Wireless Communications and Mobile Computing

Software-Defined Industrial Internet of Things

Lead Guest Editor: Jiafu Wan
Guest Editors: Chin-Feng Lai, Houbing Song, Muhammad Imran, and Dongyao Jia

Software-Defined Industrial Internet of Things

Wireless Communications and Mobile Computing

Software-Defined Industrial Internet of Things

Lead Guest Editor: Jiafu Wan
Guest Editors: Chin-Feng Lai, Houbing Song, Muhammad Imran, and Dongyao Jia

Editorial Board

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 7947638, 2 pages
https://doi.org/10.1155/2019/7947638

Editorial

Software-Defined Industrial Internet of Things

Jiafu Wan,[1] **Chin-Feng Lai**,[2] **Houbing Song**,[3] **Muhammad Imran**,[4] **and Dongyao Jia**[5]

[1]*School of Mechanical & Automotive Engineering, South China University of Technology, Guangzhou, China*
[2]*Department of Engineering Science, National Cheng Kung University, Tainan, Taiwan*
[3]*Electrical Engineering and Computer Science, Embry-Riddle Aeronautical University, Daytona Beach, USA*
[4]*College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia*
[5]*Institute for Transport Studies (ITS), University of Leeds, Leeds, UK*

Correspondence should be addressed to Jiafu Wan; mejwan@scut.edu.cn

Received 24 December 2018; Accepted 26 December 2018; Published 21 May 2019

Despite the progress of embedded systems and the development of information and communication technology in recent decades, the industrial systems are still expected to evolve due to the constant science advancement. It is commonly believed that Cyber-Physical Systems (CPS) are Industry 4.0 enablers. Based on the context of Industry 4.0, Industrial Internet of Things (IIoT) is promoting and driving the industrial upgrading [1].

In order to implement the flexible, customizable, and efficient industrial systems, all related enabling technologies (e.g., industrial wireless networks, cloud computing, big data, and social networks) or devices (e.g., intelligent robots and flexible conveyors) must be developed as well for being integrated into IIoT systems. However, we still face some challenges: the efficient interaction and coordination between IIoT for the Industry 4.0 production, the configurable data acquisition node for more application scenarios, and a heterogeneous network for all kinds of transmission of information [2]. Fortunately, Software-Defined Networking (SDN) possesses the feature that can manage network services through abstraction of higher-level functionality [3]. Therefore, enlightened by SDN, there is a new idea for the information interaction of industrial environment by introducing software-defined IIoT to make the network more elastic.

This special issue targets innovative and validated solutions for improving the information interaction of IIoT. The following 8 papers were selected for inclusion in this special issue after rigorous reviews by experts in the respective domains.

The paper entitled "Analyzing Critical Failures in a Production Process: Is Industrial IoT the Solution?" by S. Ahmad et al. focused on analysis of machine failure causes. It motivated to investigate the main causes of failures (COF) due to material deficiency and production organizational environment which were adversely affecting the manufacturing processes.

The paper entitled "LAB: Lightweight Adaptive Broadcast Control in DSRC Vehicular Networks" by L. Ye et al. proposed a novel method, named Lightweight Adaptive Broadcast Control (LAB), for DSRC safety message to make full use of channel and avoid channel congestion. Extensive simulations were also designed to evaluate the method.

The paper entitled "Software Architecture Solution Based on SDN for an Industrial IoT Scenario" by J. L. Romero-Gázquez and M. V. Bueno-Delgado identified the main problems that could arise in the I4.0 adoption for a medium-high factory. After that, an open source software solution architecture based on ODL+IoTDM has been proposed to orchestrate the whole I4.0 infrastructure, enabling interoperability and management of IIoT devices from different vendors, including conventional industrial machinery and IT networks.

The paper entitled "PEMC: Power Efficiency Measurement Calculator to Calculate Power Efficiency and CO2 Emissions in Data Centers" by M. Uddin et al. proposed a novel PEMC metrics. It measured the overall performance of datacenter from time to time as it allowed the administrators to determine the exact utilization of already installed devices and their appropriate power consumption. It also aided in

measuring CO2 emission at the same time by applying the same tool hence reducing the costs and hassle involved with using a different tool.

The paper entitled "Impact of Packet Size in Adaptive Cognitive Radio Sensor Network" by M. Al-Medhwahi et al. investigated the impact of packet size on the performance of CRSNs in terms of two main metrics, namely, the average delay and the throughput. It also examined the interactivity between the packet size and the main parameters of the radio network and showed the resultant effects on the system performance

The paper entitled "TTEthernet Transmission in Software-Defined Distributed Robot Intelligent Control System" by C. Liu et al. provided architecture for a bus-based software-defined intelligent robot system and designed scheduling algorithms to make TTEhernet play the role of scheduling in the architecture. It solved the problem of nonreal-time and uncertainties of distributed robotic systems. Moreover, a fragment strategy was proposed to solve the problem that rate limits traffic and thus there was a large delay.

The paper entitled "Pipeline Leak Aperture Recognition Based on Wavelet Packet Analysis and a Deep Belief Network with ICR" by X. Lang et al. proposed the method for leak aperture recognition of pipeline based on WPA and DBNICR. Moreover, the method was tested on sound velocity of the ultrasonic data of an experimental pipeline to recognize the different leak apertures, which showed that the proposed method can reliably recognize the different leakage apertures.

The paper entitled "Industrial Internet of Things Based Efficient and Reliable Data Dissemination Solution for Vehicular Ad Hoc Networks" by S. Latif et al. proposed a new data dissemination protocol DDP4V to overcome the challenging broadcast storm, network partition, intermittently connected network. and optimum next forwarding vehicles (NFVs) selection problems, which showed the potential to provide an efficient data dissemination in diverse VANET scenarios with varying traffic conditions.

Conflicts of Interest

The editors declare that they have no conflicts of interest regarding the publication of this Special Issue.

Jiafu Wan
Chin-Feng Lai
Houbing Song
Muhammad Imran
Dongyao Jia

References

[1] J. Wan, S. Tang, D. Li et al., "Reconfigurable Smart Factory for Drug Packing in Healthcare Industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 507–516, 2019.

[2] J. Wan, S. Tang, Z. Shu et al., "Software-Defined Industrial Internet of Things in the Context of Industry 4.0," *IEEE Sensors Journal*, vol. 16, no. 20, pp. 7373–7380, 2016.

[3] X. Li, D. Li, J. Wan, C. Liu, and M. Imran, "Adaptive Transmission Optimization in SDN-based Industrial Internet of Things with Edge Computing," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1351–1360, 2018.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 3051204, 9 pages
https://doi.org/10.1155/2018/3051204

Research Article

Impact of Packet Size in Adaptive Cognitive Radio Sensor Network

Mohammed Al-Medhwahi, Fazirulhisyam Hashim, Borhanuddin Mohd Ali, and A. Sali

Department of Computer and Communication Systems Engineering & Research Centre of Excellence for Wireless and Photonic Networks (WiPNET), Faculty of Engineering, Universiti Putra Malaysia, Selangor, Malaysia

Correspondence should be addressed to Mohammed Al-Medhwahi; med100013@yahoo.com

Received 12 April 2018; Revised 18 August 2018; Accepted 27 November 2018; Published 9 December 2018

Guest Editor: Jiafu Wan

A cognitive radio sensor network (CRSN) is a solution that enables sensor nodes to opportunistically access licensed radio channels. Data transmitted over a network are divided into packets. In machine-to-machine communication, which is a heterogeneous nature of wireless networks, small-size packets are the common form of traffic. Due to the nature of CRSNs, small data packets will not allow a balance between optimal performance of the network and fulfilling the secondary network obligations towards the primary network in terms of interference. Either interference or channel's underutilization would result from employing data packets of inadequate size. In this paper, the appropriate packet size for adaptive CRSN is investigated by examining the performances of small, medium, and large packet size. In contrast to the trends of exploiting small packets of sizes up to 128 bytes, this study demonstrates that medium-size packets are more appropriate to yield the best performance in CRSNs. Simulation results show that packets of size 375 bytes outperform smaller and larger packets in many CRSN protocols. The induced delay that is partially caused by interference is decreased at the same time the channels are efficiently utilized.

1. Introduction

Wireless sensor networks (WSN) applications are implemented in traditional and emerging applications such as security, automated industry, and e-Health. Currently, the implementation of WSN applications in many new surveillance and monitoring services is facing a disconcerting challenge due to the radio spectrum scarcity of free licensed bands. WiFi, Bluetooth, cordless phones, and microwave ovens technologies utilize the same radio spectrum. This causes a serious spectrum congestion and disruption of a wireless network considering the close proximity of the utilized frequencies [1, 2]. Interference caused by the hostile radio environment results in high rates of data loss which consumes excessive energy and shortens the WSN network's lifetime [3]. On the contrary, licensed bands are being underutilized by licensed users which can be tapped by secondary network users (SUs) [4].

Cognitive radio (CR) technology enables SUs to opportunistically utilise the licensed channels assigned to primary network users (PUs). Such technology aims at improving the spectrum utilisation and mitigating the effects of the license-free spectrum overcrowding [5]. In cognitive radio networks (CRNs), SUs are strictly obliged to avoid interference with PUs. Once a PU commences transmission on the said channel, SUs must instantly evacuate the channel known as handoff. CR uses spectrum sensing (SS) to allow SUs to determine an idle radio channel to commence data transmission. It can also defer a data transmission if the channel of interest is busy. Among many signal detection techniques, the energy detection (ED) is the most common [6, 7]. In the ED technique, the radio channel is considered busy if the energy of the detected signal exceeds a predefined threshold value. Let $E(n)$ denote the energy of the sampled signal received by the SU receiver. H_0 denotes the hypothesis of the absence of the PU signal while H_1 denotes its presence. The energy of the received signal can be expressed as

$$E(n) = \begin{cases} W(n); & H_0 \\ W(n) + S(n); & H_1 \end{cases} \quad (1)$$

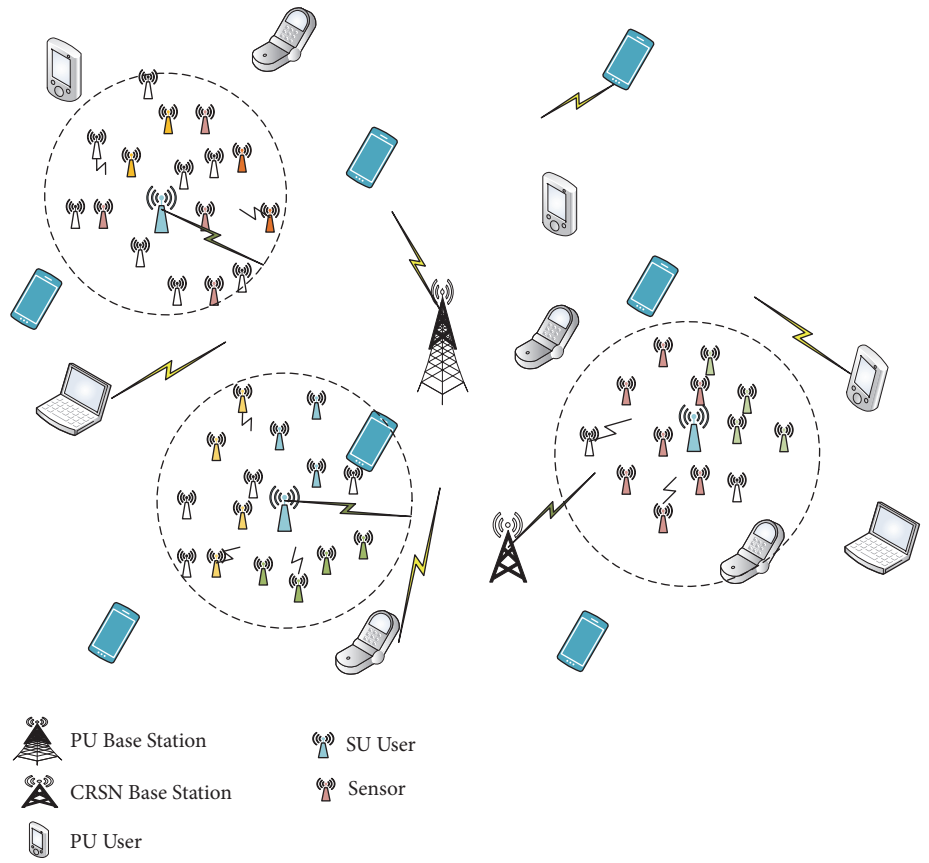


Figure 1: Typical cognitive radio sensor network.

where $W(n)$ represents the additive white Gaussian noise (AWGN) and $S(n)$ represents the transmitted signal multiplied by the channel gain.

The probability that the PU channel is busy is given by

$$P_{on} = \frac{T_{on}}{T_{on} + T_{off}} \tag{2}$$

with T_{on} as the channel busy time and T_{off} as channel idle time. The probability that the channel is free P_{off} is given as

$$P_{off} = \frac{T_{off}}{T_{on} + T_{off}} \tag{3}$$

Probability that the channel is detected as busy when it is actually busy is called the detection probability, P_{de}, and probability that the channel is detected as busy when it is idle is called the false alarm probability, P_{fa}. Optimal detection can be achieved only if the noise power is known to the SUs. The channel status is determined based on the value of the received signal's energy E and a predefined threshold value T_r. The relationship between P_{de} and P_{fa} with T_r and E is given by (4) and (5), respectively.

$$P_{de} = P_r\{D = 1 \mid H_1\} = P_r\{E > T_r \mid H_1\} \tag{4}$$

and

$$P_{fa} = P_r\{D = 1 \mid H_0\} = P_r\{E > T_r \mid H_0\} \tag{5}$$

Emerging technology that enables the opportunistic access for WSN's units is called the cognitive radio sensor network (CRSN), whose typical layout is shown in Figure 1. Appropriate data packet size helps limit the interference between the signals of the WSN units, equipped with CR capabilities, and PU signals [8–10]. Thus, size of data packet in CRSN has a twofold importance: to achieve a satisfactory performance and to mitigate harmful interference with PU signal. Small data packets offer maximum reliability and minimum latency for most types of data traffic especially in critical traffic. Together with the growth of sensor network in internet of things (IoT), the result is that small data packets, of sizes up to 128 bytes (1024 bits), have become the popular trend in data transmission [11, 12]. Subsequently, overhead data and the inefficient utilisation of resources induced by using small-size packets in emerging heterogeneous networks are often overlooked. Other substantial factors such as small-size packets have low signal noise ratios (SNRs) compared to large size packets, in effect of noise such as thermal noise [13], and the fact that large-size packets are able to achieve higher efficiency of bandwidth utilisation [14, 15] is neglected.

The contributions of this study can be summarised as follows:

(i) It investigates the impact of packet size on the performance of CRSNs in terms of two main metrics, namely, the average delay and the throughput

(ii) It examines the interactivity between the packet size and the main parameters of the radio network and shows the resultant effects on the system performance

(iii) It suggests adopting medium size by proving the outperformance of medium-size packets compared to the small and large packets.

This study is an extension of the work in [16] and it is organised as follows: Section 2 presents the related works, Section 3 introduces the system model and shows the adopted MAC protocol, Section 4 presents the mathematical model, Section 5 shows the evaluation and the results are discussed, and Section 6 concludes the work.

2. Related Works

Several studies have introduced solutions to improve the performances of MAC protocols in CRSN networks and some have analysed or modelled the performance metrics of a network but little attention was given to the influence of packet size [18–20]. Data collisions can occur during data transmission. During the data transmission phase, the PU is active until the transmission ends. The longer the SU's packet duration, the more collisions with the PU's data that result in decreasing the throughput of the SU. Furthermore, data collision can happen between SU and existing PUs and also between the SUs themselves.

An analytical model was presented in [21] to determine a suitable packet size subject to CRN parameters such as network traffic, sensing accuracy, and PU's density. A new formula was introduced to the normalised throughput of SU under perfect and imperfect scenarios of the SS function. The required complexity is not acceptable in CRSNs. Aimed at maximizing the goodput, another study based on carrier spectrum multiple access (CSMA) mechanism [22] proposed an analytical model to determine the optimal packet size based on the packet error probability, the collisions between the SUs themselves, and the collisions between SUs and PUs. The outcomes of the study do not treat the plain structure of the end sensor nodes. A similar study [23] investigated the impact of packet size variety on the throughput of a CSMA-based cognitive WLAN. In [24], a framework was proposed and modelled aimed at maximising the achievable throughput while the framework in [25] was more concerned with increasing the network lifetime. The performance of the dynamic open spectrum sharing (DOSS) MAC protocol for a CSMA-based CRSN was analysed and modelled in [17] which incorporates multiple channel access. The study partially investigated the effect of using long and small sizes of data frames on the performance of ad hoc-based CRSNs. Such study does not treat cluster-based networks of CRSNs that are more suitable for heterogeneous networks.

