Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.038305

Article

Redundant Transmission Control Algorithm for Information-Centric Vehicular IoT Networks

Abdur Rashid Sangi[1,*], Satish Anamalamudi[2], Mohammed S. Alkatheiri[3], Murali Krishna Enduri[2], Anil Carie[2] and Mohammed A. Alqarni[4]

[1]Department of Computer Science, College of Science and Technology, Wenzhou-Kean University, Wenzhou, 325060, China

[2]Department of Computer Science Engineering, SRM University AP, Amaravati, 522502, India

[3]Department of Cybersecurity, College of Computer Science & Engineering, University of Jeddah, Jeddah, Saudi Arabia

[4]Department of Software Engineering, College of Computer Science & Engineering, University of Jeddah, Jeddah, Saudi Arabia

*Corresponding Author: Abdur Rashid Sangi. Email: sangi_bahrian@yahoo.com

Received: 07 December 2022; Accepted: 13 June 2023; Published: 30 August 2023

Abstract: Vehicular Adhoc Networks (VANETs) enable vehicles to act as mobile nodes that can fetch, share, and disseminate information about vehicle safety, emergency events, warning messages, and passenger infotainment. However, the continuous dissemination of information from vehicles and their one-hop neighbor nodes, Road Side Units (RSUs), and VANET infrastructures can lead to performance degradation of VANETs in the existing host-centric IP-based network. Therefore, Information Centric Networks (ICN) are being explored as an alternative architecture for vehicular communication to achieve robust content distribution in highly mobile, dynamic, and error-prone domains. In ICN-based Vehicular-IoT networks, consumer mobility is implicitly supported, but producer mobility may result in redundant data transmission and caching inefficiency at intermediate vehicular nodes. This paper proposes an efficient redundant transmission control algorithm based on network coding to reduce data redundancy and accelerate the efficiency of information dissemination. The proposed protocol, called Network Cording Multiple Solutions Scheduling (NCMSS), is receiver-driven collaborative scheduling between requesters and information sources that uses a global parameter expectation deadline to effectively manage the transmission of encoded data packets and control the selection of information sources. Experimental results for the proposed NCMSS protocol is demonstrated to analyze the performance of ICN-vehicular-IoT networks in terms of caching, data retrieval delay, and end-to-end application throughput. The end-to-end throughput in proposed NCMSS is 22% higher (for 1024 byte data) than existing solutions whereas delay in NCMSS is reduced by 5% in comparison with existing solutions.

Keywords: Caching; data dissemination; redundancy control; ICN-vehicular IoT networks

1 Introduction

Researchers, academic experts, and industries have been exploring the research of Intelligent Automated Transportation Systems (ITS) due to the rapid advancements in wireless communication and automotive electronics. The Internet of Vehicles, a special application scenario of the mobile ad-hoc network (MANET), will have broad development in the future with ITS. Equipped with communication transmission modules, intelligent vehicles with sensing, computing, storage, and positioning capabilities can form a vehicular ad-hoc network (VANET) in the Internet of Vehicles (basic system shown in Fig. 1). Dynamic vehicle nodes can communicate not only with other vehicle nodes (Vehicle-to-Vehicle, V2V) but also with roadside wireless facilities (Vehicle-to-Infrastructure, V2I) [1]. Vehicular nodes in VANET continuously collect and perceive information about the surrounding environment, enabling interconnection and intercommunication of information that benefits the development of smart cities and society, including reducing traffic collisions, efficient traffic management, and improving quality of life [2]. The Internet of Vehicles (See Fig. 1) offers consumers a variety of multi-functional application services, mainly divided into two categories: traffic safety (collision avoidance) and path management [3]. However, the reliable transmission of data and information in the Internet of Vehicles is a critical research issue [4], due to the high mobility of vehicles and the instability of wireless transmission [5]. Traditional Transmission Control Protocol (TCP) is unrealistic in VANET since it requires frequent confirmation with each segment to maintain the transmission sequence for reliable transmission. Furthermore, TCP/IP establishes a communication link between the source and destination before data transmission, limiting communication to a predefined link [6]. This leads to decreased communication efficiency when the quality of the link channel decreases. At this point, the IP address associated with the network segment is used to identify the target host in the network [7]. The Internet of Vehicles (IoV) faces challenges in achieving reliable and efficient data transmission due to the high mobility of vehicular nodes, resulting in frequent disconnections of established connections between source and destination. To address this, researchers have turned their focus to the Information Centric Network (ICN) architecture, which prioritizes content over content providers and uses location-independent routing methods. By relying on content names instead of host identifiers, ICN can better accommodate the mobility of vehicular nodes. ICN's content name routing, caching, and multi-source, embedded multicast methods offer many advantages for efficient information distribution in the IoV. Moreover, the combination of ICN and the Internet of Things (IoT) can further enhance the performance, security, and privacy of connected systems. The use of redundant transmission control algorithms in Information-Centric Vehicular IoT networks improves the reliability and efficiency of data transmission by using multiple paths and enabling flexible data routing to find the fastest and most reliable path. Overall, the ICN architecture is more suitable for the IoV, and the combination of ICN and IoT offers new services and applications for efficient data management and access in connected systems. In Information Centric Vehicular IoT networks, network coding plays a significant role in improving information dissemination efficiency. This is achieved through the combination of multiple data streams into a single transmission, reducing the overhead of transmitting identical information multiple times. As a result, network coding can reduce latency, improve reliability, and enhance overall network performance in dynamic and unpredictable vehicular networks. Furthermore, network coding helps conserve network resources like bandwidth and energy by reducing redundant data transmission.

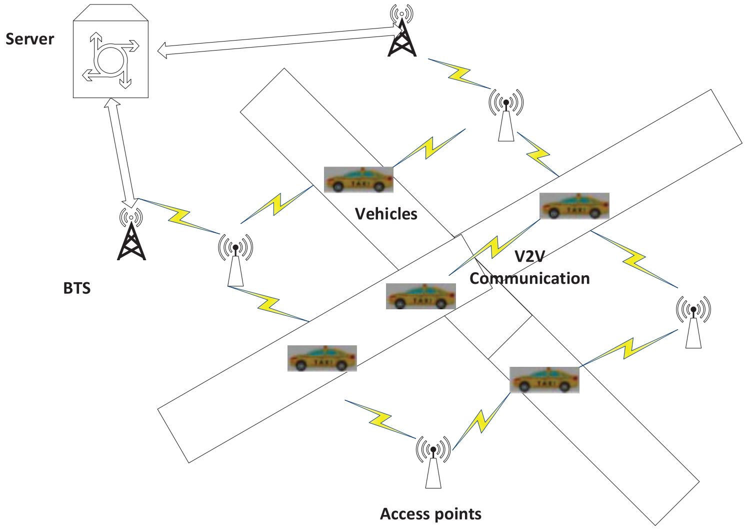

Figure 1: Vehicle-to-vehicle communication through vehicular IoT networks

Transmission control algorithms are essential in Information-Centric Internet of Things (IoT) networks to ensure that network resources, such as bandwidth and energy, are utilized efficiently. In addition, it can prioritize critical data packets and adjust the transmission rate based on network conditions to prevent network congestion and ensure the timely delivery of data. Moreover, transmission control algorithms can help to mitigate the impact of packet loss and errors in data transmission. Overall, transmission control algorithms are critical for improving the efficiency, reliability, and security of Information Centric IoT networks. 5G networks will be able to support a greater number of IoT devices than previous generations of wireless networks, which is critical as the number of IoT devices continues to grow. The combination of 5G and IoT is expected to drive innovation and create new opportunities across a range of industries, from healthcare to manufacturing to transportation [8,9].

