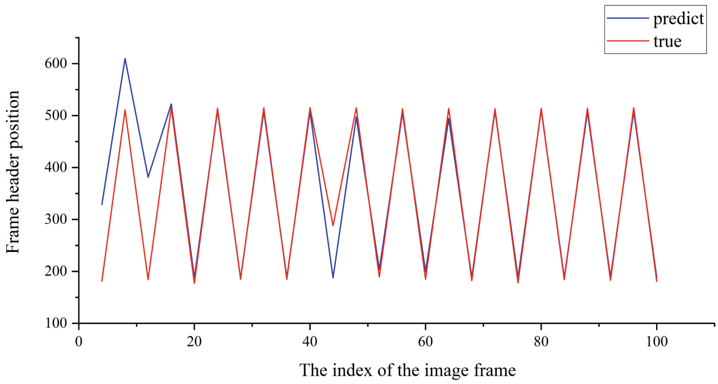

Fig. 8. Partial frame header position prediction scheme based on Transformer

5 Conclusion

In a mobile Internet of Vehicles system, when the Internet of Vehicles equipment faces irreparable or complete disconnection, visible light communication can serve as a supplementary communication method to wireless communication. It allows for instant uploading of debugging information to aid developers in analyzing the causes of equipment failure or disconnection, thus enhancing network security.This paper introduces a vehicle-road collaborative information debugging system based on VLC. The system comprises two links, the upper and lower links, facilitating bidirectional communication between them. In the uplink, we've implemented a frame header prediction scheme, harnessing the robust reasoning capabilities of the Transformer model to predict the frame header's position. This approach improves data collection accuracy and enhances the reliability of data uploads by sensor nodes.In the downlink, we've optimized compression techniques based on the transmitted code's characteristics, utilizing the Snappy compression method to significantly enhance transmission efficiency and reduce the volume of transmitted information. Experimental results demonstrate that these enhancements lead to excellent efficiency and reliability in improving upload performance.

References

1. Obeed, M., Dahrouj, H., Salhab, A.M., Zummo, S.A., Alouini, M.S.: User pairing, link selection, and power allocation for cooperative NOMA hybrid VLC/RF systems. IEEE Trans. Wirel. Commun. **20**(3), 1785–1800 (2021). https://doi.org/10.1109/twc.2020.3036444
2. Cang, Y., et al.: Joint deployment and resource management for vlc-enabled riss-assisted uav networks. IEEE Trans. Wirel. Commun. **22**(2), 746–760 (2023). https://doi.org/10.1109/twc.2022.3165853
3. Wang, F., Yang, F., Pan, C., Song, J., Han, Z.: Hybrid vlc-rf systems with multi-users for achievable rate and energy efficiency maximization. IEEE Trans. Wirel. Commun. **22**(9), 6157–6170 (2023). https://doi.org/10.1109/twc.2023.3239840
4. Abuella, H., et al.: Hybrid rf/vlc systems: a comprehensive survey on network topologies, performance analyses, applications, and future directions. IEEE Access **9**, 160402–160436 (2021). https://doi.org/10.1109/access.2021.3129154
5. Kim, Y.H., Cahyadi, W.A., Chung, Y.H.: Experimental demonstration of vlc-based vehicle-to-vehicle communications under fog conditions. IEEE Photonics J. **7**(6), 1–9 (2015). https://doi.org/10.1109/jphot.2015.2499542
6. Wang, Y., Tao, L., Huang, X., Shi, J., Chi, N.: Enhanced performance of a high-speed wdm cap64 vlc system employing volterra series-based nonlinear equalizer. IEEE Photonics J. **7**(3), 1–7 (2015). https://doi.org/10.1109/jphot.2015.2436911
7. Goto, Y., et al.: A new automotive vlc system using optical communication image sensor. IEEE Photonics J. **8**(3), 1–17 (2016). https://doi.org/10.1109/jphot.2016.2555582
8. Pan, G., Ye, J., Ding, Z.: On secure vlc systems with spatially random terminals. IEEE Commun. Lett. **21**(3), 492–495 (2017). https://doi.org/10.1109/lcomm.2016.2643632
9. Hammadi, A., Muhaidat, S., Sofotasios, P., Al-Qutayri, M., Elgala, H.: Non-orthogonal multiple access for hybrid vlc-rf networks with imperfect channel state information. Inf. Theory (2020)
10. Naser, S., et al.: Rate-splitting multiple access: unifying noma and sdma in miso vlc channels. IEEE Open J. Veh. Technol. **1**, 393–413 (2020). https://doi.org/10.1109/ojvt.2020.3031656
11. Doe, D., et al.: DSORL: data source optimization with reinforcement learning scheme for vehicular named data networks (2023)