3. System Model

Network nodes are considered to be uniformly distributed in 2D square area of size l x l. The primary network includes M data channels $(C_1, .., C_M)$, with equal bandwidths. The operating area is divided into clusters each consisting of one master node and several ideal sensing nodes that are spatially correlated. The ideal sensing node is self-powered and it is considered to be heterogeneous that is producing several types of data packets according to how critical the obtained measurements are. The traffic is classified into two categories, namely, real time (RT) traffic and non-real time (NRT) traffic. RT traffic represents the most important traffic containing critical data, which must acquire priority in transmission and avoid interference. The master node is responsible for collecting traffic of a group of ideal nodes and performing several main CR functions. The buffer of the master node is assumed to be infinite and there are no limits on the number of packets it may contain. The environment of the licensed channels is homogeneous in terms of the bandwidth and the radio physical conditions. While the communication inside each cluster is a single-hop via the licensed-free spectrum, the communication between the master node and the base station (BS) is opportunistic via the licensed spectrum, i.e., through the radio channels of the primary network. A common control (CC) channel is used for the communications of the control messages. The channels between the ideal node and the master node are assumed to be perfect because of the short distances between them. The adaptivity of the adopted network refers to the framework mechanism in which the end nodes are able to resort to the alternative workflow, which is CR-based, when the usual WSN workflow suffers from a significant degradation in its performance [26]. In the alternative flow, the sensor node reroutes its data to the master node rather than routing it to its original cluster head (CH).

3.1. Medium Access Control. Existing WSN MAC protocols such as IEEE 802.15.4, WirelessHART, and ISA-100.11a are not suitable for CRSN networks since they lack the efficiency in terms of the network heterogeneity. Likewise, using CRN's proposed algorithms such as IEEE 802.22 will result in producing huge volume of overhead data and consuming enormous amount of sensors' precious power [27–29]. Pliable Cognitive MAC (PCMAC) algorithm, proposed and modelled in [16], concerns the communication between the master node and the BS and aims at maintaining the ideal sensor's capabilities in the same time of exploiting the CR technique. The master nodes use carrier spectrum multiple access with collision avoidance (CSMA/CA) scheme to access the CC channel that they use to exchange the control messages with the BS through. Request to send/clear to send (RTS/CTS) mechanism is used by the master node and the BS through the CC channel to negotiate the data channel reservation. This is also helpful in mitigating the effect of the hidden node problem. Inside each master node's buffer, packets are virtually lined up in to two virtual queues according to their category: RT and NRT. First come first served (FCFS) discipline is adopted for scheduling the packets in each queue. The RT packets have higher priority to be submitted earlier than the NRT packets since the latter is more delay-tolerant. In e-Health applications, for instance, RT traffic might represent the monitoring data of patient's activities such as the measurements of the heart and the brain while NRT traffic represents the ordinary traffic such as the environmental measurements of the room.

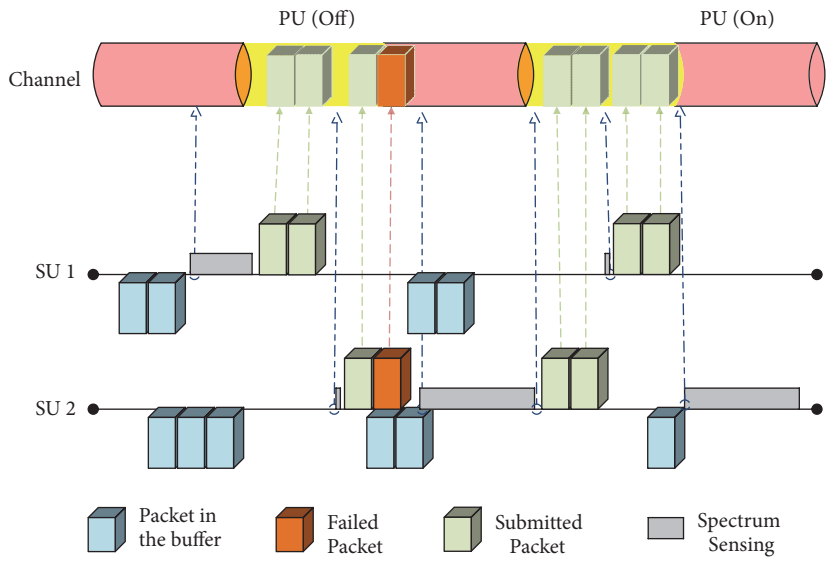


Figure 2: Packets transmission.

When the BS receives an RTS message, it responds with a CTS message and assigns an unallocated channel to the requesting node. The BS tunes its transceiver to the channel it has assigned to receive data. Concurrently, the master node receives the channel assignment message, enclosed within the CTS message; then it tunes its transceiver into the selected radio channel and begins to sense it until it becomes idle, i.e., spectrum hole. At that time, the master node can commence submitting its data in a burst-based manner. The master node will continue to transmit its data until either its data buffer becomes clear or it receives no Acknowledgement (ACK) message from the BS for the latest packet.

Figure 2 illustrates the mechanism of packet transmission where two SUs exploit the channel's idle time to submit their data. Packets are transmitted in bursts. The node tries to squeeze through as many packets as it can before the PU becomes active again. As long as the transmitting node does not receive its ACK message, a packet may interfere with the PU signal, as the second packet of the SU2. Failed packet has to be transmitted again in the next transmission cycle. In such a case, the master node will continue to sense the same channel for when the PU evacuates the channel to allow for the remaining data packets to be transmitted. Since the master node includes its buffer size information in its submission request, the BS will not reassign that channel until the allocated node finishes its transmission. The assignment is limited in time and if the node fails to submit its data for the predefined period of time, the BS can reallocate the channel to another requesting node. In such cases, the former node has to begin a new transmission cycle starting with sending an RTS message via the CC channel.

4. Mathematical Model

The M/G/1 model is a model where the arrival of packets is Markovian which occurs according to a Poisson process at rate λ with a general distribution, G, of the service time T through the CC channel, denoted by 1. The adopted model in this study is an improved M/G/1 with a non-preemptive priority scheme, inasmuch as it estimates the delay not only in the CC channel but also in the data channels.

The service time T of each packet comprises the waiting time in the queue W and the transmission time T_{tr}. The latter includes the contention time in the CC channel, T_c, the licensed channel's sensing time T_{ss}, and the packet submission time T_p,

$$T_{tr} = T_c + T_{ss} + T_p \tag{6}$$

where $\overline{T_c} = DIFS + \overline{T_b} + T_{rts} + SIFS + T_{cts}$. DIFS and SIFS are the interframe space and short interframe space durations of the distributed coordination function (DCF), respectively. T_{rts} and T_{cts} represent durations of the RTS and CTS frames while T_b represents the backoff estimated time that can be expressed as [30]

$$\overline{T_b} = CW_{min}\frac{1 - p - 2^N p^{N+1}}{2 - 4p} - \frac{1}{2} \tag{7}$$

CW_{min} is the minimum contention window size, p represents the conditional collision probability, and N is the number of the contender nodes.

Since the node can arrive at the assigned channel at any moment within the $[T_{on} + T_{off}]$ period, the time required to observe the channel until it becomes idle can be estimated as

$$\overline{T_{ss}} = \frac{T_{on}}{2}P_{on} \tag{8}$$

where P_{on} is the licensed channel's occupancy probability (by PU) obtained by (2).

Assuming that both RT and NRT packets have the same size, packet submission time can be expressed as

$$T_p = CCA + \frac{L}{D_{dr}} + SIFS + T_{ack} \tag{9}$$

where

CCA is clear channel assessment average time

L is packet size

D_{dr} is data channel's capacity (bits/s)

For simplification, the perfect sensing is assumed, meaning that $P_{de} = 1$ and $P_{fa} = 0$. However, in the case of imperfect sensing, the new value of the probability that a channel will be sensed idle will be given by $P_{off}(1 - P_{fa})$, whereas the old value, i.e., for the perfect sensing P_{off}, can be obtained by (3). In several applications, the acceptable values for P_{fa} and P_{de} are 0.1 and 0.9, respectively.

The occupancy ratio of the CC channel ϕ_{cc} is the sum of the occupancy ratios of RT and NRT, $\phi_{cc} = \phi_{rtc} + \phi_{nrtc}$, where

$$\begin{aligned} \phi_{rtc} &= \lambda_{rt}\overline{T_c} \\ \text{and } \phi_{nrtc} &= \lambda_{nrtc}\overline{T_c} \end{aligned} \tag{10}$$

To maintain the system stability, the occupation rate for the CC channel must be less than one. Similarly, the occupation rate of the data channels after considering the number of utilised channels can be estimated for RT and NRT packets, respectively, as

$$\begin{aligned} \phi_{rtd} &= N\lambda_{rt}\frac{\overline{T_{ss}} + T_p}{C} \\ \text{and } \phi_{nrtd} &= N\lambda_{nrt}\frac{\overline{T_{ss}} + T_p}{C} \end{aligned} \tag{11}$$

The mean waiting time for an RT packet, $\overline{W}_{rt}$, is estimated as

$$\overline{W}_{rt} = \overline{L}_{rt}\overline{T} + \Phi\overline{\mathscr{R}} \tag{12}$$

where $\overline{\mathscr{R}}$ is the mean residual service time of a current served packet, even if it is NRT, and Φ represents the general occupation rate for the entire traffic in the CC and the data channels. The model is nonpreemptive; thus,

$$\overline{\mathscr{R}} = \frac{\overline{T}^2}{2\overline{T}} = \frac{\sigma_T^2 + \overline{T^2}}{2\overline{T}} \tag{13}$$

By using Little's Law and when $\overline{L}_{rt}$ represents the mean size of the RT packet queue and equals $\lambda_{rt}\overline{W}_{rt}$, (12) can be rewritten as

$$\overline{W}_{rt}\left(1 - \phi_{rt}\right) = \phi_{rt}\overline{\mathscr{R}} \tag{14}$$

where ϕ_{rt} represents the total occupation rate for RT packets. Thus, the mean waiting time for an RT packet is estimated as

$$\overline{W}_{rt} = \frac{\Phi\overline{\mathscr{R}}}{1 - \phi_{rt}} \tag{15}$$

and the mean throughput time, i.e., the average delay time, is

$$T_{rt} = \overline{W}_{rt} + \overline{T} \tag{16}$$

Considering that a new packet may arrive at any moment during the service time, the mean residual service time can be estimated as

$$\overline{\mathscr{R}} = \frac{1}{2}\left(c_s^2 + 1\right)\overline{T} \tag{17}$$

where c_s is the coefficient of the service variation time and is equal to σ_s/T. Note that, in case of exponential service time, c_s^2 is equal to 1 and in case of deterministic service time it equals 0. For simplicity, $c_s^2 = 0.5$. Thus, (17) can be rewritten as

$$\overline{\mathscr{R}} \approx \frac{3}{4}\overline{T} \tag{18}$$

The mean delay time for the NRT packet can be estimated as

$$\overline{W}_{nrt} = \frac{\overline{W}_{rt}}{1 - \phi} \tag{19}$$

The sum of gaps between the sequential transmissions is considered as wasted time and it is denoted by W_t. W_t includes the contention periods in which the channel is not chosen for submission. Thus,

$$W_t = P_i \sum_{j=1}^{\max F_n} \left|T_{cj} - T_{pj}\right| + \left(1 - P_i\right)\left\lfloor \frac{T_{off}}{\overline{T_c}} \right\rfloor \overline{T_c} \tag{20}$$

where P_i is the probability that the BS will choose this channel for the next submission. The maximum number of transmissions that can be achieved in the channel's idle time, T_{off}, after deducting the time in which the channel was not chosen for the submission, T_{cm}, is given by

$$\max F_n = \frac{T_{off} - T_{cm}}{\max\left(T_c, T_p\right)} \tag{21}$$

Accordingly, (20) can be rewritten as

$$W_t \approx P_i \max F_n \left|T_{cj} - T_{pj}\right| + \left(1 - P_i\right)T_{off} \tag{22}$$

Each channel has the same probability to be chosen among M available channels since a uniform distribution is used to provide the fairness property for the data channels; thus the probability of allocating a channel, P_i, equals $1/M$. Consequently, the available time for data transmission equals $(T_{off} - W_t)$ and the achievable number of packet transmissions is given as

$$NT_p = \left\lfloor \frac{T_{off} - W_t}{T_p} \right\rfloor \tag{23}$$

The net transmission time is estimated as

$$NT_p\left(T_p - SIFS - T_{ack}\right) = NT_pL \tag{24}$$

Therefore, the achieved throughput can be estimated as

$$\Psi_{off} = NT_pLD_{dr} \tag{25}$$

Table 1: Evaluation parameters values.

Parameter	Value
Contention Window min. size CW_{min}	*32*
Contention Window max. size CW_{max}	*1024*
CC Data Rate C_{dr}(Mbps)	*1*
Data Channel Data Rate D_{dr}(Mbps)	*1*
Slot time in CC Channel (ms)	*0.02*
CCA (ms)	*0.02*
DIFS (ms)	*0.05*
SIFS (ms)	*0.01*
T_{rts} (ms)	*0.352*
T_{cts} (ms)	*0.304*
T_{ack} (ms)	*0.304*

Taking into account the probability that a node can transmit in a specific idle time, θ_s, and the CC channel's blocking probability, θ_b, that can be expressed as

$$\theta_s = \exp^{-(N/M)} \quad \text{and } \theta_b = 1 - \frac{\overline{T_c}}{\overline{T}} \tag{26}$$

(25) can be rewritten as

$$\Psi_{off} = (1 - \theta_s\theta_b)\, NT_pLD_{dr} \tag{27}$$

5. Packet Size Impact Evaluation

The evaluation of the impact based on different data packet size was performed using MATLAB. The main evaluation parameters values are listed in Table 1 and they are similar to that defined in IEEE 802.11 MAC. The duty cycle time is assumed to be *one* second.

Figure 3 shows the effects of changing the packet size on the relationship between the node population and the induced delay for the two categories of the packets, RT and NRT. In the simulated scenario, the channel idle time is fixed, $T_{off} = 0.5$ s, number of the channels is 10, and the arrival rate at both of the queues is the same, $\lambda_{rt} = \lambda_{nrt} = 3$ pkt/s. The performance shows linear behavior of the RT packets starting from the low to the high node densities while the performance of the NRT packets behaves exponentially starting from the medium nodes population, $N = 5$. From the figure, the average delay increases as the packet size increases even though the delay values are similar in low nodes density with values less than 0.2 s. The difference of the induced delay values between RT and NRT packets increases as nodes density increases and that can be explained given that the RT packets have the advantage in the CC contention.

In Figure 4, the trend towards the exponential behavior for both RT and NRT packet delay curves is so clear. In this simulation, the node population is fixed at $N = 5$, 3 pkt/s is the arrival rate for each node, and 10 channels are being utilised. In short busy time periods, the values of the average delay are low, below 0.1 s. As the channel's busy time increases, the delay increases for all packet sizes although

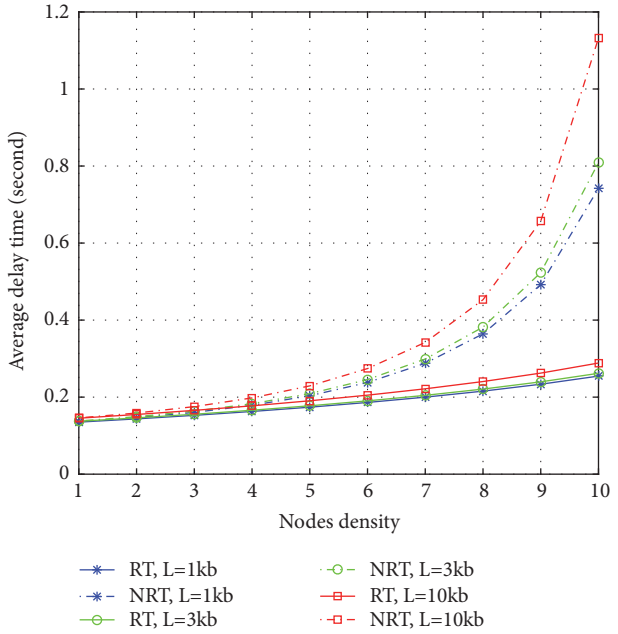


Figure 3: Average delay vs nodes density.