The remainder of this paper is structured as follows: Section 2 provides a brief overview of related work, highlighting the pros and cons of existing ICN-based VANETs in terms of performance degradation caused by packet drops resulting from producer node mobility. In Section 3, we introduce a proposed redundant transmission control protocol to improve the performance of ICN-based Internet of vehicles. Section 4 presents experimental results and compares performance metrics with various existing scenarios. Finally, Section 5 concludes the paper and outlines directions for future research.

2 Related Works

The Internet of Vehicles faces challenges related to the frequent changes in network topology due to the high mobility of vehicle nodes. As vehicles enter and exit the network frequently, the network topology changes accordingly, which can be affected by drivers' behavior [10]. These changes in network topology, combined with the unstable wireless environment of the Internet of Vehicles, significantly reduce the link survival time between communication nodes [11]. Additionally, vehicle nodes have energy and battery power limitations, which can affect node energy consumption. However, vehicle nodes also have sufficient space to load more sensor equipment, providing stronger perception and information acquisition capabilities. Moreover, vehicle nodes have computing capabilities, including storage and processing, which enable complex computing applications in the Internet of Vehicles [12,13]. The quality of the wireless channel in an urban environment is often poor due to obstacles such as buildings, trees, and houses. This results in frequent transmission interruptions. Additionally, traffic flow is unevenly distributed, leading to network interruptions in areas with low vehicle density. These challenges pose a significant impact on the design and implementation of efficient and reliable transmission control protocols for the Internet of Vehicles [14]. Traditional mobile ad hoc network protocols are not well-suited for the highly dynamic environment of car networking, characterized by frequent topology changes and high mobility. The information-centric network (ICN), which decouples content and terminals at the network layer, has become the most potential and highly concerned future network architecture for the Internet of Vehicles [15]. The named data network (NDN) is currently the mainstream of ICN, and its development has attracted the attention of academia and industry. By focusing on content distribution rather than location, NDN improves the efficiency of information distribution. In NDN, the data structure and packet forwarding process are driven by consumers sending interest packets for the data packets from the producer. When a packet arrives at the NDN router, the router will first decode the data packet type to check whether the incoming data packet is an interest packet or a data packet. Fig. 2 explains the functionality of the NDN router, which has storage, routing, and forwarding functionalities. It has three corresponding data structures: Forwarding Information Base (FIB), content storage (CS), and pending interest table. In [16], the router determines whether cached content can be executed based on the set caching strategy when there is a matching content item. Meanwhile, the content is forwarded to the next-hop router based on the in-record and out-record mapping relationships stored in the PIT table. Afterward, the entry is deleted from the PIT table. The content needed by users is spread across various NDN nodes, where the data fragments may be identical or different. NDN's multi-source capability enhances data transmission rates compared to the traditional TCP protocol [17]. However, this feature also causes issues such as excessive data redundancy, leading to underutilization of node cache resources and waste of bandwidth resources as only the first-arriving data copy can be collected by the receiver. Unlike TCP/IP, NDN is not a connection-oriented transmission but rather a connectionless transmission driven by the receiver. Therefore, several transmission control techniques used in TCP/IP are not appropriate for NDN networks. Reference [18] have proposed new flow control and congestion control methods based on the characteristics of Information-Centric Networks. Currently, transmission control algorithms are divided into three categories: receiver-driven, hop-by-hop control, and hybrid. The receiver-driven transmission control protocol is primarily based on the interest packet-data packet pair transmission mechanism in NDN, where the receiver controls the transmission by managing the interest packet [19]. In contrast, the hop-by-hop transmission control protocol performs interest packet control at the intermediate routers to regulate transmission control. The hybrid transmission control protocol integrates both hop-by-hop and receiver control mechanisms, enabling both the receiving node and the intermediate routing node to manage the sending of interest packets. Some researchers have also focused on transmission control under multiple sources [20]. However, most of them establish independent congestion windows and RTT values for each source and use source prediction to execute the corresponding source's congestion algorithm and retransmission. One study assumes that the receiver knows the position of each content block before the transmission starts, and the position remains unchanged during the streaming transmission. Still, this is difficult to achieve in Information-Centric vehicular IoT networks due to the dynamic mobility of nodes from one location to another [21]. There are three different ways of designing and implementing ICN-IoT networks for vehicle-to-vehicle communication. The first approach is SDN-based vehicular ICN-IoT networks, where software dynamically controls and manages network behavior in a vehicular IoT environment. SDN separates the data plane from the control plane, providing a network overview by collecting information from network nodes. In the context of ICN-based VANET, SDN offers scalability, manageability, and a universal network view. In-network caching is a popular ICN module that enhances content availability and reduces retrieval delay. However, integrating ICN and SDN in VANETs is challenging due to vehicle mobility. The second approach is cloud-based vehicular ICN-IoT networks, where data centers are necessary for storing records due to hardware/software constraints and security issues. Exponential growth in internet traffic has led to the use of vehicles as cloud nodes. Utilizing the cloud as an infotainment feature is easy in the IP-based internet, but linking it with the ICN model is challenging due to vehicle mobility [22]. Proposals for enabling subscriber mobility include proactive caching and prompt request/reply recovery. The third approach is edge computing-based ICN-IoT vehicular networks, where edge computing is integrated with vehicular networks to enable fast content retrieval by bringing storage and computational processes closer to customers. However, integrating ICN with vehicular edge computing is challenging due to the mobility of vehicles. Accessing content stored in an edge node is easier in IP-based internet models but is complex with ICN, which uses names instead of IPs.

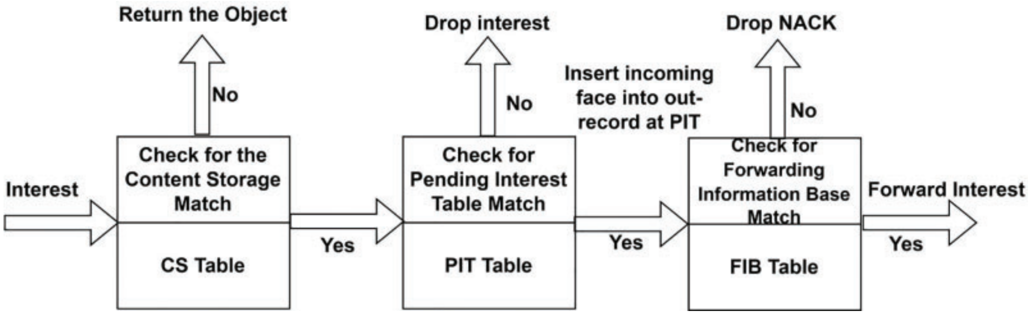
Return the Object
No
Drop interest
No
Insert incoming face into out-record at PIT
Drop NACK
No
Interest
Check for the Content Storage Match
CS Table
Yes
Check for Pending Interest Table Match
PIT Table
Yes
Check for Forwarding Information Base Match
FIB Table
Forward Interest
Yes