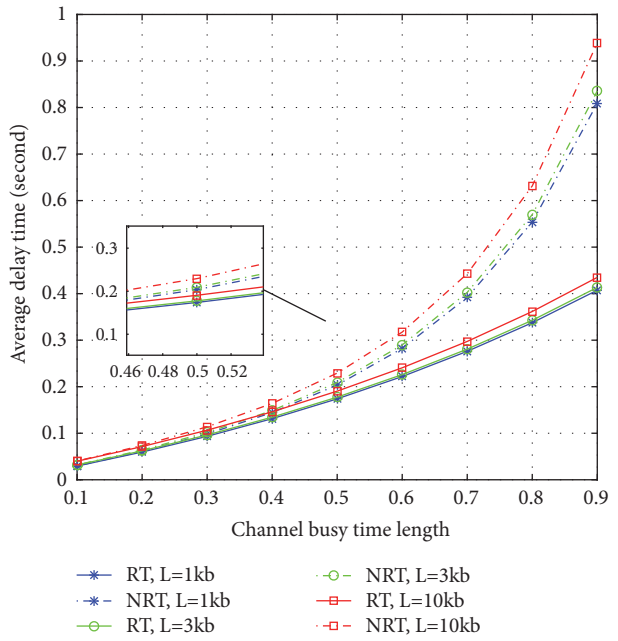


Figure 4: Average delay vs channel's busy time.

the large-size packets are exposed to longer delay. When the packet size is 10 kb and $T_{on} = 0.9$ s, the induced delay records its highest values: 0.43 s and 0.94 s for RT and NRT packets, respectively.

Figure 5 presents the effects of the packet size on the relationship between the number of the available channels and the average delay. Nodes density is fixed at $N = 5$, and channel idle time is $T_{on} = 0.5$ s. The induced delay for NRT packets is very long when the number of the available channels is less than 5 to the extent that the system losses its

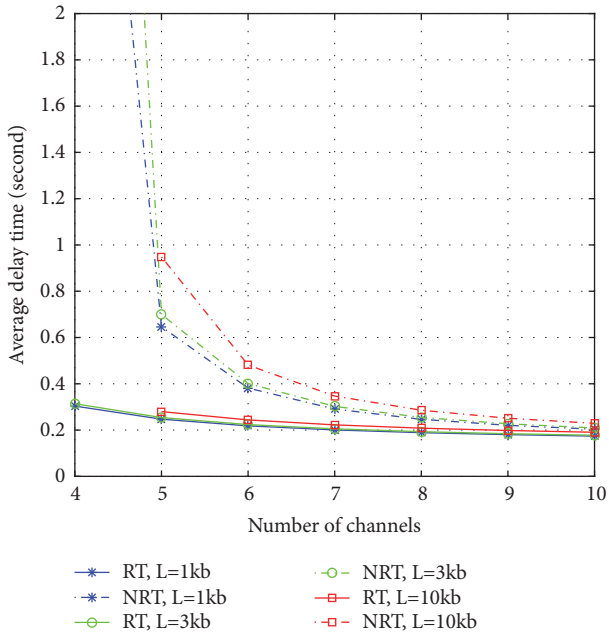


Figure 5: Average delay vs number of channels.

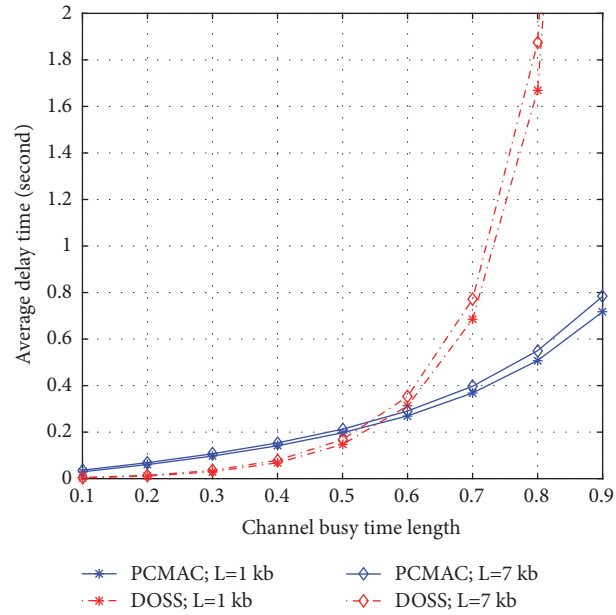


Figure 6: Impact of the packet size on the induced delay in PCMAC [16] and DOSS [17] protocols.

stability with large-size packets, i.e., $L = 10$ kb. The curves of the induced delay fall steeply for NRT packets as the number of channels increases to 5 channels and gradually decline as M exceeds 5 channels. For large-size packets of 10 kb, the declining trend is slower compared to other packet sizes. From the figure, it is obvious that medium-size packets of 3 kb can cope with the miserly licensed radio environment while large-size packets cannot. Resulting interference contributes more to increasing the induced delay for large-size packets since the collided packet has to be retransmitted again.

Figure 6 shows the performances of the PCMAC protocol and DOSS MAC protocol with various sizes of the data packet. The latter protocol was improved and modelled in [17]. Nodes density equals 5 and 7 data channels are exploited. Although DOSS MAC offers a shorter delay time in short lengths of channels busy times, it behaves exponentially in longer busy times starting from 0.4 s. PCMAC presents a more stable performance in both short and long lengths of channel idle times. Generally, the average delay time increases as the packet size becomes larger in both of PCMAC and DOSS protocols.

Figure 7 illustrates that the throughput can be significantly affected by packet size. In the simulation, the idle time and the number of channels are fixed at $T_{off} = 0.5$ and $M = 5$, respectively. The throughput curves growths have an exponential trend for all packet sizes; however the curve of small-size packet, $L = 1$ kb, shows the slowest trend. The impact of the time gap between the contention time T_c and the packet size L is clear in case of $L = 1$ kb. As the number of nodes increases, the contention time increases; i.e., the time gap between T_p and T_c decreases, until it approaches the value of the packet size at which the throughput reaches its peak, in the range of $N = 7$ and $N = 8$, before declining gradually.

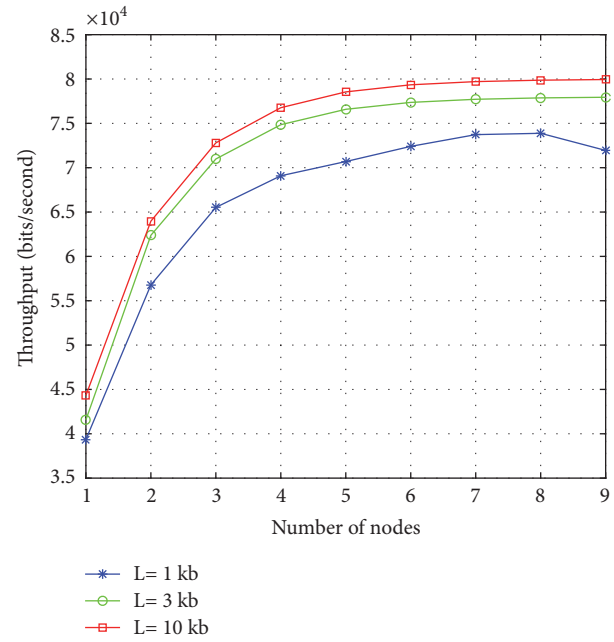


Figure 7: Throughput vs nodes density.

In this simulation, medium-size packets show a much closer performance to that of large-size packets.

In Figure 8, the significant impact of the packet size on the relationship between the channel idle time and the achieved throughput is illustrated. Generally, the throughput increases dramatically as channel idle time increases, but with packets of medium size, $L = 3$ kb, the throughput performance is better compared to other sizes. For packets of 1 kb size, the time gap between the contention time and the packet time plays a main role since the packet

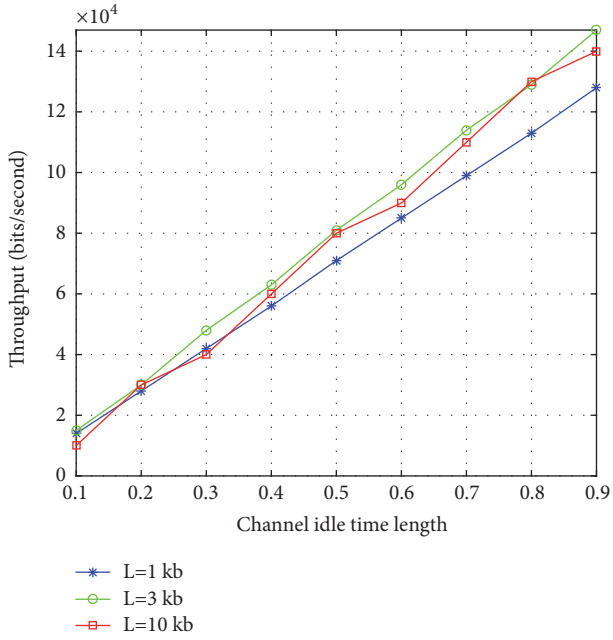


Figure 8: Throughput vs channel's idle time.

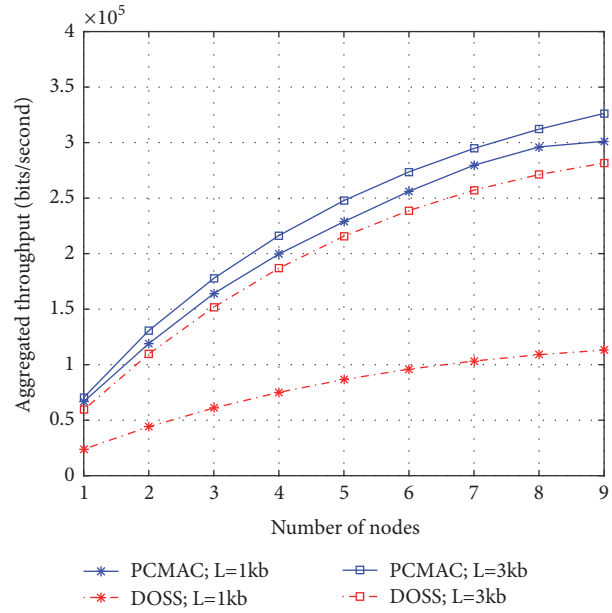


Figure 9: Achieved throughput comparison between PCMAC [16] and DOSS [17].

size determines throughput performance for larger packets. The higher the achieved throughput, the more efficient the utilization of the radio channels. The fluctuation of the throughput curve for packets of 10 kb size is because of the submission nature in which the receiver deals with the data packet as a block. As a matter of fact, the latter feature is also responsible for the high probability of interference leading to longer delay for large-size packets. Because of the burst nature of the traffic in PCMAC protocol, it shows better performance than DOSS MAC protocol in terms of the achievable throughput especially with small-size packets as illustrated in Figure 9. Using medium-size packets, $L = 3$ kb, improves DOSS MAC protocol's performance significantly although the nodes density increasing will result in slowing down the throughput increasing due to the contention in the CC channel. The simultaneous transmission nature, i.e., concurrent submission through many channels, for the packets in DOSS causes the high sensitivity against increasing nodes density.

In general, at the same time the medium-size packets outperform small-size packets in terms of the achieved throughput and channel's utilization; they show closer performances to them in terms of the latency. Moreover, employing packets of size 375 byte can be the optimal choice when using an adaptive architecture for the network such as that proposed in [26]. Packets of medium size that are closer to the common sizes used in WSN standards, up to 128 bytes, ensure a robust performance in the alternative workflow that is CR-based

6. Conclusion

The size of the data packet in CRSN networks plays a key role not only to enhance the throughput and increase the efficiency of channels' utilization, but also to mitigate the harmful interference. This study has examined the impact of packet size on the performance of CRSN in terms of two main performance metrics, namely, the delay and the throughput. Simulation results show that exploiting the appropriate size within an efficient MAC protocol, such as PCMAC, significantly enhances the performance. Medium-size packets outperformance is proven by the short induced latency, which is suitable for critical data, and the increased throughput. Furthermore, the results show that large-size packets not only fail to cope with a poor radio environment, but also do not enhance the throughput significantly in contrast to the packets of medium size that perform better in several conditions. Through this study, it is evident that medium-size packets are the optimal choice for adaptive CRSN networks despite the current trends of using small-size packets.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] S. Farahani, "Zigbee basics," in *ZigBee Wireless Networks and Transceivers*, Elsevier, 1st edition, 2008.

[2] M. Becker, *Services in Wireless Sensor Networks: Modelling and Optimisation for the Efficient Discovery of Services*, Springer Science & Business Media, 2014.

[3] D. Yang, Y. Xu, and M. Gidlund, "Wireless coexistence between IEEE 802.11- and IEEE 802.15.4-based networks: a survey,"

International Journal of Distributed Sensor Networks, vol. 2011, Article ID 912152, 17 pages, 2011.

[4] Federal Communications Commission (FCC), *Docket no 03-222 notice of proposed rule making and order*, 2003.

[5] A. M. Wyglinski, M. Nekovee, and Y. T. Hou, *Cognitive Radio Communications and Networks: Principles and Practice*, Academic Press, 2009.

[6] D. Cabric, S. M. Mishra, and R. W. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in *Proceedings of the Conference Record of the 38th Asilomar Conference on Signals, Systems and Computers*, pp. 772–776, Pacific Grove, Calif, USA, November 2004.

[7] W.-Y. Lee and I. F. Akyildiz, "Optimal spectrum sensing framework for cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 10, pp. 3845–3857, 2008.

[8] G. P. Joshi, S. Y. Nam, and S. W. Kim, "Cognitive radio wireless sensor networks: applications, challenges and research trends," *Sensors*, vol. 13, no. 9, pp. 11196–11228, 2013.

[9] J. Ren, Y. Zhang, N. Zhang, D. Zhang et al., "Dynamic channel access for energy efficient data gathering in cognitive radio sensor networks," https://arxiv.org/abs/1507.06188.

[10] Zilong Jin, Yu Qiao, Alex Liu, and Lejun Zhang, "EESS: An Energy-Efficient Spectrum Sensing Method by Optimizing Spectrum Sensing Node in Cognitive Radio Sensor Networks," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 9469106, 11 pages, 2018.

[11] A. Sawabe, K. Tsukamoto, and Y. Oie, "QoS-aware packet chunking schemes for M2M cloud services," in *Proceedings of the 28th IEEE International Conference on Advanced Information Networking and Applications Workshops, IEEE WAINA 2014*, pp. 166–173, Canada, May 2014.

[12] L. Song, D. Niyato, Z. Han, and E. Hossain, *Wireless Device-to-Device Communications and Networks*, Cambridge University Press, Cambridge, UK, 2015.

[13] N. U. Hasan, W. Ejaz, and H. S. Kim, "Frame size selection in CSMA-based cognitive radio wireless local area networks," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 1, 2017.

[14] G. Durisi, T. Koch, and P. Popovski, "Towards massive, ultra-reliable, and low-latency wireless communication with short packets," https://arxiv.org/abs/1504.06526.

[15] S. Atapattu, C. Tellambura, and H. Jiang, "Conventional Energy Detector," in *Energy Detection for Spectrum Sensing in Cognitive Radio*, pp. 11–26, Springer, New York, NY, USA, 2014.

[16] M. Al-Medhwahi, F. Hashim, B. M. Ali, and A. Sali, "Pliable cognitive MAC for heterogeneous adaptive cognitive radio sensor networks," *PLoS ONE*, vol. 11, no. 6, p. e0156880, 2016.

[17] G. A. Shah and O. B. Akan, "Performance analysis of CSMA-based opportunistic medium access protocol in cognitive radio sensor networks," *Ad Hoc Networks*, vol. 15, pp. 4–13, 2014.

[18] O. Ergul and O. B. Akan, "Cooperative coarse spectrum sensing for cognitive radio sensor networks," in *Proceedings of the 2014 IEEE Wireless Communications and Networking Conference, WCNC 2014*, pp. 2055–2060, Turkey, April 2014.

[19] G. A. Shah and O. B. Akan, "Cognitive adaptive medium access control in cognitive radio sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 2, pp. 757–767, 2015.

[20] L. Ma, X. Han, and C.-C. Shen, "Dynamic open spectrum sharing MAC protocol for wireless ad hoc networks," in *Proceedings of the 2005 1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN 2005*, pp. 203–213, USA, November 2005.

[21] W. Tang, M. Z. Shakir, M. A. Imran, R. Tafazolli, and M.-S. Alouini, "Throughput analysis for cognitive radio networks with multiple primary users and imperfect spectrum sensing," *IET Communications*, vol. 6, no. 17, pp. 2787–2795, 2012.

[22] N. U. Hasan, W. Ejaz, and H. S. Kim, "Frame size selection in CSMA-based cognitive radio wireless local area networks," *Transactions on Emerging Telecommunications Technologies*, 2014.

[23] T. O. Kim, A. S. Alfa, and B. D. Choi, "Performance Analysis of a CSMA/CA Based MAC Protocol for Cognitive Radio Networks," in *Proceedings of the 2010 IEEE Vehicular Technology Conference (VTC 2010-Fall)*, pp. 1–5, Ottawa, ON, Canada, September 2010.

[24] K. J. Kim, K. S. Kwak, and B. D. Choi, "Performance analysis of opportunistic spectrum access protocol for multi-channel cognitive radio networks," *Journal of Communications and Networks*, vol. 15, no. 1, pp. 77–86, 2013.

[25] B. Gulbahar and O. B. Akan, "Information theoretical optimization gains in energy adaptive data gathering and relaying in cognitive radio sensor networks," *IEEE Transactions on Wireless Communications*, vol. 11, no. 5, pp. 1788–1796, 2012.

[26] M. Al-Medhwahi and F. Hashim, "The adaptive Cognitive Radio Sensor Network: A perspective towards the feasibility," in *Proceedings of the 1st International Conference on Telematics and Future Generation Networks, TAFGEN 2015*, pp. 6–11, Malaysia, May 2015.

[27] IEEE 802 Working Group, "Ieee standard for local and metropolitan area networks" part 15.4: Low-rate wireless personal area networks (lr-wpans)," *IEEE Std*, vol. 802, pp. 4–2011, 2011.

[28] P. Huang, L. Xiao, S. Soltani, M. W. Mutka, and N. Xi, "The evolution of MAC protocols in wireless sensor networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 101–120, 2013.

[29] IEEE Standards Association, "Part 22: Cognitive wireless ran medium access control (mac) and physical layer (phy) specifications: Policies and procedures for operation in the tv bands," *IEEE Standard*, vol. 802, 2011.

[30] H. Zhao, E. Garcia-Palacios, J. Wei, and Y. Xi, "Accurate available bandwidth estimation in IEEE 802.11-based ad hoc networks," *Computer Communications*, vol. 32, no. 6, pp. 1050–1057, 2009.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 6951318, 12 pages
https://doi.org/10.1155/2018/6951318

Research Article

Analyzing Critical Failures in a Production Process: Is Industrial IoT the Solution?

Shafiq Ahmad, Ahmed Badwelan, Atef M. Ghaleb, Ammar Qamhan, and Mohamed Sharaf

King Saud University, College of Engineering, Department of Industrial Engineering, Riyadh 11421, Saudi Arabia

Correspondence should be addressed to Shafiq Ahmad; ashafiq@ksu.edu.sa

Received 11 May 2018; Revised 5 August 2018; Accepted 10 October 2018; Published 3 December 2018

Guest Editor: Dongyao Jia

Machine failures cause adverse impact on operational efficiency of any manufacturing concern. Identification of such critical failures and examining their associations with other process parameters pose a challenge in a traditional manufacturing environment. This research study focuses on the analysis of critical failures and their associated interaction effects which are affecting the production activities. To improve the fault detection process more accurately and efficiently, a conceptual model towards a smart factory data analytics using cyber physical systems (CPS) and Industrial Internet of Things (IIoTs) is proposed. The research methodology is based on a fact-driven statistical approach. Unlike other published work, this study has investigated the statistical relationships among different critical failures (factors) and their associated causes (cause of failures) which occurred due to material deficiency, production organization, and planning. A real business case is presented and the results which cause significant failure are illustrated. In addition, the proposed smart factory model will enable any manufacturing concern to predict critical failures in a production process and provide a real-time process monitoring. The proposed model will enable creating an intelligent predictive failure control system which can be integrated with production devices to create an ambient intelligence environment and thus will provide a solution for a smart manufacturing process of the future.

1. Introduction

Manufacturing companies traditionally are the largest multipliers of economy, employment provider, and GDP contributor. Due to such a critical position in any country's economy, maintaining operational efficiency of a manufacturing concern is crucial. This also results in obtaining higher value products and gaining satisfaction and content of the consumers. Thus, in today's manufacturing environment, maintaining machines and production equipment plays an important role which directly affects the service life of equipment and its production efficiency [1]. The organizational environment is also affected by how a company plans, utilizes and maintains its facilities and manufacturing process in order to enhance its market shares and consumer appreciation [2]. Machine failures are undesirable to any manufacturing company as they result in low production rates and lead to inferior quality. In addition, failures make it difficult for manufacturers to accomplish their commitments to consumers.

It is always vital to maintain production throughput rates and a higher level of product quality. For this reason, an effective failure control system must be implemented to ensure smooth production operations in any company [3]. Process owners should practice a systematic way to study and detect machine failures and their associated causes. The reason is that, in some cases, machine failures can lead to a breakdown situation where machines will be unable to perform any further according to the operational specifications. Some machine failures might be critical to safety and cause very harmful accidents, especially those with hazardous materials. Some failures in manufacturing facilities might be internal as discussed above; however; some factors are external which could positively or negatively affect production operations [4]. Factors such as product demand, global warming, and production breakdowns are just a few examples in the present

scenario. The more knowledge gained about internal failures as well as the knowledge of external factors will lead to a stable and smooth production operation.

In addition, having knowledge about all (major and minor) failures is considered a valuable input for developing an efficient and effective maintenance program [5]. This is why it is crucial for a manufacturing concern to adopt an effective maintenance strategy to augment production, decrease machine downtime, and consider corrective and preventive maintenance [6]. Subsequently, facility planning facilitates identifying breakdown occurrences and provides and easy access for maintainability of a production process. Facility planning also clarifies what types of repair activities are required. Strategies should be designed and adopted to minimize travelling distances as well as minimize the cost required. This will result in improving the manufacturing processes and reducing their failures in order to achieve the least abruptions in production processes. Raman et al. in their research study argue that facility design depends mainly on the cost of facility construction and the cost of transportation. Further, minimizing the travelling distances can be achieved by better facility layouts which can yield significant impact in reducing machine failures in any manufacturing concern [2].

It is an established fact that study of machine failures and their effect on production processes is popular among scientists and researchers [7]. Maintenance policies and their applications which have significant effect on failure durations are also investigated by many researchers in [8, 9]. Problems of testing the homogeneity of critical machine failures such as mean time between failures (MTBF) are also discussed by Pandey and Singh, 2000 [10]. Some studies have discussed how failure analysis could support in building an effective layout model and improve operational efficiency and manufacturing environments [11, 12]. Bröchner highlighted the significance of the relationship between design and facilities management [13], while Jones and Sharp discussed the importance of integrating business strategies in any maintenance program [14]. Most recently, Moohialdin and Hadidi, 2016 [15], discussed the effects of failure types on downtime durations.

The behaviour of machine failures and their associated relations with failure causes using experimental techniques is a well-established research area. Roy and Sutapa, 2004 [16], conducted factorial and fractional factorial experimental techniques to analyze the machine failure data and presented its research outcomes. Another case study is described by Adhikary, 2014 [17], which involves the application of experimental design to reduce machine cycle times in order to eliminate unevenness in the cycle times of two operations. Shen and Wan studied simulation modelling and provided an in-depth evaluation of performance of an existing manufacturing system for a serial production line in a printed circuit board factory. Their simulation analysis is based on full factorial design approach [18]. The most important factors that contribute significantly to operational problems are also studied by Chan and Chan, 2003 [19]. EK used factorial design to conduct experiments where the flow process of boiling and the factors parameters that influence it were studied [20]. Several such studies using experimental design techniques are presented in the literature ([21], (Christen and Soccol, 2000), [22]).

This research study focuses on analysis of machine failure causes. Unlike other research studies presented in the scientific literature which are mainly dedicated to mechanical and electrical failure causes, this study motivates us to investigate the main causes of failures (COF) due to material deficiency and production organizational environment which are adversely affecting the manufacturing processes. To conduct this study, real-time data is collected and analyzed using experimental design methodology and results are presented to help industry practitioners to preclude such critical failures affecting negatively. This case study will also support professionals in controlling machine failures, based on failure prevention strategy rather than failure prediction strategy.

To adapt with the newly introduced smart manufacturing process technologies which are capable of failure prediction rather than prevention strategy, such as Industrial Internet of Things (IIoT) and cyber physical systems (CPS) technologies, we have proposed a conceptual model for detection of critical machine failure more promptly and efficiently. Today's modern era demands a failure prediction strategy and for this purpose, a new approach which can support failure control system with a high level of accuracy and fast delivery is proposed. A smart factory which uses emerging technologies such as IoT and CPS is proposed in this research which will yield significant reduction in failures, enhancing productivity and process efficiency for the entire manufacturing industry [23].

Due to advancement of IoT and other smart sensor technologies, industries are now able to seize the large amounts of data [24] in an industrial process with affordable operational cost. This has motivated industry practitioners to adapt traditional process monitoring functions with IIoT and other smart technologies. This trend is revolutionizing the industrial operations and is also introduced as Industry 4.0. It will help manufacturers to improve production efficiency and product quality [25]. Many researchers have recently contributed to the development of smart manufacturing systems and the data analytics using IIoT based solutions for machine condition monitoring and detection of any failures on real-time bases. For instance, Guo and Qui [26] discussed the new generation of information technology as cloud computing, big data, IoT, and artificial intelligence (AI) and how it became the source of changes in the modern smart manufacturing and production processes. Zhong et al., 2017 [27], also reviewed the Industry 4.0 concept by describing worldwide movements in intelligent manufacturing.

This paper is organized as follows: Section 2 describes the methodology to be used in the study. A description of the conceptual framework of a smart factory is presented in Section 3. Sections 4 and 5 present the data collection and data evaluation, respectively. Section 6 discusses the results. Details of the proposed smart factory model are illustrated in Section 7 and finally, the conclusions and recommendations for future research are detailed in Section 8.

2. Methodology

Machine failures affect a company's performance. If the machine failures are predominant, then companies will be unable to continue their operations appropriately and will not be able to meet customer demands on time. This research will focus on studying failure patterns and the source cause of these (critical) failures using root-cause analysis. The critical failures (CFs) affect mean-time-between-failure (MTBF) in any manufacturing operation which is a basic measure of a system's reliability and availability. It is defined as the expected elapsed time between two failures of a system which take place throughout normal system operation. It is calculated as an average time between failures of a system. It is normally expressed in units of hours. The higher the MTBF number, the longer the reliability of the system [28]. The failure durations (facility downtime) based on the failure type are also considered in this study. A factorial design model to study behaviour of CFs is developed. Factorial design is a well-established analytical approach, which consists of experimental analysis of two or more failures (factor), each factor with discrete possible values having all possible combinations of these values across all CFs considered. For any process improvement endeavour, factorial design is probably the most powerful experimental and statistical technique for conducting research [20]. Such an experimental technique allows the investigator to study the effect of each factor (failure) on the response variable (i.e., MTBF in this study), as well as the interaction effects that occurred between factors on the response variable.

The experimental methodology is as follows:

(i) Data collection: daily production reports data (secondary) and primary data by conducting several interviews with production staff and maintenance crew are collected for this study.

(ii) Data verification: data verification is undertaken by comparing the production reports data with the machine's time log sheet data.

(iii) Model development: collected data is tested for normality assumption. If the collected data fails to follow normality, an appropriate mathematical transformation technique is used to tackle the issue of nonnormality of the data.

(iv) Checking the model adequacy.

(v) Conducting the experimental analysis using full factorial design approach and investigating which machine failure is dominant (critical) compared to other failures included in the analysis.

(vi) Simultaneous testing: multiple comparisons of results are conducted to determine which means (critical failures) are significantly different among each other.

(vii) Validation: validation of analysis results is conducted by having discussions with industry experts and plant management and results (if needed) are revised afterwards by the stakeholders.

(viii) Future direction: by using smart devices, a smart factory conceptual model is proposed which will help to detect the critical failures more efficiently and promptly.

3. Conceptual Framework of a Smart System

A conceptual framework towards a smart manufacturing system is proposed for future. It will comprise three levels which are illustrated in Figure 1.

Level 1: Sensory Devices. Level one will include sensors and cameras which are required for failure detection and data collection as well as alerting if any abnormality occurs during machine functioning and production operations. For instance, thermal cameras and sensors can detect different temperatures; the cameras will alert when any abnormal action occurs.

Level 2: Data Storing and Sharing. A cloud platform will be employed to store large data and therefore allowing for easier access and sharing. The database technology in a cloud platform is used for data storage and analysis. The cloud service system provides data channels for information exchange between the levels. The cloud based systems can provide massive storage resources and low cost computing as well as the flexibility of customizing the operating environment [29]. Thus, data generated by the equipment can be collected and analyzed and messages are sent to the user level. Programmers and developers will be needed to preprogram the optimization unit [30].

Level 3: User Interface. A convenient means of interaction and an appropriate communication platform are required at this level. For a computer network, a software development system can be installed. In order to achieve the required mobile services, we must carry out the development of mobile application software. Therefore, the same system can be flexibly selected and adapted to a mobile terminal. In this process, a terminal device must use Wi-Fi or a 3G/4G signal to connect to the network and access the corresponding Internet cloud service system through the HTTP protocol [31].

Besides many other merits, this proposed system will help in reducing failures and predicting them in advance. This will help to bring down MTBF and to achieve higher productivity and efficiency. Most of the manual work will be replaced by the above-mentioned automated systems, allowing for faster planning as well. Although these systems are cost effective in the long run, initial investment is a bit high. Moreover, the system can be accessed from all networks, making it user friendly anywhere and everywhere.

Limitations. Like any other system, some limitations may be encountered. For example, without motivating and providing any incentives, workers may not be eager to learn and use smart systems correctly and precisely. Machines part failures might occurs and need on time replacements; conversely machines might be unable to deliver production targets.

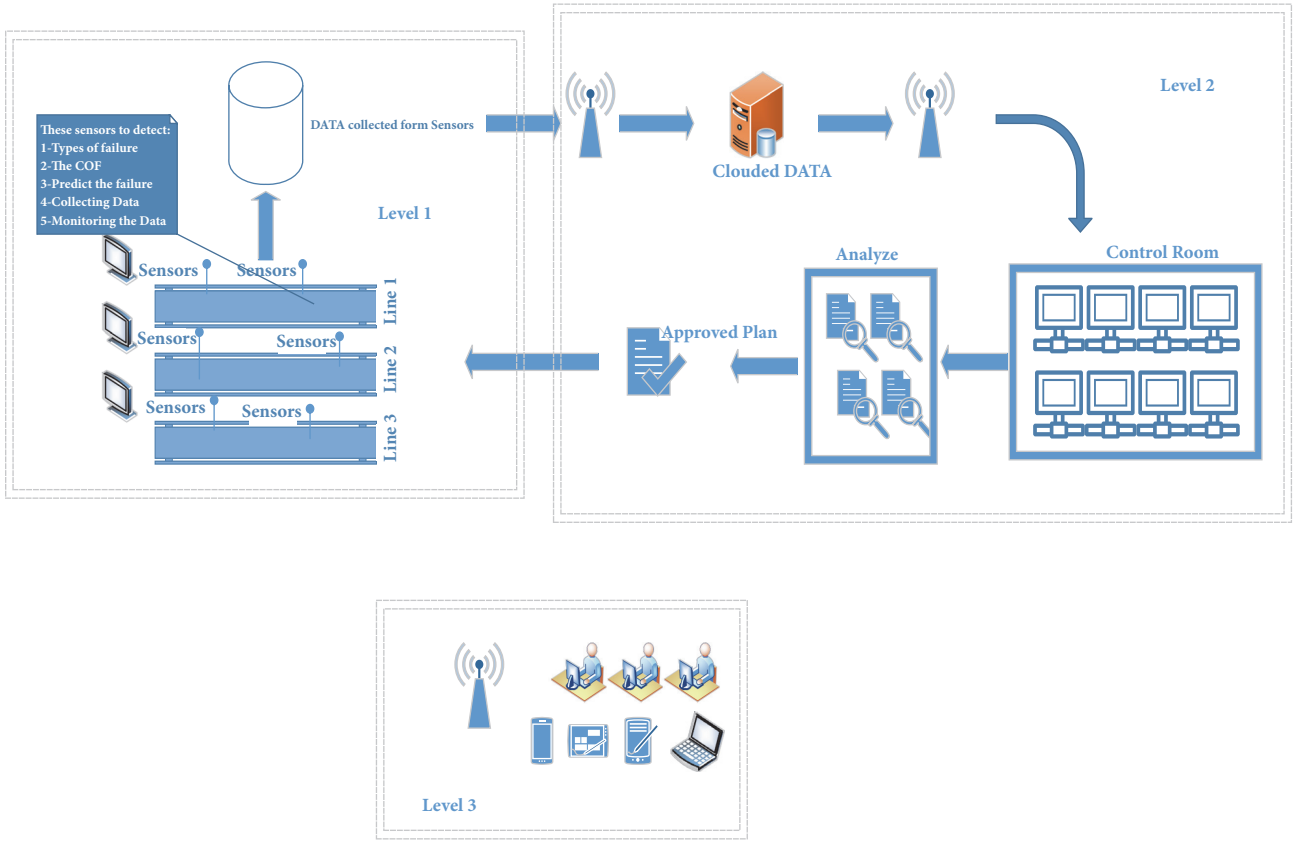


Figure 1: Conceptual framework of a smart manufacturing system.