Figure 2: Operation of the NDN router at V2V-IoT networks

In [23], transmission control protocol (TCP) based on network coding is proposed to utilize cutting-edge transmission coding technology. This theory of network information flow has numerous applications, including peer-to-peer (P2P) communication, multi-path TCP, and multi-path routing [24]. Network coding comprises of two categories: "linear coding" and "non-linear coding". Non-linear coding is rarely used in practical applications due to its complexity in decoding. Linear coding involves coding packets that are linear combination packets made up of several content blocks. The linear combination coefficients can form a coding vector, which is categorized into "random linear encoding," "fountain code," and "system code," depending on whether the encoding vector is determined. Random linear network coding and fountain codes are the most commonly used coding methods in network coding applications [25]. The primary focus of this paper is to propose a receiver-assisted transmission control mechanism in NDN-based ICN-IoT vehicular networks. The objective is to minimize content data redundancy within intermediate routers and enhance the efficient utilization of both node and network resources.

3 Proposed Work

In order to effectively control the redundant transmission and better adapt to other characteristics of ICN, this paper proposes an algorithm for effective control of redundant transmission based on the integrated network coding suitable for the Internet of Vehicles. The network coding can hide the data sent by multiple sources. This paper first analyzes and designs the network coding methods suitable for the application scenarios. Later on, the setting of the size of the IoT router/L3 switch is analyzed based on the characteristics of the network. Furthermore, the overall structure of the designed transmission control protocol is described with each function module in detail. Finally, an information source selection algorithm based on node reliability analysis and a multi-information source scheduling algorithm is introduced to reduce redundancy and improve the overall IoT network throughput.

3.1 Design and Analysis of Network Coding

Due to the existence of multiple sources in the application scenario, information redundancy has become a key reason for the network congestion that results in extreme waste of bandwidth in the Internet of Vehicles. In order to reduce the packet drops with respect to data redundancy caused by multiple sources, this paper introduces network coding to control the number of redundant data packets. The characteristics of the fast-moving vehicle nodes, frequent network topology changes, and frequent network interruption were taken into prime consideration to analyze the packet drops. The fountain code coding method is more suitable for transmitting sensitive data on error-prone networks. However, since the fountain code uses a random encoding algorithm and the random matrix reconstructed by the receiver cannot be guaranteed to be of full rank, this method has a non-negligible possibility of decoding failure. Due to the existence of multiple sources in the application scenario, information redundancy has become a key cause of network congestion, and it will also cause extreme waste of bandwidth, an important resource in the Internet of Vehicles. In order to reduce the network impact caused by multiple sources, this section introduces network coding to control the number of redundant data packets.

The first thing to do is to design the network coding. Although network coding itself has many advantages, such as increasing network throughput by reducing the amount of data, making the data after network coding the same to realize the difference of shielding packets, etc., when introducing network coding, we need to adapt to local conditions. We will follow the application environment and Network characteristics, analysis, and design of network coding, so as to maximize the use of the advantages of network coding. This paper emphatically describes two practical coding methods, namely random linear coding and fountain codes. The advantages of random linear coding in comparison with the fountain codes is described in different aspects. The application scenario has both the characteristics of a named data network and the Internet of Vehicles. First of all, the characteristics of fast-moving vehicle nodes, frequent network topology changes, and frequent network interruptions make packet transmission prone to packet loss. Fountain code is a rateless coding and decoding method with linear complexity and extremely high coding efficiency. Hence, this coding method is more suitable for transmitting sensitive data on error-prone networks. At the receiver side decoding, when the sink node cannot receive enough coded packets, the percentage of successful decoding of the data is low, so it is not a good choice in the application scenario of this subject. Furthermore, since vehicle nodes often move at high speeds, the mobility of vehicles may be considered when selecting network coding. Compared with fountain codes, random linear codes are completely distributed due to their basic characteristics and are more robust to the joining and leaving of information source nodes. Hence, random linear coding is more suitable for fully distributed dynamic networks that change the topology with time, such as the Internet of Vehicles. On the other hand, since the information-centric Internet of Vehicles has the characteristics of universal caching, the choice of encoding method also needs to consider its impact on cache management and utilization. In the fountain code encoding method, the encoding object is to randomly select any number of parts from the Chunk to encode, which brings great troubles to the management and utilization of the buffer of the intermediate route vehicle node. However, the coding object of random linear network coding is the entire Chunk. This coding method makes the intermediate routing vehicle nodes more efficient for caching. In addition, when the fountain code reconstructs the same number of data symbols, it will generate more redundant data than random linear coding, which is contrary to our design goal of reducing the amount of redundant data as much as possible. Especially in the multi-source scenario, a set of fountain code symbols generated by different information source nodes may basically overlap, which will reduce the efficiency of information decoding and further reduce the transmission efficiency. Although the coding and decoding of random linear coding is more complicated than the Rapidor code, the current computer can easily handle such coding requirements. Considering the above aspects, in view of the characteristics of multi-source and car networking scenarios, this paper selects random linear coding to encode data packets in its best choice. In general, the information that will be returned to the requester will be called as content block “CHUNK”. The Chunk is the basic data unit for caching, transmission, and security in the network, and only one interest packet is needed to retrieve each chunk to avoid unnecessary interest packet cost and header overhead. Hence, its size setting has an important influence on the efficiency of network coding. Usually, the chunk size is set to 1500 bytes in most of the existing solutions. But due to many header fields in the transmission control protocol’s packet header, it is not suggestable to set it to 1500 bytes. When the chunk size is set too large, even though the signaling and control overhead is reduced to a certain extent, the poor quality of the wireless channel in the vehicular networking scenario leads to low link transmission efficiency. The retransmission of the entire block will trigger longer delays and reduced network throughput. In addition, due to the large size of the chunk block, the vehicle needs an excessive cache to store the chunk, which makes inserting/deleting content operations occur frequently. With this, the phenomenon of cache oscillation occurs.

3.2 Random Linear Network Coding

Random Linear coding is a type of network coding used in data transmission, which combines multiple data packets to create coded packets that are transmitted over the network. In Random Linear coding, a coding coefficient matrix is randomly generated, and each coding coefficient is chosen uniformly at random from a finite field. The coding process involves taking a group of data packets and forming a linear combination of those packets using the randomly generated coding coefficient matrix. The resulting coded packets are then transmitted to the destination node. At the destination node, the original data packets can be recovered using linear algebra operations on the received coded packets. In ICN-IoT vehicular networks, Random Linear coding is used to encode content packets into coded packets that are sent over the network. The coding coefficients used to generate the coded packets are randomly generated and sent along with the coded packets, which enables the receiver to reconstruct the original content packets.