Finally, network systems may have outages, which might delay functioning. These limitations should be taken into consideration during system design and implementation.

4. Data Collection

A large-scale manufacturing company is selected to conduct this research study. The company produces beverages in several production plants throughout Saudi Arabia with an average annual production capacity of 343,040 metric tons and staff size of 1,400 employees. The company is facing different types of machine failures which cause operational breakdown resulting loss of productivity and throughputs. Machine failures, for instance, mechanical failures, are related to machine components and parts and occurs during machine functioning in the bearings, conveyers, pulleys, rotating shafts, hydraulic and lubricating systems, etc.; electrical failures are related to outage of an electricity supply or a blackout resulting from an electrical breakdown or failure of circuit breakers, etc.; material deficiency failures are related to shortage of raw material or to the material which are available but does not conform to required specification standards; production organization failures are related to any failure in the plant organization which affects production activity such as unavailability of working staff; and planning failures are related to problems which occur due to inadequate planning such as lack of material due to delay in ordering or improper scheduling. A research team was created to approach the facility and collect data about the manufacturing process as well as its failures.

5. Data Analytic

The research team reviewed the company's production reports data. The data was collected by conducting interviews involving questions whose answers will support in defining the potential failure areas in the company's equipment and machinery. Facility site visit and taking notes for all observations were also part of the data collection process. Failure types were finally categorized into five factors mentioned earlier: mechanical, electrical, material deficiency, production organization, and planning. Abbreviations used in this study are listed in Table 1.

Hypotheses (H_0, H_1) were generated to complete this research study. H_0 stated that different failures had no significant impact on response, i.e., MTBF, whereas the alternative hypothesis (H_1) stated that at least one of the failures had a significant impact on MTBF. By applying a significant level of $\alpha = 0.05$, the above suggested hypotheses were tested using Minitab version 17.1.0. The model we used included three factors: production type (PT), cause of failure (COF), and production line (PL). The collected data is presented in Table 2.

Table 1: Abbreviations.

Failure Causes	Abbreviation
Mean time Between Failure	MTBF
Critical to quality	CTQ
Production Type	PT
Production Line	PL
Cause of Failure	COF
Mechanical	Failure 1
Electrical	Failure 2
Material defects	Failure 3
Production organization	Failure 4
Planning defect	Failure 5
Repetition	REP

Data in Table 2, for instance, MTBF (i.e., response), described the time in hours between two failures of the same type. PL, PT, and COF are critical to quality (CTQ) factors. Line 1 and 2 are the production lines in the factory. The products 1, 2, and 3 are different beverage type products. To understand the data with an illustrative example, consider a mechanical failure (Failure 1) is 8.8 hours per week which is recorded as MTBF in line 1 (PL1), Shift 1 for the product 1. Further, factorial analysis is conducted to discover which factors are affecting the production rate significantly. These factors are products (PT), causes of failure (COF), and production line (PL). Furthermore, daily working shift is considered as the blocking factor in this study. As shown in the variable factorial design for model creation, for 3 selected factors with full factorial design (23), 8 runs or more are required. Even though full factorial design requires many runs, it was used to ensure the best results for each factor level and combination. Full factorial design provides complete information unlike to the fractional factorial design which might not provide full information regarding all factors due to the confounding effect.

Model Adequacy Check. To determine whether the model is fitted appropriately to the data; model adequacy test is conducted; residual analysis output plots are presented in Figure 2 which concludes that model adequacy check is successful for further analysis. One can observe that the residuals are distributed normally; residuals versus fits are also randomly scattered; and there is no visible pattern which means that the residuals have a constant variance.

In Figure 2, the histogram of the residuals illustrates that data is slightly positively skewed. However, the residuals versus order of data plot indicates that the residuals are randomly scattered and are equally above and below zero which means that the residuals are uncorrelated with each other. After successfully testing the model adequacy further analysis can be proceeded.

6. Discussion of Results

Factorial analysis results are presented in Table 3. It is customary that if the p-values are greater than α value, which is 0.05 (significance level), then there is no effect on the critical factor and vice versa. In Table 3, for instance, three critical factors (PL, PT, and COF) all have p-value < 0.05 which is strong evidence that there is a significance effect of these factors at the MTBFs.

Further interactions effects are also studied and presented in Figure 3. The relationship between one categorical factor and a continuous response which is MTBF depends on a second categorical factor value. For instance, considering first column in Figure 3 where product line (PL) represents one categorical variable which has 2 manufacturing lines, 1 and 2, interaction effects of other 2 categorical variables product type (PT), and causes of failure (COF) with respect to product lines is studied. The relationship between PL and PT in the first column indicates that there is no significant interaction effect; no lines are crossing each other and all are parallel to each other. On the contrary, the relationship between PL and COF in the first column shows a significant interaction effect for Failure 2 (which is electrical failure) with product line 2 (PL2). Similarly, other 2 factors (PT and COF) in the proceeding columns in Figure 3 can be interpreted. It can be easily observed that in the last column the interaction effect is significant between PT and COF, more precisely COF (Failure 2 (electrical)) on product type (PT) 2.

A statistical model is developed as

$$\begin{aligned}\text{MTBF} = {} & 7.053 + 0.0\ \text{PL}(1) + 1.808\ \text{PL}(2) \\ & + 0.0\ \text{PT}(1) - 0.715\ \text{PT}(2) + 1.052\ \text{PT}(3) \\ & + 0.0\ \text{COF}\ (\text{Failure1}) \\ & + 4.65\ \text{COF}\ (\text{Failure2}) \\ & + 0.78\ \text{COF}\ (\text{Failure3}) \\ & + 1.13\ \text{COF}\ (\text{Failure4}) \\ & + 1.12\ \text{COF}\ (\text{Failure5})\end{aligned} \tag{1}$$

It represents the relationships and interaction effects between CFs and the COFs and an improved model is presented as

$$\begin{aligned}MTBF = {} & 7.053 + 1.808L_2 - 0.715P_2 + 1.052P_3 \\ & + 4.65F_2 + 0.78F_3 + 1.13F_4 + 1.12F_5\end{aligned} \tag{2}$$

Note: notations used in (2) *are* $L_i = PL$, $P_i = PT$, $F_i = COF(Failure)$.

In Figure 4, the main effects plot and other interval plots are presented. The main effects plot displays the means for each group within a categorical variable. From the plot, we can see that for all three factors there exists a main effect. For the production line (PL) factor, the MTBF has a higher response in PL2 than PL1. In the PT factor, MTBF response shows a significant difference at PT1, PT2 and PT3; however, the highest response is at PT3. In the third factor, cause of failure (COF), Failure 2, has the greatest main effect since its slope is the highest.

Figure 4 also shows interval plots, which is the graphical summary of the distribution of a sample showing its central tendency and variability. As mentioned earlier, Failure 2 has the highest impact on the MTBF.

Table 2: Experimental data.

Rep	PL	PT	COF	MTBF	Rep	PL	PT	COF	MTBF	Rep	PL	PT	COF	MTBF
1	1	1	Failure 1	8.8	2	1	1	Failure 1	8.6	3	1	1	Failure 1	7.3
1	1	1	Failure 2	9.6	2	1	1	Failure 2	8.0	3	1	1	Failure 2	11.3
1	1	1	Failure 3	8.5	2	1	1	Failure 3	10.1	3	1	1	Failure 3	12.7
1	1	1	Failure 4	4.3	2	1	1	Failure 4	11.2	3	1	1	Failure 4	8.7
1	1	1	Failure 5	7.6	2	1	1	Failure 5	8.1	3	1	1	Failure 5	6.4
1	1	2	Failure 1	10.2	2	1	2	Failure 1	7.7	3	1	2	Failure 1	6.6
1	1	2	Failure 2	7.8	2	1	2	Failure 2	4.6	3	1	2	Failure 2	6.7
1	1	2	Failure 3	7.2	2	1	2	Failure 3	6.3	3	1	2	Failure 3	8.3
1	1	2	Failure 4	13.9	2	1	2	Failure 4	14.2	3	1	2	Failure 4	5.6
1	1	2	Failure 5	7.6	2	1	2	Failure 5	7.0	3	1	2	Failure 5	8.1
1	1	3	Failure 1	7.6	2	1	3	Failure 1	5.6	3	1	3	Failure 1	7.0
1	1	3	Failure 2	8.8	2	1	3	Failure 2	8.8	3	1	3	Failure 2	8.0
1	1	3	Failure 3	8.1	2	1	3	Failure 3	9.0	3	1	3	Failure 3	12.6
1	1	3	Failure 4	9.9	2	1	3	Failure 4	7.7	3	1	3	Failure 4	9.4
1	1	3	Failure 5	9.8	2	1	3	Failure 5	14.1	3	1	3	Failure 5	12.4
1	2	1	Failure 1	13.2	2	2	1	Failure 1	7.5	3	2	1	Failure 1	8.7
1	2	1	Failure 2	15.4	2	2	1	Failure 2	16.0	3	2	1	Failure 2	16.4
1	2	1	Failure 3	9.6	2	2	1	Failure 3	5.8	3	2	1	Failure 3	6.8
1	2	1	Failure 4	6.5	2	2	1	Failure 4	11.3	3	2	1	Failure 4	8.5
1	2	1	Failure 5	9.6	2	2	1	Failure 5	6.9	3	2	1	Failure 5	11.5
1	2	2	Failure 1	7.0	2	2	2	Failure 1	6.3	3	2	2	Failure 1	8.3
1	2	2	Failure 2	16.1	2	2	2	Failure 2	18.6	3	2	2	Failure 2	11.4
1	2	2	Failure 3	10.4	2	2	2	Failure 3	6.5	3	2	2	Failure 3	4.3
1	2	2	Failure 4	7.0	2	2	2	Failure 4	9.0	3	2	2	Failure 4	10.4
1	2	2	Failure 5	6.9	2	2	2	Failure 5	10.4	3	2	2	Failure 5	8.9
1	2	3	Failure 1	8.4	2	2	3	Failure 1	7.9	3	2	3	Failure 1	8.8
1	2	3	Failure 2	23.4	2	2	3	Failure 2	22.1	3	2	3	Failure 2	16.0
1	2	3	Failure 3	11.4	2	2	3	Failure 3	13.0	3	2	3	Failure 3	8.7
1	2	3	Failure 4	7.4	2	2	3	Failure 4	11.1	3	2	3	Failure 4	9.4
1	2	3	Failure 5	11.9	2	2	3	Failure 5	12.2	3	2	3	Failure 5	6.0

Table 3: Analysing the significance of critical factors.

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Model	31	806.73	26.02	4.99	0.00
Blocks	2	8.6	4.30	0.82	0.44
Linear	7	354.71	50.67	9.71	0.00
PL	**1**	**73.57**	**73.57**	**14.1**	**0.00**
PT	**2**	**47.37**	**23.68**	**4.54**	**0.02**
COF	**4**	**233.77**	**58.44**	**11.2**	**0.00**
2-Way Interactions	14	383.83	27.42	5.25	0.00
PL∗PT	2	6.97	3.49	0.67	0.52
PL∗COF	4	304.97	76.24	14.61	0.00
PT∗COF	8	71.88	8.99	1.72	0.11
3-Way Interactions	8	59.59	7.45	1.43	0.21
PL∗PT∗COF	8	59.59	7.45	1.43	0.21
Error	58	302.67	5.22		
Total	89	1109.41			

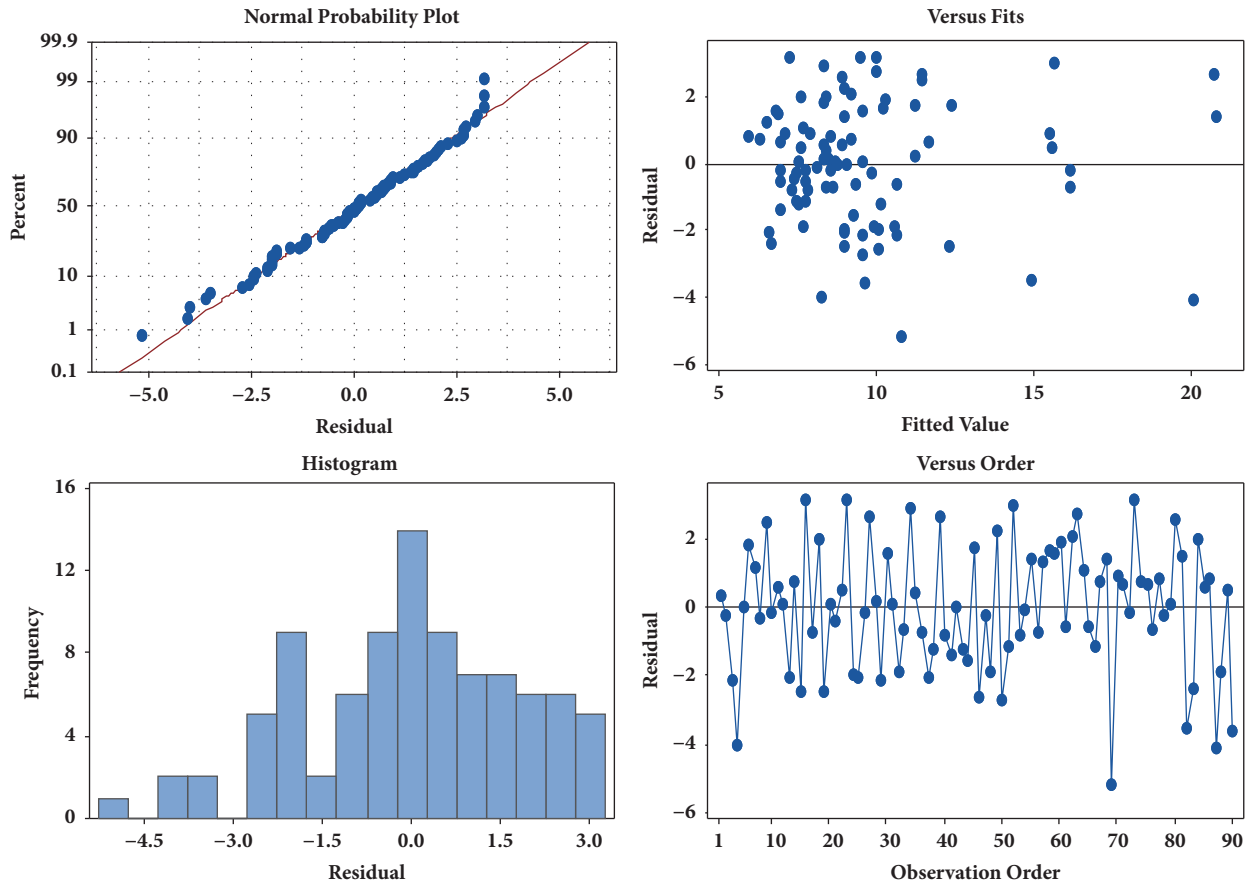


Figure 2: Residual plots for testing model adequacy.

Figure 5 illustrates Tukey Simultaneous 95% confidence intervals test, which supports in finding the means of data that are significantly different from each other. If an interval does not contain zero, the corresponding means are significantly different. When a significant difference is found, it indicates that the corresponding response will be significantly affected.

In Figure 5, dotted lines are indicating that Failure 2 is significantly affecting all the other COFs which includes Failure 1, Failure 3, Failure 4, and Failure 5. From the above results and discussions, it can be concluded that electrical failures (Failure 2) are affecting the most to the model factors (PL, PT, and COF). This is reflected using Tukey differences of means test for MTBF.

In the subsequent sections, details of the proposed conceptual model for a smart manufacturing process will be discussed.

7. Towards a Smart System

It is a fact that machine failures interrupt production operations causing barricades in meeting consumer demands. So, it is crucial for manufacturing companies to develop and implement an appropriate machine failure control system in order to avoid machine breakdowns and their financial impacts. Since the communication network and sensory technologies are growing rapidly, future manufacturing factories need to embrace them promptly in order to support noninterrupted and smooth production operations. This direction is heading towards inception of fourth industrial revolution called Industry 4.0 [32].

Industry 4.0 era will utilize the Industry Internet of Things (IIoT), which is the interconnection via the Internet of computing devices embedded in everyday objects, enabling them to send and receive data, to create a smart factory environment. The smart factory is the integration of all IoT technological with computer networks, data integration, and analytics to bring clarity to all manufacturing factories [33]. Smart factories have proven to achieve transparency as well as productivity [34]. To implement this smart factory, manufacturing companies use cyber physical systems (CPS) which is a term describing a broad range of complex, multidisciplinary, physically aware next generation engineered system that integrates embedded computing technologies (cyber part) into the physical world [35].

CPSs have great potential for social and economic impact. For example, a CPS-based smart factory can increase energy efficiency of production systems at such factories by providing them with decision making at production management level (e.g., production scheduling and maintenance management) [32]. According to a survey by American Society for Quality (ASQ) in 2014, 82% of organizations which

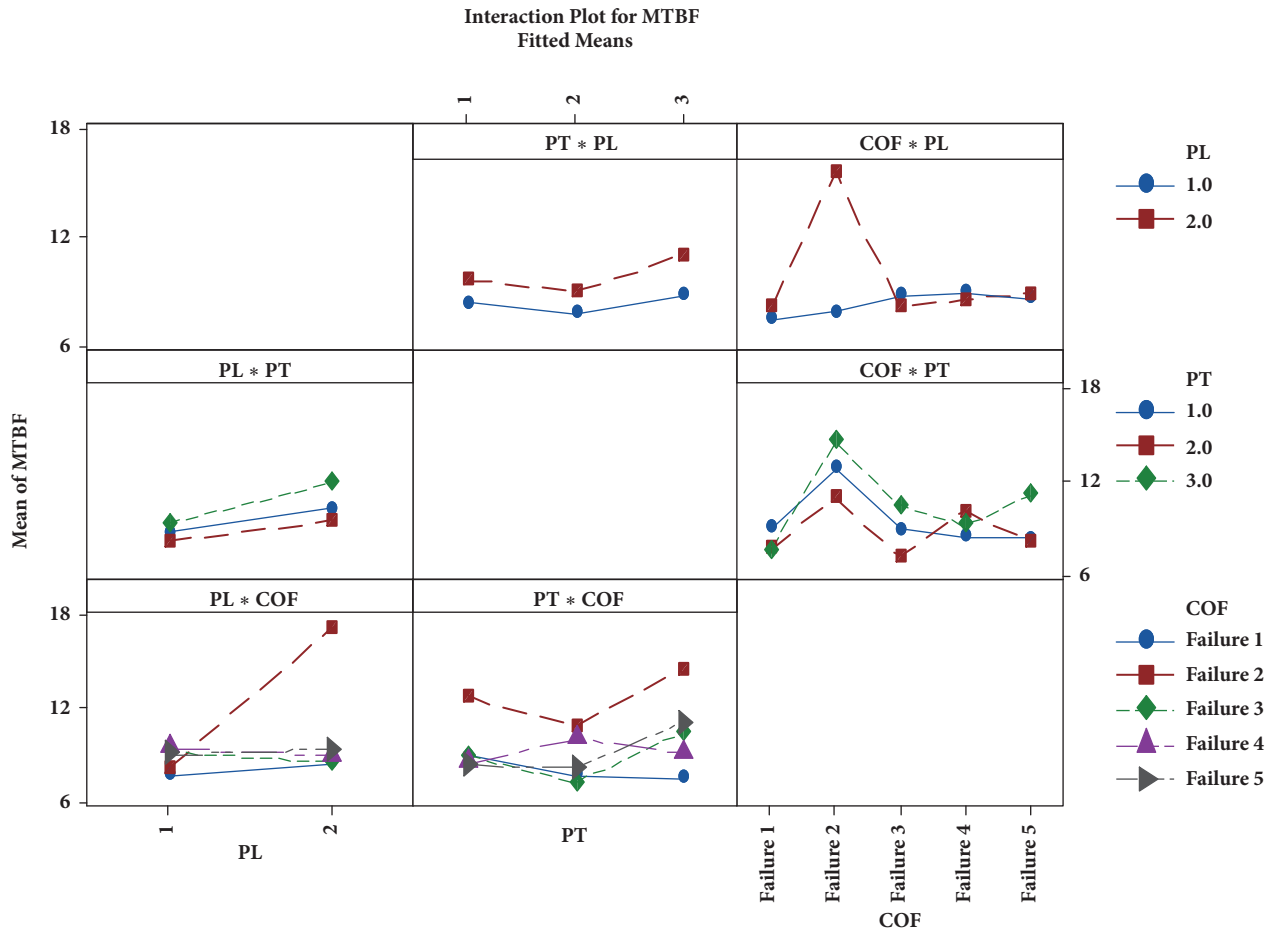


Figure 3: Interaction plots.

claim to have implemented smart manufacturing systems have experienced increased efficiency, 49% experienced fewer product defects, and 45% experienced increased customer satisfaction [36].

7.1. Towards Smart Manufacturing. Monitoring production systems and collecting performance data in real time using smart sensor and communication technologies will have positive impact on maintenance planning and improved production efficiency. For instance, using temperature sensors will allow preemptive actions to be taken when it goes out of range and prevent breakdown. Similarly, preemptive actions can be taken when energy consumption jumped above than normal level for a certain period for time. This will save energy, reduce wastage of defective products, and avoid machine breakdowns. Furthermore, IoT-enabled vision devices will enable machines to predict machine failures and trigger maintenance actions promptly. Thus, the companies which will embrace smart factory concept will benefit high efficiency, less failures due to predictive maintenance strategies [37, 38].

In the subsequent sections, a smart factory design using CPS and IoT is discussed.

Some modifications might be required to adapt with the existing manufacturing operations. The proposed smart manufacturing system will consist of the three levels as depicted in Figure 6. Each layer of the system depends on corresponding communication technologies [32] and protocols to communicate with the other layers and coordinate production activities.

7.1.1. Smart System Structure

Level 1: Physical Level. This level consists of three different sections which are connected using sensors enabling sensing, communication, and computing (processing; memory) elements in the proposed model. Three levels are discussed as follows.

(A) Materials. Materials which are processed in the manufacturing process are of two types:

(i) *Raw Materials.* This kind of materials will be used to manufacture the finished products. In this section, the layer is responsible to detect if there is sufficient raw material inventory available to continue operation without facing shortage. For instance, if the raw

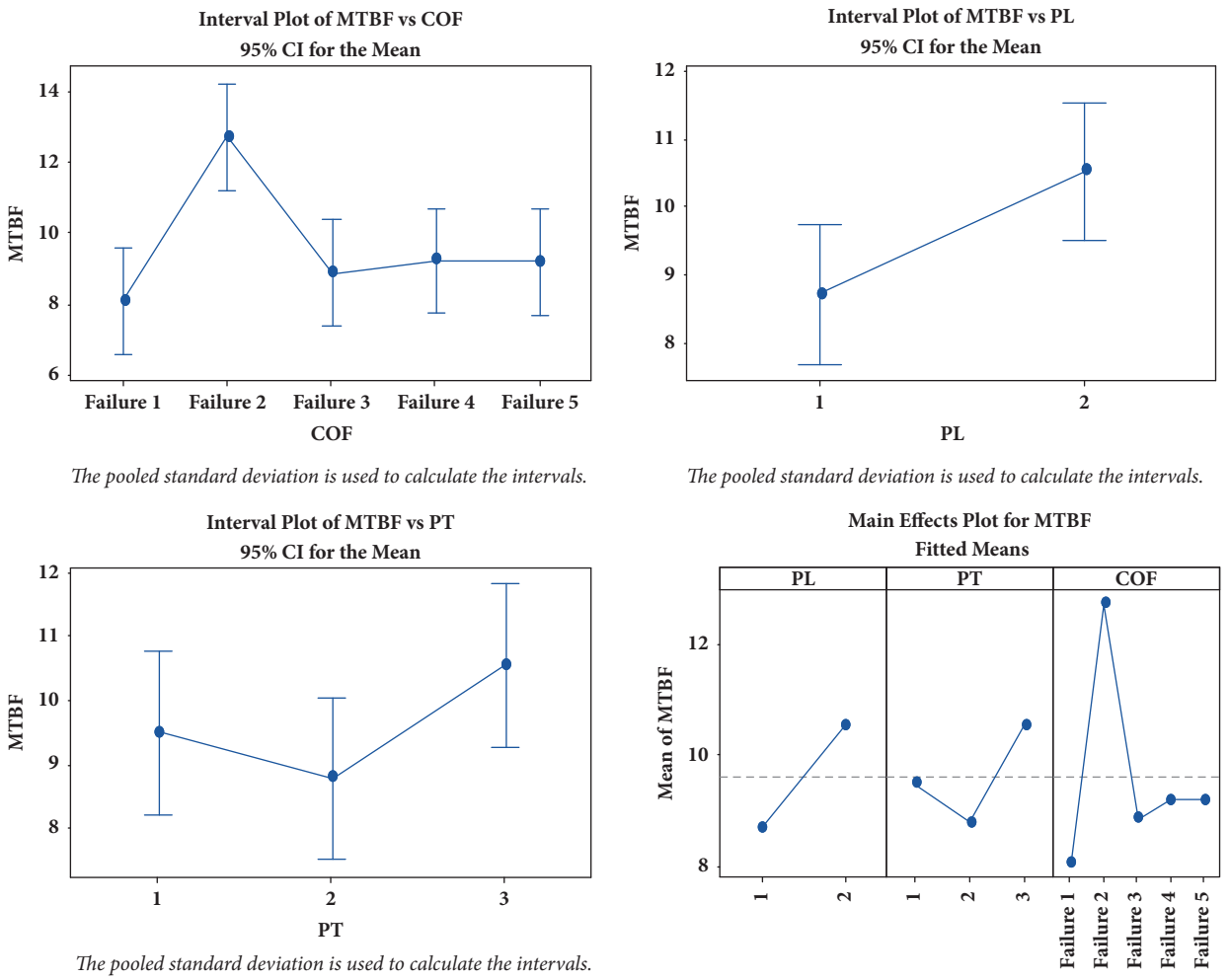


Figure 4: Main effects and interval plots.

material analysis report returns to the system with a raw material which is either not qualified to be used or does not comply with required raw material specification requirements needed by manufacturing process, this means material is unfit for use. The software will alert a signal for raw material shortage to avoid any abruptions in the production process.

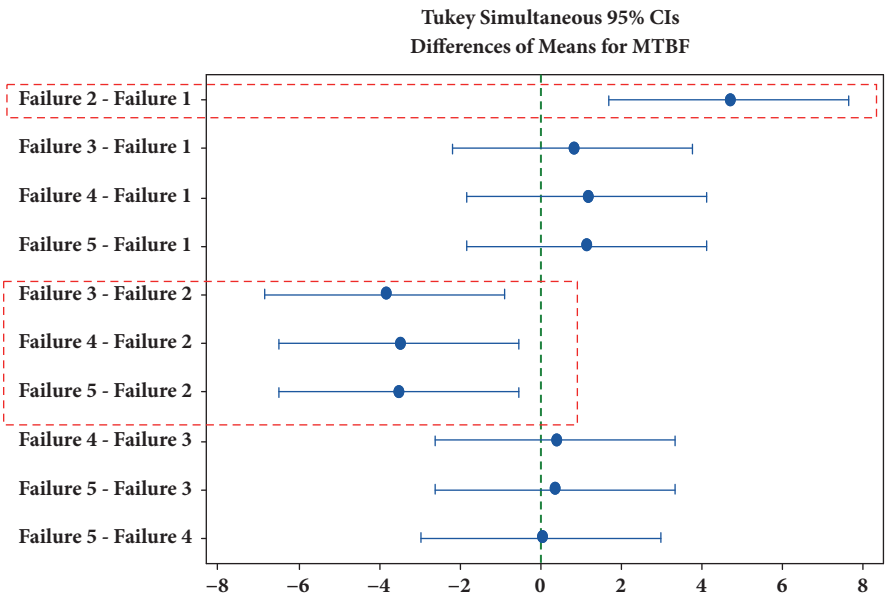
(ii) *Processed Raw Material*. The materials which have already met quality requirements and are being processed to eventually produce the new and finished products and ready to be delivered. At this stage, the smart system will detect any issues during material processing, for instance, over or under filled beverage bottles, packaging, etc. The smart system will collect all data of the faults and send it to the control unit.

(B) Robots. Robots are an integral part of any smart system. Robots are designed to carry on various activities in the factory. This section should consider developing ability that a robot can conduct required tasks can acquire the required level of skills and knowledge. In this layer, the control unit analyzes the professional ability and skills of the attending workers and then assigning them to the specific machine based on their skills and production demands. In this way, it can be guaranteed that a machine is being operated by an expert worker and it will help to avoid any planning failures.

(C) Equipment. These are the machines and devices that will be handled by the workers. Through detectors, such as sensors, cameras, and industrial wireless systems, data will be collected and analyzed to alert if there is any instability in the equipment.

Level 2: Control Unit Level. This level is comprised of four units:

(i) *Database Unit*. This unit collects and categorizes the data received from level 1. It is the main database for the system, acting as an interface between level 1, level 3, and the unit itself. All data is stored and sent by IoT devices and can be accessed by the users whenever the data is needed. This unit consists of servers and mirrors to support the data storage in case of any loss of data. It can be accessed through Wi-Fi anywhere and everywhere.



If an interval does not contain zero, the corresponding means are significantly different.

Figure 5: Tukey differences of means test for MTBF.

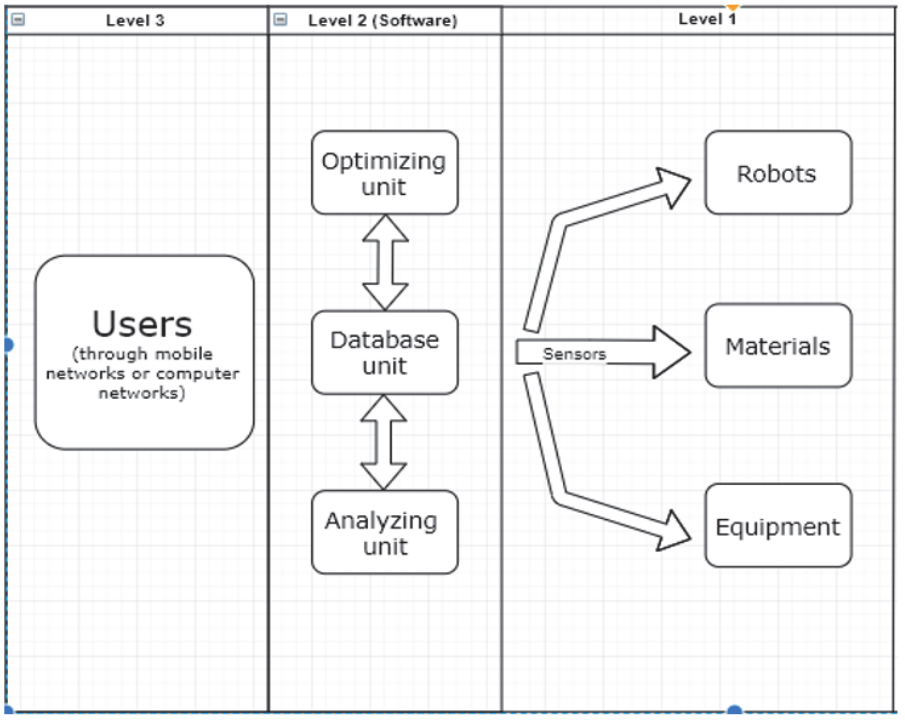


Figure 6: IoT based smart manufacturing system.

(ii) *Analyzing Unit*. This unit analyzes the data in the Database Unit. It reads the data and decides how it should be categorized so that the data can be organized at later stage with its respective category. This categorization facilitates to search the required category data whenever needed.

(iii) *Optimization Unit*. This unit is preprogrammed to carry out the decision-making process. Its job is to data sorting with its respective category.

(iv) *Software Unit*. Using software, this unit connects the database unit, analyzing unit and optimization unit so

they can work together to carry out all of the above-mentioned actions.

Level 3: User Level. This level will be used for accessing and planning by all users through mobile networks or computer networks. This layer is utilized to provide a personalized customization service and dynamic production process monitoring capabilities to users. In order to achieve effective service, this layer must be user-friendly and have reliable interaction methods.