3.3 Protocol Architecture

The content requested by the requester mostly exists in multiple vehicles. At the same time, the information-centric Internet of Vehicles also has built-in caching features. Multi-source has become the main feature of information distribution in the information-centric Internet of Vehicles. However, in the Internet of Vehicles, bandwidth is one of the most important network resources, and due to the existence of multiple information sources, there are a large number of redundant data packets in the network. When they are not uniformly controlled, then it will undoubtedly cause a huge waste of bandwidth resources. In addition, in the Internet of Vehicles, poor wireless channel quality can easily cause wireless packet loss. Therefore, when the requester does not receive the complete desired content, it will retransmit, which further causes a waste of network bandwidth resources. In view of the above factors, this paper proposes a transmission control protocol based on network coding to solve the issue of data redundancy and source uncontrollable in application scenarios. This protocol uses random linear network coding (RLNC) to encode the chunk of each content source. Firstly, we introduce the overall architecture design of the transmission control protocol and the functional description of the functional modules. In the application scenario, the vehicle application node plays three roles: interest requester, content source, and intermediate routing node. Unlike TCP/IP where the transmission is initiated by the sender, NDN transmission is initiated by the interested requesters, so these interested requesters play an important role in enhancing the network throughput.

In Network Cording Multiple Solutions Scheduling (NCMSS) protocol architecture (See Fig. 3), the requester and the information source can perform scheduling control, whereas the intermediate routing node has no authority to perform or schedule operations. The requester side and the information source side will mainly consist of seven parts, which can be divided into four functional modules, namely the dispatch control module, congestion control module, codec module, and information source selection module.

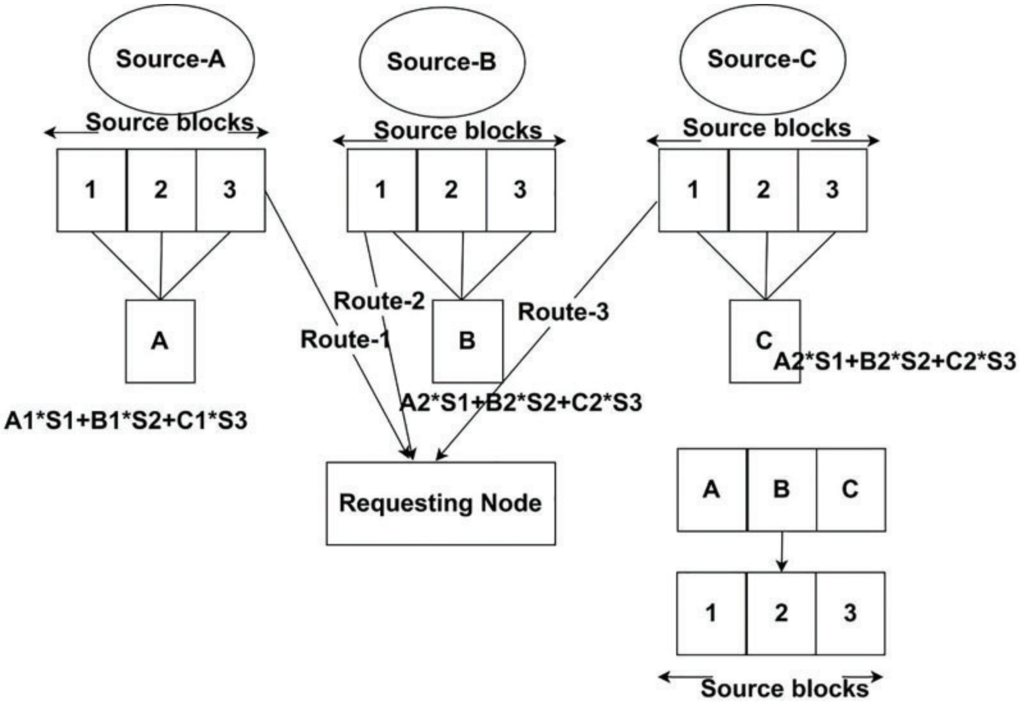

Figure 3: Overview of proposed network cording multiple solutions scheduling (NCMSS) protocol

Scheduling Control Module: The main function of this module is to determine the global control parameter based on the information in the received data packet, i.e., the expected reception cut-off time (Expected Packet Access Termination Deadline, EPATD). This parameter is also used as a reference for each content source to independently determine the number of encoded packets that need to be transmitted. It is mainly divided into four parts, namely the network monitoring unit, information source prediction unit, parameter control unit, and interest packet scheduling unit.

1) ***Network Monitoring Unit*:** Once the requester receives the encoded packet, it abstracts some source information (total number, source sending rate, source idle time, packet loss rate, etc.) from it and then sends this information to the Information source prediction unit and parameter control unit.

2) ***Information Source Prediction Unit:*** This unit is mainly used for universal caching. In addition, due to the characteristics of the application scenario, the information source presents random and dynamic characteristics. Since the interest packets sent by a requester are most likely to be continuous, some continuous blocks of the same content can be stored in the same node. Therefore, we can predict the source of the next content block from the source feedback obtained from the network monitoring unit.

3) ***Parameter Control Unit:*** When the request sent by the requester has timed out and still has not received a sufficient number of coded packets, then the requester will re-send the interest packet request, which will increase the delay and reduce the unnecessary window size. In order to prevent this problem from happening in advance, the architecture adds a parameter control unit, which can use the parameters passed by other units (such as the prediction unit: the number of content sources) to estimate the latest time (EPATD) that the global parameter requester expects to receive the data packet.)

4) ***Interest Packet Scheduling Unit:*** When the requester sends an interest packet, it inserts EPATD and the current time into it, and then the network finds the next hop according to the routing strategy.

Congestion Control Module: The main function of this module is to adjust the window size according to the degree of network congestion to make full use of network bandwidth. In the traditional TCP/IP network performance optimization, congestion control algorithms play a key role, and repeated ACK and timeout are congestion indicators in TCP/IP networks, and they are also preliminary requirements for window adjustment. In recent years, there have been many research results in the information-centric network congestion control protocol, such as ICTP, ICP [21,22], Connect, and CCTPM. ICP regards timeout as a sign of congestion while preserving out-of-order transmission. In ICTP and CCTP, it is only based on timeout to determine whether congestion occurs. Contug considers the unpredictability of multiple sources in the ICN network and proposes a multi-window mechanism that maintains an independent window for each source so as to achieve flow control of uncertain sources. Compared with the above-mentioned literature work, this protocol uses network coding, so out-of-order transmission is no longer a problem to be solved in the application scenarios of this subject. Hence, this paper adopts the same congestion control scheme (Contug) to detect congestion in Information-Centric Vehicular IoT networks [23].

Codec Module: This functional module is mainly distributed in two parts: the requester and the information source. On the requester side, the decoding function is that the requester recovers the requested data block from the received linearly independent encoded packet. On the information source side, it is mainly divided into two parts: one is a calculation control unit for controlling the number of packets sent by the information source, and the other is a coding unit that performs random linear coding on data packets. The calculation control unit mainly controls the number of packets that should be sent according to the EPATD set by the requester after the information source receives the interest packet request. The coding unit is mainly based on the number of sending packets and the source sending rate to generate the coding group, where the coding group is composed of a random linear combination of all packet groups in the block.