8. Conclusion and Recommendations

In any manufacturing concern, machine failures can occur due to several reasons. This research study has investigated and identified the critical failures and their associated causes which adversely affect the manufacturing process. Real-time machine failure data (secondary data) from production reports and employee interviews (primary) data is collected. Statistical analysis using experimental design technique is performed and the results also demonstrate that the electrical failures (Failure 2) are the most occurring problem among all critical failure causes studied in this research. Production line failure is the one which has also significant effect on the machine failure. However, the product type failure does not contribute to the machine failure. The relationship between production line and production type indicates that there is no significant interaction effect. On the contrary, the relationship between production line and the cause of failures indicates a significant interaction effect, especially for the electrical failure with production line 2. This research study concludes that manufacturing concerns should focus on preventing critical failure and elimination of cause of failures and develop an active predictive failure control system rather than prevention system.

In future research, an effective and efficient failure control system is proposed which is based on smart manufacturing factory and will be comprised of smart sensor and communications technologies. Detailed description of the conceptual model is illustrated in this study for future development and implementation purposes.

Data Availability

The data used to support the findings of this study are available from the co-author Ammar Almoalem (ammar.imse@gmail.com) upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors of the paper gratefully acknowledge the technical and financial support of the Research Center of College of Engineering, Deanship of Scientific Research, King Saud University. Also, the authors would like to express their sincere gratitude to Engineer Ammar Moohialdin for providing industrial data to conduct this research study.

References

[1] J. Wan, S. Tang, D. Li et al., "A manufacturing big data solution for active preventive maintenance," *IEEE Transactions on Industrial Informatics*, 2017.

[2] D. Raman, S. V. Nagalingam, and G. C. I. Lin, "Towards measuring the effectiveness of a facilities layout," *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 1, pp. 191–203, 2009.

[3] C. Shekhar, M. Jain, J. Iqbal, and A. A. Raina, "Threshold control policy for maintainability of manufacturing system with unreliable workstations," *Arabian Journal for Science and Engineering*, vol. 42, no. 11, pp. 4833–4851, 2017.

[4] M. Fernández-Olmos and M. Ramírez-Alesón, "How internal and external factors influence the dynamics of SME technology collaboration networks over time," *Technovation*, vol. 64-65, pp. 16–27, 2017.

[5] K. L. Carper, "Failure information: Dissemination strategies," *Journal of Performance of Constructed Facilities*, vol. 1, no. 1, pp. 1–10, 1987.

[6] J. Moubray, "21st century maintenance organization part I: the asset management model," *Maintenance Technology*, vol. 16, no. 2, pp. 25–32, 2003.

[7] H. Groenevelt, L. Pintelon, and A. Seidmann, "Production lot sizing with machine breakdowns," *Management Science*, vol. 38, no. 1, pp. 104–123, 1992.

[8] L. Swanson, "Linking maintenance strategies to performance," *International Journal of Production Economics*, vol. 70, no. 3, pp. 237–244, 2001.

[9] I. P. S. Ahuja and J. S. Khamba, "Total productive maintenance: literature review and directions," *International Journal of Quality & Reliability Management*, vol. 25, no. 7, pp. 709–756, 2008.

[10] A. Pandey and A. K. Singh, "A Bayes test of homogeneity of several means for one parameter exponential populations," *Applied Mathematics and Computation*, vol. 108, no. 1, pp. 23–32, 2000.

[11] B. A. Peters and T. Yang, "Integrated facility layout and material handling system design in semiconductor fabrication facilities," *IEEE Transactions on Semiconductor Manufacturing*, vol. 10, no. 3, pp. 360–369, 1997.

[12] B. Moua and J. S. Russell, "Evolution of formal maintainability program for large manufacturing company," *Journal of Performance of Constructed Facilities*, vol. 15, no. 2, pp. 46–53, 2001.

[13] J. Bröchner, "Integrated development of facilities design and services," *Journal of Performance of Constructed Facilities*, vol. 17, no. 1, pp. 19–23, 2003.

[14] K. Jones and M. Sharp, "A new performance-based process model for built asset maintenance," *Facilities*, vol. 25, no. 13-14, pp. 525–535, 2007.

[15] A. S. Moohialdin and L. A. Hadidi, "Effect of failure type on downtime duration for a manufacturing facility," *Journal of Performance of Constructed Facilities*, vol. 31, no. 3, article 04016114, 2016.

[16] S. K. Roy and I. N. Sutapa, "CASE studies of use of design of experiments in material research," *Jurnal Teknik Industri*, vol. 5, no. 1, pp. 32–40, 2004.

[17] S. N. Adhikary, "A case study in increasing productivity by DOE in manufacturing sector," *International journal of Advancement in Engineering Technology, Management and Applide Science*, vol. 1, no. 7, 2014.

[18] H. Shen and H. Wan, "Controlled sequential factorial design for simulation factor screening," *European Journal of Operational Research*, vol. 198, no. 2, pp. 511–519, 2009.

[19] F. Chan and H. Chan, "Simulation analysis of a PCB factory using factorial design–a case study," *The International Journal of Advanced Manufacturing Technology*, vol. 21, no. 7, pp. 523–533, 2003.

[20] L. T. Ek, "Quality improvement using factorial design," Paper presented at the 9th Int Convention on Quality Improvement, 2005.

[21] A. B. Medeiros, A. Pandey, R. J. Freitas, P. Christen, and C. R. Soccol, "Optimization of the production of aroma compounds by Kluyveromycesmarxianus in solid-state fermentation using factorial design and response surface methodology," *Biochemical Engineering Journal*, vol. 6, no. 1, pp. 33–39, 2000.

[22] N. S. Zulu and A. T. Modi, "A factorial experiment to study the effect of seeding rate and nitrogen side-dressing on yields of two dry bean (Phaseolus vulgaris L.) cultivars," 2011.

[23] S. Haller, S. Karnouskos, and C. Schroth, "The internet of things in an enterprise context," in *Proceedings of the Future Internet Symposium*, 2008.

[24] P. Jiang, K. Ding, and J. Leng, "Towards a cyber-physical-social-connected and service-oriented manufacturing paradigm: Social Manufacturing," *Manufacturing Letters*, vol. 7, pp. 15–21, 2016.

[25] J. Wang and J. Zhang, "Big data analytics for forecasting cycle time in semiconductor wafer fabrication system," *International Journal of Production Research*, vol. 54, no. 23, pp. 7231–7244, 2016.

[26] L. Guo and J. Qiu, "Optimization technology in cloud manufacturing," *The International Journal of Advanced Manufacturing Technology*, vol. 97, no. 1-4, pp. 1181–1193, 2018.

[27] R. Y. Zhong, X. Xu, E. Klotz, and S. T. Newman, "Intelligent Manufacturing in the context of industry 4.0: a review," *Engineering Journal*, vol. 3, no. 5, pp. 616–630, 2017.

[28] J. Lienig and H. Bruemmer, *Reliability Analysis, Fundamentals of Electronic Systems Design*, Springer International Publishing, 2017.

[29] Z. Shu, J. Wan, D. Zhang, and D. Li, "Cloud-integrated cyber-physical systems for complex industrial applications," *Mobile Networks and Applications*, vol. 21, no. 5, pp. 865–878, 2016.

[30] J. Wan, M. Yi, D. Li, C. Zhang, S. Wang, and K. Zhou, "Mobile services for customization manufacturing systems: an example of industry 4.0," *IEEE Access*, vol. 4, pp. 8977–8986, 2016.

[31] J. Wan, S. Tang, Z. Shu et al., "Software-Defined Industrial Internet of Things in the Context of Industry 4.0," *IEEE Sensors Journal*, vol. 16, no. 20, pp. 7373–7380, 2016.

[32] F. Shrouf, J. Ordieres, and G. Miragliotta, "Smart factories in Industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm," in *Proceedings of the 2014 IEEE International Conference on Industrial Engineering and Engineering Management, IEEM 2014*, pp. 697–701, Malaysia, December 2014.

[33] J. Lee, "Smart Factory Systems," *Informatik-Spektrum*, vol. 38, no. 3, pp. 230–235, 2015.

[34] J. Lee, H.-A. Kao, and S. Yang, "Service innovation and smart analytics for industry 4.0 and big data environment," *Procedia Cirp*, vol. 16, pp. 3–8, 2014.

[35] V. Gunes, S. Peter, T. Givargis, and F. Vahid, "A survey on concepts, applications, and challenges in cyber-physical systems," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 12, pp. 4242–4268, 2014.

[36] K. Braley, "Manufacturing Growth Continues but Economy Still a Challenge, According to ASQ Outlook Survey," 2013, http://www.prweb.com/releases/2013/12/prweb11430148.htm.

[37] *Building Smarter Manufacturing With The Internet of Things (IoT)*, Lopez Research LLC, 2014.

[38] L. Wang, "Cyber manufacturing: research and applications," in *Proceedings of the Tenth International Symposium on Tools and Methods of Competitive Engineering*, 2014.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 2946575, 12 pages
https://doi.org/10.1155/2018/2946575

Research Article

Software Architecture Solution Based on SDN for an Industrial IoT Scenario

José L. Romero-Gázquez [1] **and M. Victoria Bueno-Delgado** [1,2]

[1] *Universidad Politécnica de Cartagena, Grupo en Ingeniería en Redes de Telecomunicación, Plaza del Hospital 1, Edif. Antigones, 30202, Cartagena, Spain*
[2] *E-lighthouse Network Solutions S.L., 30203, Cartagena, Spain*

Correspondence should be addressed to José L. Romero-Gázquez; josel.romero@upct.es

Received 12 April 2018; Revised 20 July 2018; Accepted 5 September 2018; Published 27 September 2018

Guest Editor: Muhammad Imran

The Industry 4.0 (I4.0) adoption comprises the change of traditional factories into *smart* using the ICTs. The goal is to monitor processes, objects, machinery, and workers in order to have real-time knowledge about what is going on in the factory and for achieving an efficient data collection, management, and decision-making that help improve the businesses in terms of product quality, productivity, and efficiency. Internet of Things (IoT) will have an important role in the I4.0 adoption because future smart factories are expected to rely on IoT infrastructures composed of constellations of hundreds or thousands of sensor devices spread all over the industrial facilities. However, some problems could arise in the massive IoT deployment in a medium-high factory: thousands of IoT devices to cope from different technologies and vendors could mean dozens of vendor tools and user interfaces to manage them. Moreover, the heterogeneity of IoT devices could entail different communication protocols, languages, and data formats, which can result in lack of interoperability. On the other hand, conventional IT networks and industrial machinery are expected to be managed together with the IoT infrastructure, maybe using a tool or a set of tools, for *orchestrating* the whole smart factory. This work meets these challenges presenting an open-source software architecture solution based on OpenDaylight (ODL), a Software Defined Network (SDN) controller, for orchestrating an industrial IoT scenario. This work is addressed by shedding light on critical aspects from the SDN controller architectural choices, to specific IoT interfaces and the difficulties for covering the wide range of communication protocols, popular in industrial contexts. Such a global view of the process gives light to practical difficulties appearing in introducing SDN in industrial contexts, providing an open-source architecture solution that guarantees devices and networks interoperability and scalability, breaking the vendor lock-in barriers and providing a vendor-agnostic solution for orchestrating all actor of an I4.0 smart factory.

1. Introduction

The European Union (EU) is moving towards what is already considered the fourth industrial revolution, the Industry 4.0 (I4.0). The aim of I4.0 is to transform the conventional factories into smart factories using the Information and Communication Technologies (ICTs). These add flexibility in manufacturing, mass customization, improve the quality of products and processes, and increase the productivity. A subset of ICTs, the Key Enabling Technologies (KET), have been proposed as drivers to carry on this change: the use of sensors, actuators, wireless communications, next generation networks, robots, additive manufacturing, cloud computing, big data and cybersecurity, among others [1]. In this new technological scenario many strategic points (products, processes, workplaces, workers, etc.) will be monitored by using different technologies and devices. These will be wired or wireless connected to the factory local network or to the Internet. One or more software tools will be in charge of collecting, storing, filtering, and managing the enterprise data. This scenario will be carried out by setting an Internet of Things (IoT) infrastructure [2, 3]. Hundreds or thousands IoT devices will be spread in the factory collecting data at real-time about industrial processes, machinery, and workers performance. The goal is to achieve an efficient factory management and decision-making to improve the

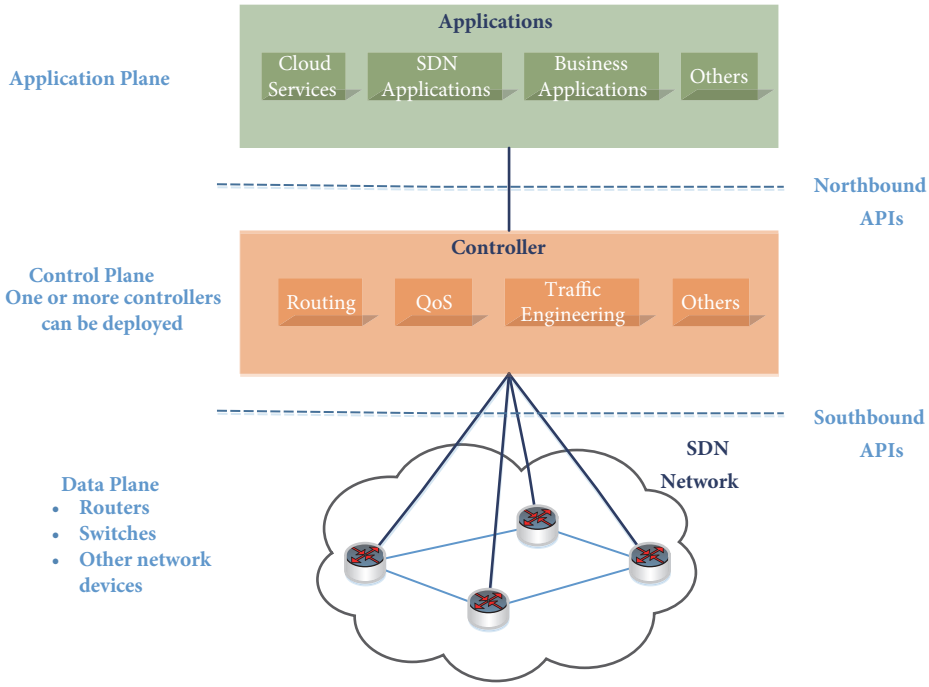


Figure 1: Three layered-based Software Defined Network architecture.

business in terms of product quality, productivity, and efficiency.

The I4.0 adoption in the above scenario entails different problems for a medium-high factory:

(i) Thousands of IoT devices to cope, from different vendors and with different technologies, could mean dozens of vendor tools and user interfaces to manage them.

(ii) The heterogeneity of IoT devices could entail different communication protocols, languages and data formats, among others, which can result in lack of interoperability.

(iii) Conventional IT networks and industrial machinery will coexist with the new IoT infrastructures, and all will be "connected" to achieve the full I4.0 adoption.

(iv) The data traffic in the industrial network infrastructure deployed could significantly increase, deriving in undesirable packet delays and network congestion. The data traffic should be routed efficiently to avoid it.

The two former could be managed by using a local or cloud-based IoT platform [4, 5] but the two latter would remain partially or totally unsolved. Some works in the recent scientific literature have proposed some software solutions to deal with these problems (see Section 2). However, they do not deal with heterogeneous scenarios where noncompliant IoT devices or industrial machinery are operating. This work tackles these challenges, by proposing a single open-source software architecture solution based on a layered-based Software Defined Network (SDN) architecture (see Figure 1).

SDN was firstly designed to orchestrate IT networks, but nowadays some SDN controllers include plugins to connect in the southbound with IoT devices and networks. The proposed software uses OpenDaylight (ODL (https://www.opendaylight.org/)), a SDN controller with a dedicated IoT plugin called IoTDM that manages and stores data generated by IoT based devices according to oneM2M (http://www.onem2m.org) standard. This is required to provide a standardized interface to manage and interact with user-applications. Since IoTDM by default only understands native JSON/XML data formats, this work breaks the lack of interoperability with noncompliant industrial devices by providing a software architecture solution that guarantees interoperability among these devices/technologies and the IoTDM plugin in the southbound. Finally, ODL has to transfer the data from the southbound to the northbound and vice versa. Here another software entity is needed because in the southbound IoTDM manages data with a native JSON/XML format, and applications in the northbound could work with a generic JSON/XML format. This work proposes the use of a system of plugins based on oneM2M standard. This ensures agile management, collection, and presentation to the final user applications.