3.4 Source Selection Algorithm Based on Three-Phase Traffic Flow

In comparison with the traditional NDN routers, the Internet of Vehicles are more densely distributed, which results in more information sources. This undoubtedly brings a great amount of data redundancy to the vehicular network, which results in inefficient utilization of network bandwidth in Internet of Vehicles. In order to effectively control the amount of data redundancy, this paper proposes an information source selection algorithm in which vehicle nodes judge whether to join the information source combination based on their own reliability. The theory of three-phase traffic flow and the use of real data sets helps to analyze the mobility characteristics of vehicle nodes, thereby further analyzing the connectivity and stability of the network.

3.4.1 Overview of the Three-Phase Traffic Flow

The three-phase traffic flow divides the traffic flow into three phases, which can simulate a variety of realistic traffic phenomena. In theory, the traffic flow state is described by vehicle speed V, vehicle density p, and flow q, and the relationship between the three parameters is q = pv. Changes in the relationship between these three parameters affect the division of traffic flow.

Free Flow**:** The speed of the vehicle is basically assumed as stable, and the distance between the vehicle to the vehicle is far, whereas the flow rate is proportional to the density. When the density gradually increases to a critical value, it will change to other flow states.

Synchronous Flow**:** The speed of the vehicle has dropped significantly, and the density of the vehicle has increased. It usually exhibits a “balanced state”; that is, the vehicles are traveling at the same speed, and the distance between the vehicles is assumed to be the same. In the three-phase traffic flow theory, a microscopic traffic flow model is used to simulate traffic conditions. In this model, the synchronous flow state can be understood from the two-dimensional area in Fig. 3.

Wide-Movement Blocking Flow: It can only be transformed from a synchronous flow. In this state, the flow rate and the speed decrease significantly, and the speed drops to a very low value. In an urban scenario, this flow state generally appears at an intersection.

3.4.2 Node Mobility Analysis in Three-Phase Traffic Flow

When the requesting party sends an interest package to obtain traffic conditions, then the vehicle must be selected as the information source in the target area. Since the vehicle will continue to provide information to the requesting party, the information source vehicle must be able to maintain efficient transmission for a period of time and the surrounding area. In the Internet of Vehicles, the speed of the vehicle is different, and the distance between the vehicle and the preceding vehicle is different. In order to prevent the physical characteristics of the information source vehicle from affecting the information transmission, this paper uses node reliability as the performance index to select the information source. The higher the reliability of the vehicle node is, the more inclined to choose such vehicle nodes as the information source.

3.4.3 Algorithm for Node Reliability Analysis

While requesting the content to be obtained by the requesting party, the vehicle speed and the distance between the nodes are different when the vehicle nodes are in different flow states. Thus, the network topology connectivity and link stability formed around the vehicle nodes are also different. For the reliability analysis of vehicle nodes, this paper selects two performance indicators, namely network connectivity and link stability.

In general, when there is more neighbor nodes, then the probability of node stability is very high. This is theoretically valid with the below mathematical formula: Assume that the communication range of the vehicle is R, the length of a single vehicle is Lv, and the road is a one-way street. Vehicles on the same road are arranged in a straight line. Assuming that the average distance "d" between vehicles is the number of neighbour nodes around the vehicle, and the vehicle speed is V.

$$dd = \frac{2R}{Num(\{Vj, |D_{vi}, D_{vj}| < R\})} \tag{1}$$

According to the formula, when the number of neighbour nodes increases, then the average distance between vehicles decreases, which results in a decrease of the relative movement of the vehicular nodes. In the Internet of Vehicles, rapid changes in vehicle speed will seriously affect the reliability of the workshop connection and the stability of the cluster. The reliability of a vehicle depends not only on the number of neighbour nodes but also on the stability of its links with surrounding nodes, which can be indicated by the retention rate of the nodes around the vehicle over a period of time. Vehicle nodes can record surrounding nodes through periodic beacon packets and compare the number of overlapping nodes recorded over a period of time. Since the vehicle node in the free flow is not restricted by the preceding vehicle, overtaking often occurs, and the neighbouring nodes around the vehicle will change from time to time; the vehicle node in the synchronous flow is restricted by the preceding vehicle, and the distance between the vehicles is maintained within a certain range. But due to the vehicle, the movement trend may show deceleration or acceleration, and there are still changes in neighbour nodes to a certain extent because the vehicle nodes in the blocked flow are almost in a stationary state, the neighboring nodes of the vehicle are stable for a period of time.

3.4.4 Protocol Architecture

In the application perspective, an interest packet sent by the requester finally reaches the target area through the Internet of Vehicles routing strategy, and then multiple vehicle nodes are selected as the information source through the information source selection algorithm. After the information source vehicle receives the interest package, the vehicle will determine whether it contains the stored content or whether it has the ability to collect available information. When the vehicle can obtain the requested information, it will send a data block CHUNK back to the requester. When multiple information sources return data blocks, it is obvious that the duplicate data block CHUNK is returned, thus wasting most of the bandwidth in the network.

Algorithm 1: Information source selection algorithm for node reliability analysis

Input: Vehicle speed v_i, Vehicle acceleration a_i, Front vehicle speed v_j, Vehicle distance g_n, Free flow speed v_{free}

Output: Information source vehicle collection Set_v

```
START
$G_n = \max(0, [Ktv_i + a^{-1}v_j(v_i - v_j)])$
$FSD = \lambda * \left(\frac{v_i}{v_{free}}\right) * e^{-G_{n/gn}\left(v_{sf} - vn\right)}$
Judging the current flow 'State' according to the previous traffic flow state and itself
if State in {F, StoF, JtoF}
$NSv_i = Num(\{v_j, \left|Dv_i, v_j\right| < R\})/(R/L)$
$NNRv_i = ||sv_i, t_1|| |/| |sv_i, t_0| |$
else if State in {S}
```

(Continued)

Algorithm 1 (continued)

```
$NSvi = 1$
$NNRvi = ||Svi, t1||/||Svi, t0||$
else
$NSvi = 1$
$NNRvi = 1$
end if
$VRvi = NSvi * NNRvi$
Broadcast its $VR_{vi}$ to other vehicles and set a threshold
when the node exceeds the threshold, add it to the $Set_v$.
END
```

Algorithm 1 is used for judging the current flow state in a vehicular network and broadcasting the estimated velocity ratio (VRvi) of a vehicle to other vehicles in the network. The algorithm is described below:

- The first equation calculates the G_n value, which is used to determine the probability of a vehicle entering a jammed state. The G_n value is calculated based on the speed difference between the current vehicle (v_i) and the nearby vehicles (v_j) within a certain range. If the G_n value is greater than zero, the vehicle has a higher chance of entering a jammed state. The parameter Kt is a constant, a^{-1}) is the inverse of the acceleration, and v_i and v_j are the speeds of the current and nearby vehicles, respectively.
- The FSD equation calculates the probability of a vehicle entering a stopped state (State = S) based on the density of the traffic flow and the free-flow speed (v_{free}). The parameter λ is a constant, G_n is the previously calculated G_n value, and g_n and $Ø(v_{sf-}v_n)$ are functions that depend on the network density and the speed difference between the vehicle and the flow.
- Based on the current flow state (State), the algorithm calculates the number of nearby vehicles (NSvi) and the change in velocity ratio (NNRvi) of the current vehicle compared to the previous time (t_0) and the current time (t_1). The velocity ratio is calculated as the product of NSvi and NNRvi. If the current flow state is F, StoF, or JtoF, NSvi is calculated as the number of nearby vehicles (v_j) within a certain range (R) divided by the lane length (L), and NNRvi is calculated as the norm of the difference in speed vectors between the current and previous times divided by the norm of the speed vector at the previous time. If the current flow state is S, NSvi is set to 1, and NNRvi is calculated as the norm of the difference in speed vectors between the current and previous times divided by the norm of the speed vector at the previous time. If the current flow state is none of the above, NSvi and NNRvi are set to 1.
- The VRvi value is calculated as the product of NSvi and NNRvi and represents the estimated change in velocity ratio of the current vehicle.
- The estimated VRvi is broadcasted to other vehicles in the network. A threshold is set, and if the VRvi value of a vehicle exceeds the threshold, the vehicle is added to the Setv, which represents the set of vehicles with a high probability of changing their velocity ratio.

Algorithm 2: Operation when the requester receives Chunk

```
Input: $T_{idle}(S)$ at idle time, loss rate $p_s$, source sending rate $R_s$, requester sending Interest
Time $T_{start}(i)$, transmission delay $RTT_S/2$
```

(Continued)

Algorithm 2 (continued)

```
Output: Global parameter expected packet reception termination time EPATDi
START
for S = 1 to n do
if Source$_s$ contains content corresponding to Chunki
$T_{receive}(i, s) \leftarrow \text{Tstart(i)} + \text{RTTS}/2$
$T_{send}(i, s) = \max(T_{idle}(\text{s}), T_{receive}(\text{i, s}))$
end if
END
```

Algorithm 2 explains about the requester chunk reachability, and interest packet received from the information source. This scheduling algorithm requires the requester and information at the source side to perform the coordinated operations. At the requesting end, the expected receiving a period of global parameters is calculated by collecting and estimating the local parameters of the information source, and then the EPATD is added to the interest packet and sent to the information source. When the information source receives the EPATD, it is combined with the local node information. The parameter information determines the setting of the number of encoded packets. At the same time, the information source side adds part of the local parameter information to the encoding package, which is used to feed back to the requesting party to further calculate the EPATD so as to realize the control of the number of information source encoding packages. From the perspective of a single information source, independent of the operations of other sources, each information source can control the number of encoded packets to be sent based on the global parameters contained in the interest packet sent from the requester. But from the perspective of the entire network, the transmission of encoded packets of all information sources is controlled by the requesting party based on the information of the entire network.

3.4.5 Analytical Analysis of the NCMSS Protocol

Eq. (2) analyzes the relationship between the end-to-end throughput, data retrieval time, and caching delay time in an Information Centric IoT (IC-IoT) network.

$$T_{Network} = \frac{RT_{Data} + Delay_{cache}}{Delay_{trans}} \tag{2}$$

Let ‘t’ be the total time it takes to retrieve data from the source or the cache. Let ‘t_d’ be the data retrieval time, which is the time taken to retrieve data from the source. Let ‘t_c’ be the caching delay time, which is the time taken to retrieve data from the cache. Let ‘R’ be the data transfer rate or the network bandwidth, which is the rate at which data can be transmitted between the source and the destination. Let ‘D’ be the size of the data to be transferred. Assume that the data is either available in the cache or must be retrieved from the source. Assume that there is no packet loss, network congestion, or other factors that can affect the data transfer rate.

Based on these variables and assumptions, we can derive the following equations:

- $t = t_c + t_d$ (total time is the sum of caching delay time and data retrieval time)
- $t_d = D/R$ (data retrieval time is the data size divided by the data transfer rate)
- $t_c = t - t_d$ (caching delay time is the total time minus the data retrieval time)

We can substitute the equations for t_d and t_c into the equation for t to get:

$t = (D/R) + t_c$

Substituting this into the equation for end-to-end throughput, we get:

$$\text{End-to-end throughput} = D/((D/R) + t_c) \tag{3}$$

Simplifying this equation, we get:

$$\text{End-to-end throughput} = (R * D)/(D + R * t_c) \tag{4}$$

Algorithm 2 explains about the cache delay at intermediate ICN-IoT router. It is noteworthy that the channel reliability and cache retrieval delays are directly proportional to achievable end-to-end throughput. The proposed NCMSS protocol attain increased network throughput due to reduced caching delays and discovering the reliable end-to-end paths at the network layer.

4 Experimental Results

In order to verify and evaluate the performance of the transmission control protocol (NCMSS), this paper uses the Eclipse SUMO (Simulation of Urban Mobility), traffic simulation software, to simulate the vehicle movement. Subsequently, the network simulation software NS-3 is used to simulate the performance of the NCMSS and comparison protocols. NS-3 is a discrete-time event-driven network simulator that simulates timestamp events so that the internal clock of the simulator advances in a discrete manner. The development of NS-3 began in 2006 and is released approximately every 3 months or so. The main application areas are academic research and education, aiming to meet their teaching or research needs. NS-3 is mainly developed by C++ language and Python language, so it is convenient for users to operate and realize more powerful functions. A complete NS-3 simulator mainly includes functional modules such as Node, Channel, Network Device, and Application Program.

ndnSIM: As a new network protocol model, ndnSIM follows the structure of NDN and can be carried on any link layer protocol (802.11p) or transport protocol (TCP, UDP). This makes ndnSIM the possibility to simulate diverse scenarios. This simulator uses C++ language to implement different functions in a modular manner: PIT (Panding Interest Table), forwarding table FEB (Forwarding Information Base), content storage CS (Containt System and application interface, etc.) Multiple modules are independent of each other, so users can modify the modules individually without affecting the operation of other modules.

SUMO: Simulation of Urban Mobile (SUMO) is a widely used open-source, microscopic vehicle model that can be generated to simulate an urban traffic simulation system. It allows users to model transportation systems including road vehicles and public transportation. At the same time, SUMO's portability and interoperability are relatively high because it can provide real vehicle trajectories in any map input, and it can also collaborate with a network simulator for protocol performance evaluation. In this simulation, we output the vehicle motion model simulated by SUMO to rninSIM for use. The performance evaluation of the proposed NCMSS transmission control Information centric protocol is being tested and compared with ICTP (Information Centric Transmission Protocol) and NetInF TP vehicular transmission control protocol. Fig. 4a compares the network achievable throughput with varying number of IoT nodes in between the source and destination. When the number of nodes in between source to a destination increases with the active node mobility, then there is a high probability that node connectivity will be disrupted, which has a huge impact on the overall network performance. We experimented with this analysis to understand how the throughput will be affected with both the increased number of nodes and frequent node mobility within the communication range. Fig. 4 explains about the performance evaluation of proposed NCMSS protocol with the existing solutions.

In Fig. 4a Number of users in comparison with Network Throughput. (b) Network Transmit Power with respect to the distance between source and destination. (c) MAC data rate with Aggregate network throughput with packet size as 512 bits (d) MAC data rate with respect to end-to-end average packet delay for 512 bit data packet (e) MAC data rate with respect to end-to-end average packet delay for 1024 bit data packet (f) MAC data rate with Aggregate network throughput with packet size as 1024 bits data packet.

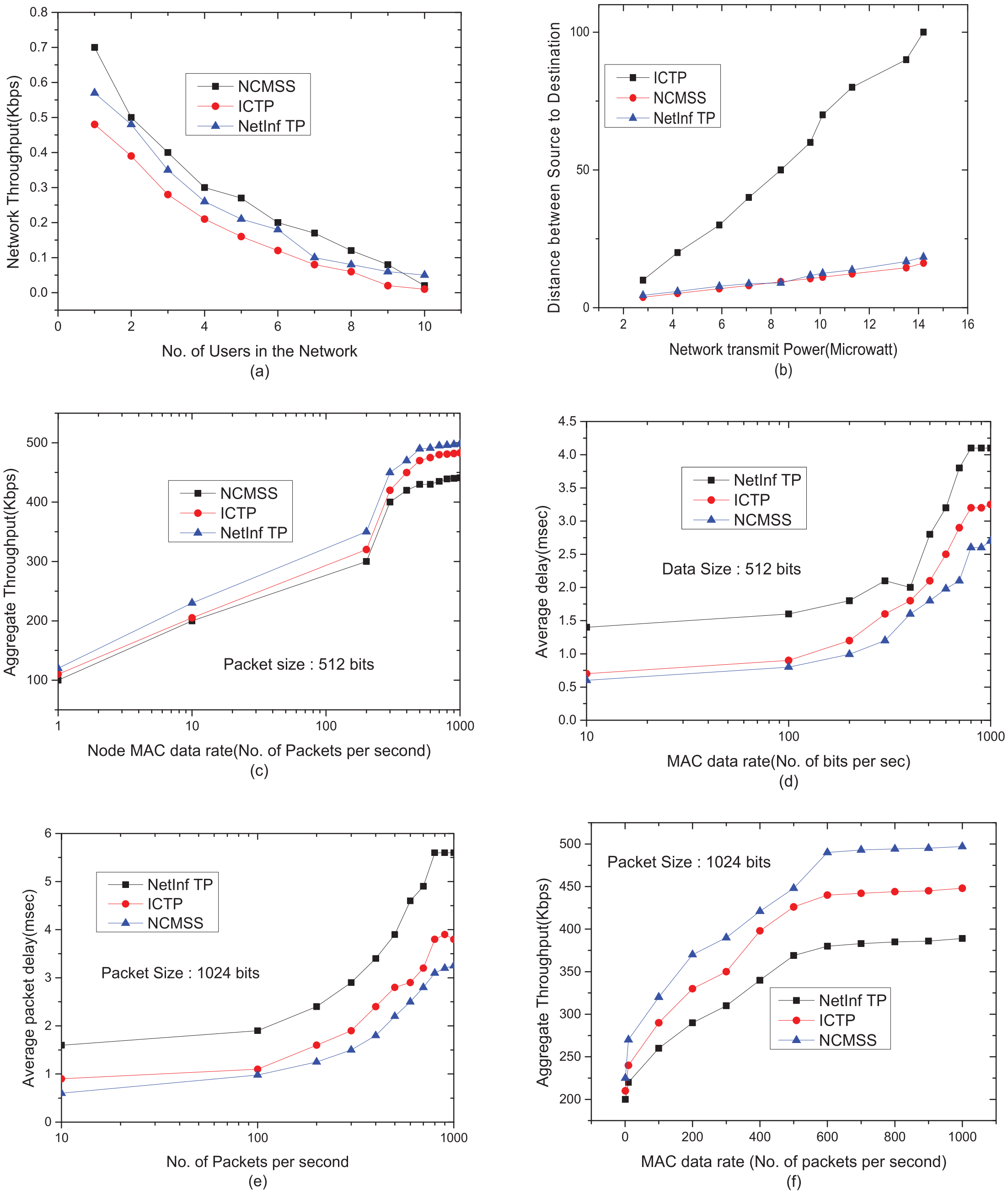

Figure 4: Performance evaluation of proposed NCMSS protocol with the existing solutions

From Fig. 4a, achievable network aggregate throughput in the proposed NCMSS protocol is outperformed in comparison with the existing Information Centric transmission protocols. This is due to the support of the node mobility aspect in the proposed NCMSS protocol with the implementation of packet buffering to store the inflight IoT packets during node handover from one access point to another access point. Fig. 4b explains about the network transmit power with respect to the distance between source node and the destination node. The node transmits power is calculated for channel setup (route control messages), application data transmission and channel route recovery during node mobility. Based on Fig. 4b, it is clear that the proposed NCMSS transmission protocol has overall less node transmit power in comparison with the other existing transmission control protocols. Fig. 4c explains about the MAC data rate with respect to the aggregate network Throughput (Kbps) for 512 bit packet size. From the Fig. 4c, it is clear that channel saturation is attained after certain number of packet transmissions in between the source and destination. The overall achievable throughput is better in NCMSS in comparison with the existing transmission control protocols. Fig. 4d explains about the average packet delay with respect to a number of packet transmissions with the packet size as 512 bits whereas Fig. 4e explains about the average packet delay with respect to the data transmission rate for 1024-bit data packets. From Figs. 4d and 4e, it is clear that there is increased delay for 1024 bit data in comparison with the 512 bit data packet. Fig. 4f describes about the network throughput for 1024 bit data. When we compare the network throughput for 512 bit and 1024-bit data, it is clear that 1024-bit packets are attaining with higher throughput. This is due to the reduced number of acknowledgment packets back from destination to source in 512-bit packets. Since, the channel reconstruction time due to packet drops are reduced in the proposed solution, the achievable throughput is increased and the end-to-end packet delay is reduced when compared with the existing solutions.

The contributions of the proposed redundant transmission control algorithm in Information-Centric IoT networks can be summarized as follows:

Improved reliability: By using multiple redundant paths for data transmission, the risk of data loss or corruption due to network failures or errors is reduced.

Increased efficiency: The use of redundant transmission algorithms can improve network efficiency by allowing for more flexible data routing, enabling the network to find the fastest and most reliable path for data transmission.

Enhanced network resilience: The use of redundant transmission algorithms can enhance network resilience by providing alternative data paths in the event of network failures or errors.

Better Quality of Service (QoS): The use of redundant transmission algorithms can help improve the QoS of the network by reducing latency and improving data transmission rates.

Robustness against interference: In Information Centric Vehicular IoT networks, the use of redundant transmission algorithms can help increase the robustness of the network against interference from other electronic devices, such as other vehicles or cellular networks.

5 Conclusion

Information Centric Vehicular IoT networks are used to provide end-to-end node connectivity based on the content information for disseminating vehicle safety, warning messages, emergency events, and passenger infotainment. Due to the frequent broadcast of constrained information through vehicular nodes, it is noteworthy that existing host-centric IP-based networks can’t be efficient in vehicular IoT networks. Thus, we propose a novel transmission control protocol for Information-Centric based IoT vehicular Networks (ICN) that address the issues like efficient redundant transmission control based on network coding to reduce data redundancy and accelerate the efficiency of information dissemination. The significance of the redundant transmission control protocol (TCP) in information-centric IoT networks is that it provides improved reliability and performance compared to traditional TCP. Redundant TCP can effectively handle network failures and congestion, reducing data loss and ensuring the timely delivery of critical information. However, it also has some limitations and trade-offs, such as increased overhead and complexity, which need to be carefully considered when deploying in real-world IoT networks. In the future, the performance of the proposed redundant transmission control protocol is going to be tested with different node mobility and spectrum mobility scenarios. The end-to-end throughput in proposed NCMSS is 22% higher (for 1024 byte data) than existing solutions whereas delay in NCMSS is reduced by 5% in comparison with existing solutions.

Funding Statement: This research is funded by Wenzhou Kean University under the IRSP Program "Hop by Hop Resource Reservation based Scheduling Function for Deterministic IoT networks".

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] H. Hartenstein and L. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Communications Magazine*, vol. 46, no. 6, pp. 164–171, 2008.

[2] T. Kosch, C. J. Adler, S. Eichler, C. Schroth, M. Strassberger *et al.,* "The scalability problem of vehicular ad hoc networks and how to solve it," *IEEE Wireless Communications*, vol. 13, no. 5, pp. 22–28, 2006.

[3] M. Obaidat, M. Khodjaeva, J. Holst and M. Ben Zid, "Security and privacy challenges in vehicular ad hoc networks," in *Connected Vehicles in the Internet of Things*, Cham: Springer, pp. 223–251, 2020.

[4] M. Chahal, S. Harit, K. K. Mishra, A. K. Sangaiah and Z. Zheng, "A survey on software-defined networking in vehicular ad hoc networks: Challenges, applications and use cases," *Sustainable Cities and Society*, vol. 35, no. 4, pp. 830–840, 2017.

[5] S. Panichpapiboon and W. Pattara-Atikom, "A review of information dissemination protocols for vehicular ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 3, pp. 784–798, 2011.

[6] C. Lochert, H. Hartenstein, J. Tian, H. Fussler, D. Hermann *et al.,* "A routing strategy for vehicular ad hoc networks in city environments," in *IEEE IV2003 Intelligent Vehicles Symp. Proc. (Cat. No. 03TH8683)*, Columbus, OH, USA, 2003.

[7] S. Kannan, G. Dhiman, Y. Natarajan, A. Sharma, S. N. Mohanty *et al.,* "Ubiquitous vehicular ad-hoc network computing using deep neural network with iot-based bat agents for traffic management," *Electronics*, vol. 10, no. 7, pp. 785, 2021.

[8] V. O. Nyangaresi and A. J. Rodrigues, "Efficient handover protocol for 5G and beyond networks," *Computers & Security*, vol. 113, no. C, pp. 102546, 2022.

[9] V. O. Nyangaresi, A. J. Rodrigues and S. O. Abeka, "Machine learning protocol for secure 5G handovers," *International Journal of Wireless Information Networks*, vol. 29, no. 1, pp. 14–35, 2022.

[10] A. Alferaidi, K. Yadav, Y. Alharbi, N. Razmjooy, W. Viriyasitavat *et al.,* "Distributed deep CNN-LSTM model for intrusion detection method in IoT-based vehicles," *Mathematical Problems in Engineering*, vol. 2022, pp. 8, 2022.

[11] D. Bahmanyar, N. Razmjooy and S. Mirjalili, "Multi-objective scheduling of IoT-enabled smart homes for energy management based on Arithmetic Optimization Algorithm: A Node-RED and NodeMCU module-based technique," *Knowledge-Based Systems*, vol. 247, no. C, pp. 108762, 2022.

[12] A. R. Gad, A. A. Nashat and T. M. Barkat, “Intrusion detection system using machine learning for vehicular Ad Hoc networks based on ToN-IoT dataset,” *IEEE Access*, vol. 9, pp. 142206–142217, 2021.

[13] F. A. Alhaidari and A. M. Alrehan, “A simulation work for generating a novel dataset to detect distributed denial of service attacks on Vehicular Ad hoc NETwork systems,” *International Journal of Distributed Sensor Networks*, vol. 17, no. 3, 2021.

[14] F. Azam, S. K. Yadav, N. Priyadarshi, S. Padmanaban and R. C. Bansal, “A comprehensive review of authentication schemes in vehicular ad-hoc network,” *IEEE Access*, vol. 9, pp. 31309–31321, 2021.

[15] Y. Yang and T. Song, “Energy-efficient cooperative caching for Information-Centric wireless sensor networking,” *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 846–857, 2021.

[16] D. Gupta, S. Rani, S. H. Ahmed, S. Verma, M. F. Ijaz *et al.,* “Edge caching based on collaborative filtering for heterogeneous ICN-IoT applications,” *Sensors*, vol. 21, no. 16, pp. 5491, 2021.

[17] H. S. Shrisha and U. Boregowda, “An energy efficient and scalable endpoint linked green content caching for named data network based internet of things,” *Results in Engineering*, vol. 13, no. 2, pp. 100345, 2022.

[18] S. Mori, “A fundamental analysis of an erase code-enabled data caching scheme for future UAV-IC-WSNs,” in *Proc. IARIA the 20th Int. Conf. Networks (ICN 2021)*, Porto, Portugal, pp. 8–12, 2021.

[19] W. Rafique, A. S. Hafid and S. Cherkaoui, “Complementing IoT Services using software defined information centric networks: A comprehensive survey,” *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 23545–23569, 2022.

[20] Q. N. Nguyen, R. Ullah, B. S. Kim, R. Hassan, T. Sato *et al.,* “A cross-layer green information-centric networking design toward the energy internet,” *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 3, pp. 1577–1593, 2022.

[21] S. Dilek, K. Irgan, M. Guzel, S. Ozdemir, S. Baydere *et al.,* “QoS-aware IoT networks and protocols: A comprehensive survey,” *International Journal of Communication Systems*, vol. 35, no. 10, pp. e5156, 2022.

[22] B. Yi, X. Wang and M. Huang, “Content delivery enhancement in vehicular social network with better routing and caching mechanism,” *Journal of Network and Computer Applications*, vol. 177, no. 6, pp. 102952, 2021.

[23] M. Amadeo, “A literature review on caching transient contents in vehicular named data networking,” *Telecom.*, vol. 2, no. 1, pp. 75–92, 2021.

[24] S. Anamalamudi, M. S. Alkatheiri, E. Al Solami and A. R. Sangi, “Cooperative caching scheme for machine-to-machine information-centric IoT networks,” *IEEE Canadian Journal of Electrical and Computer Engineering*, vol. 44, no. 2, pp. 228–237, 2021.

[25] D. Grewe, M. Wagner, U. Ambalavanan, L. Liu, N. Nayak *et al.,* “On the design of an information-centric networking extension for IoT APIs,” in *2021 IEEE 94th Vehicular Technology Conf. (VTC2021-Fall)*, online, pp. 1–6, 2021.