The software architecture proposed in this work gives light to the practical difficulties of adopting I4.0 to medium-high factories, proposing a modular architecture solution based on pre-existing open-source software. Its modularity allows anyone to develop new plugins for those technologies involved in any industrial scenario not considered in this work. The solution is also vendor-agnostic, breaking the vendor lock-in barriers, and it is scalable in terms of amount of devices to manage and technologies to support, because ODL permits to create a cluster of controllers running as

a single entity to allow fast scaling, high availability, data persistence, and fault tolerance.

Finally, the solution also exploits some of the SDN benefits:

(i) The management of tasks is automated and isolated from the complexity of the physical infrastructure through easy-to-use interfaces.

(ii) New services and applications could be provided in short time; in addition the IT administrators have the possibility of programming network functionalities and services as required, eliminating dependence on hardware manufacturers.

(iii) The network routing policies can be configured and managed for the whole network, using the same software solution, monitoring the network reliability, safety, and efficiency.

(iv) The user experience improves because the applications can exploit the centralized information about the state of the network and data collected, making it possible to react in real time and to carry out modifications that can improve the network performance.

(v) The operating costs of the industrial network management could be significantly reduced.

An Industrial IoT (IIoT) pilot scenario has been deployed to test the software architecture proposed. An application in the northbound has been implemented for testing some management tasks: add, remove, configure, and manage different IIoT devices, commonly used in industrial context. The scenario has been deployed with a single instance of ODL controller, giving the possibility of setting up a second ODL controller as backup to address risk situations such as high network latency, bottleneck, cybersecurity attacks, or faults.

Note that this work gives light to the practical difficulties appearing in introducing SDN in industrial contexts. It provides light to critical aspects from the controller architectural choices, to specific IoT interfaces and the difficulties for covering different communication protocols popular in industrial contexts. To do this, the paper is organized as follows: Section 2 reviews the related work. Section 3 summarizes the current IoT platforms and explains the reasons to use ODL and IoTDM in industrial scenarios. Section 4 reviews the most extended technologies and communication protocols in industrial facilities, remarking those that need a parser to be compatible with ODL and IoTDM. Section 5 presents the software architecture solution and Section 6 describes an IIoT pilot scenario testing the proposal. Finally Section 7 concludes.

2. Related Work

As we stated, it seems clear that the IIoT adoption is an unstoppable fact within the I4.0 framework. In a medium-high factory, it entails different problems that could be solved by using software solutions based on SDN, permitting the orchestration of IT, IoT, and IIoT networks. It seems right to focus the effort on the development of software services that allow (i) unifying the management of those heterogeneous networks given in an industrial scenario with a single and common modular software entity and (ii) providing interoperability in terms of technologies and devices popular in industrial contexts. Finally, (iii) the use of open standards should be mandatory, permitting to break the vendor lock-in problems and to create vendor-agnostic solutions.

Although the management of IoT and IIoT has received a lot of attention from the research community, to the best of the authors' knowledge there are only a few proposals where SDN is used to orchestrate these networks in an I4.0 scenario, but they do not provide all the features enumerated above. Some works solve the orchestration of IIoT networks without SDN, like in [6], where an IoT network is orchestrated by a network of *fog nodes*, which manage all layers of an IoT ecosystem to integrate different service modules into a complex topology. In [7] two architectures for an IoT networks orchestration are presented, using OSGi (https://www.osgi.org) and REST (https://restfulapi.net/), paying attention only to higher layers, not to physical connection. Moreover, universal standards for IoT like oneM2M are not taken into account, missing interoperability. Both proposals lack any single tool (industrial application and/or SDN orchestrator) to have a complete network knowledge and management. The same scenario is set in [8] where the main concepts involved in an IoT orchestrated network are explained, but without the use of SDN.

Other works present solutions based on SDN, but they do not look into to understand if they fulfil the requirements and solve the problem that I4.0 adoption entails in a medium-high factory. Some examples are in [9–17]. In [9] the authors analyse the IoT challenges that the network and IT infrastructures face, showing the Network Functions Virtualization (NFV) and SDN benefits from a network operator point of view. Work in [10] is focused on the fifth generation (5G) of mobile networks technology (5G) [11], applying SDN, NFV and IoT technologies. Works addressed in [12, 13] show an IoT architecture based on SDN, and [14] is a use case of [13] focused on 5G. As in the previous works, they do not provide a single solution where the IoT operations are managed on the SDN platform. In [15] the theory behind the use of SDN to manage an industrial network is explained, as well as in [16, 17], where the use of SDN for wireless networks (also called SDWN) and IoT is study in depth, highlighting some of the future research directions and open research issues based on the limitations of the existing SDN-based technologies.

At the time of writing this work, only a few works [18–20] provide a vendor agnostic IoT and SDN software architecture solution. However, they do not deal with heterogeneous scenarios where noncompliant IoT devices or industrial machinery are operating, and their solutions do not take into account oneM2M standard. In [18] a novel framework of IIoT with SDN and Edge Computation (EC) is proposed to address an optimal adaptive transmission routing mechanism. Open Mul (http://www.openmul.org/) is used as SDN controller. The authors do not look into how the architecture is designed, because the goal of the work is focused on the management of the network at upper layers. In [19] a SDN solution for

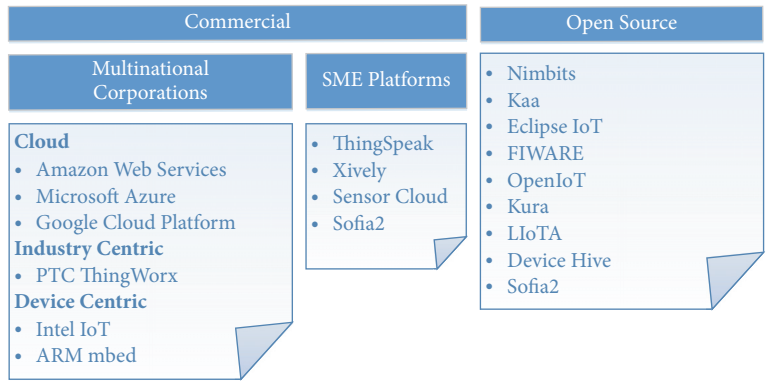


Figure 2: Leading IoT platforms.

IoT built with Open Network Operating System (ONOS (https://onosproject.org/)) as SDN orchestrator is proposed. The architecture is suitable for orchestrating end-to-end service chains deployed across heterogeneous SDN/NFV domains and define a related high-level and vendor-agnostic intent-based northbound interface (NBI). They propose a network management architecture based on the use of a dedicated controller for each type of network. In [20], the authors analyse the deployment of SDN in the Industry 4.0 paradigm, pointing out the three main problems and future issues to address for a fast and success implementation in industrial scenarios: data safety and system reliability, technology standardization and practical implementation. The authors suggest a solution to the above problems the development of a hybrid node device capable of handling with IT and IoT networks.

As a conclusion of this review, there are some works focused on explaining and developing SDN scenarios for conventional IT or IoT networks, but most of them are missing the interoperability of popular protocols and technologies of industrial contexts, the modularity, and the open-source benefits. This work is a contribution in this matter, providing light to critical aspects of SDN controller choice, IoT interfaces, and widely extended technologies and protocols of industrial contexts.

3. ODL as Industrial IoT Platform

There are many IoT platforms [4, 5] aiming at managing IoT networks, but it is difficult to find one that complies with the requirements of open source and heterogeneity for an Industrial IoT (IIoT) scenario. Figure 2 summarizes the main features of the most popular IoT platforms.

As the authors' knowledge most of open source IoT platforms enables interoperability, some of them through a multilanguage and multiprotocol system. They usually provide an SDK to develop applications. However, a real implementation in an industrial scenario with a wide range of industrial technologies requires big human resources to develop the functionalities required. In this point is where some companies offer their expensive commercial versions. This picture invited us to explore other software alternatives, like SDN, because nowadays some SDN platforms allow the integration of IoT services together with traditional human-based services, through a global orchestration of distributed cloud, heterogeneous IT, and IoT networks. In SDN (see Figure 1), the software entity *controller* (i) carries the huge amount of data generated by the IT and IoT networks to any distributed computing *node*, (ii) allocates computing and storage resources into distributed data centers, and (iii) processes the collected data to make decisions. In addition, SDN offers the possibility of network orchestration through the use of a cluster of controllers, which adds scalability to the management system, high availability, data persistence, and fault tolerance using a backup controller.

Among all available SDN controllers, this work has addressed a comparison from those that are open source and widely accepted by the scientific community [21, 22] to point out the one that meets the requirements of an IIoT scenario.

ONOS is an open source SDN controller written in Java with a distributed network operating system and good performance. It is composed by an extensible and modular platform and a set of applications. As southbound interfaces (SBI), ONOS supports OpenFlow, NETCONF, and PCEP. As northbound interfaces, it uses REST. Ryu (http://osrg.github.io/ryu/) is a component-based SDN controller with a set of predefined components written in Python, which can be modified, extended and composed to create a customized controller application. It supports different OpenFlow versions. Since it is written in Python, the network performance seems to be lower than other Java based SDN controllers, as stated in [21]. Floodlight (http://www.projectfloodlight.org/floodlight/) consists of a set of modules which provide service to other modules and/or to the control logic application through a simple Java or REST API. It can run on the top of Linux, Mac, and Windows OS. According to [23], Floodlight shows high performance in terms of packet loss. OpenDaylight is an open-source controller written in Java, with a high performance and widely supported by the networking industry. ODL includes IoT Data Broker (IoTDM) plugin, which opens the door to use ODL as a SDN platform for IoT networks management. IoTDM is a data-centric middleware that acts as an oneM2M compliant and enables authorized applications to retrieve IoT data uploaded by any device in the network. With IoTDM, ODL seems to be the most suitable candidate to manage an

IIoT. Table 1 summarizes a comparison of the main features provided by the SDN controllers reviewed.

4. Technologies and Communication Protocols in an Industrial IoT Scenario

In the previous section it was concluded that ODL together with IoTDM is our most suitable solution as IIoT platform. However, not all technologies and communication protocols coexisting in an IIoT scenario are compatible with ODL and IoTDM. Note that ODL and IoTDM works with XML/JSON data formats and HTTP, MQTT, or CoAP as communication protocols

In this section we review what technologies and communication protocols are widely used in an industrial facility, showing their data formats and compatibility with ODL and IoTDM. Table 2 summarized them. The goal is to identify the need of a *parser* in the southbound to enable communication between industrial technologies and ODL together with IoTDM. The technologies reviewed are the follows:

(i) Radio Frequency Identification (RFID) is one of the wireless technologies widely used in the industry as an alternative to the bar code. RFID uses the EPCglobal Class-1 Gen-2 standard as communication (and anticollision) protocol [24]. The data received and stored in the RFID reader from RFID tags can adopt XML format and can be sent via HTTP.

(ii) Bluetooth Low Energy (BLE (https://www.bluetooth.com)), the power-conserving variant of Bluetooth for personal area networks (PAN), is commonly used by Internet-connected machines and appliances. Its name also refers to its protocol, which is a full protocol stack. The BLE data format is not JSON or XML structure.

(iii) Modbus (http://www.modbus.org) is an industrial standard communication protocol, used for programmable logic controllers (PLCs), to transmit signals from instrumentation and control devices back to a main controller or data gathering system, and it is typically used for connecting a supervisory computer with a remote terminal unit (RTU) in supervisory control and data acquisition (SCADA) systems. There are different versions of Modbus protocol: Modbus RTU and Modbus ASCII for serial lines and Modbus TCP for Ethernet. The data format of Modbus is an Application Data Unit (ADU).

(iv) Controller Area Network (CAN or CAN Bus) [25] is a vehicle bus standard designed to allow electronic control units and devices to communicate with each other in applications without a host computer. Examples of devices working with CAN Bus are electronic control units, microcontrollers, devices, sensors, actuators, and other electronic components. CAN Bus is also a message based protocol, originally designed for multiplex electrical wiring within motor vehicles, but is used in the context of industrial applications. This standard generates DBC files, which are a proprietary format that describes the data over a CAN bus.

(v) Ethernet for Control Automation Technology (EtherCAT (https://www.ethercat.org)) is based on the CAN open protocol and on Ethernet but differs from network communications in being specifically optimized for industrial automation control, and its data format is EherCAT Slave Information (ESI).

(vi) Profibus and Profinet (https://us.profinet.com) are standards for fieldbus communications in the industrial automation. Profibus links controllers or control systems to decentralized field devices (sensors and actuators) on the field level and also enables consistent data exchange with higher ranking communication systems. Profinet connects devices, systems, and cells, facilitating faster, safer, less costly, and higher quality manufacturing. It easily integrates existing systems and equipment while bringing the richness of Ethernet down to the factory floor. Profinet communications are commonly TCP/IP based. Both Profinet and Profibus devices can connect to SCADA systems and IoT platforms through a gateway based on OPC (https://opcfoundation.org/), a communication standard for industrial telecommunication processes. OPC Unified Architecture (OPC UA) is the client-server architecture (see Figure 3) that serves as gateway to convert Profinet and Profibus data to proprietary XML/JSON data format. OPC UA supports real-time data communications, alarms, security features, etc. The OPC server is the data source and any application based on OPC standard can access to this server as OPC client to read and write on it. Software vendors only need to include OPC client capabilities in their products. Unfortunately, the industrial OPC software is not traditionally open source, and the companies are obeyed to spend a lot of money for locked-vendor OPC client-server solutions. Recently, some open-source OPC UA software solutions are available in [26].

5. Software Architecture Based on SDN and IoT for Industrial Iot Scenarios

Figure 1 is extended to Figure 4 to illustrate the software architecture proposed for an industrial I4.0 scenario if only open-source software is used. All devices (networks) are placed in the bottom, *orchestrated* by the SDN controller. In the bottom-left the IIoT network is placed, with IoT devices from different technologies (sensor motes, RFID, BLE) and industrial machinery working under Profinet, Ethercat, CAN Bus and Modbus. The IIoT network coexist with the conventional IT enterprise network, composed of routers, switches, PCs, printers, etc.

In the middle of Figure 4 a single controller is plotted, managing the IT and IIoT networks. A second controller is set as backup, to guarantee scalability and react to risk situations such as high computational load in the main controller, cybersecurity attacks, bottleneck in the IT or IIoT network,

Table 1: Comparison of SDN controllers.

SDN Controller	Language	Southbound	Northbound	Own IoT component	Modularity	Performance∗	Cluster capability
ONOS (https://wiki.onosproject.org/display/ONOS/System+Components)	Java	OpenFlow, OVSDB, NETCONF, ...	REST, RESTCONF, ...	Missing	High	High	Yes
Ryu (http://osrg.github.io/ryu/)	Python	OpenFlow, OVSDB library, NETCONF, ...	REST	Missing	Medium	Low	No
Floodlight (https://floodlight.atlassian.net/wiki/spaces/floodlight-controller/pages/1343549/Architecture)	Java	OpenFlow	REST	Missing	High	High	No
ODL (https://www.opendaylight.org/what-we-do/current-release/carbon)	Java	OpenFlow, PCEP NETCONF, OVSDB, BGP, ...	REST, RESTCONF, NETCONF, AMQP.	IoTDM	High	High	Yes

∗ Big amount of devices to orchestrate.

Table 2: Comparison of the most common technologies and communication protocols in an IIoT scenario.

Technology	Communication Protocol to	XML or JSON format	Parser in the southbound for SDN and IoTDM communication
IoT devices	HTTP, CoAP, or MQTT	Yes, XML, JSON	No
RFID	EPCglobal Class-1 Gen-2 (among RFID tags and reader) HTTP (between RFID reader and server)	Yes, XML	No
BLE	BLE	No, if OPC-UA is not used	Yes, if OPC-UA is not used
Modbus	Modbus RTU, Modbus ASCII, Modbus TCP	No∗	Yes∗
CAN Bus	CAN Bus	No∗	Yes∗
EtherCAT	CAN open-Ethernet	No∗	Yes∗
Profibus	CAN open-Ethernet	No∗	Yes∗
Profinet	TCP/IP	No∗	Yes∗

∗If OPC-UA is not used.

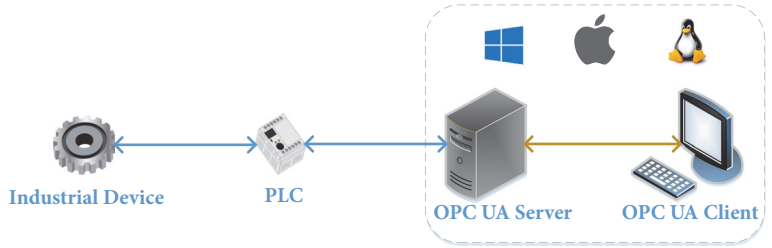


Figure 3: General OPC UA client/server communication.

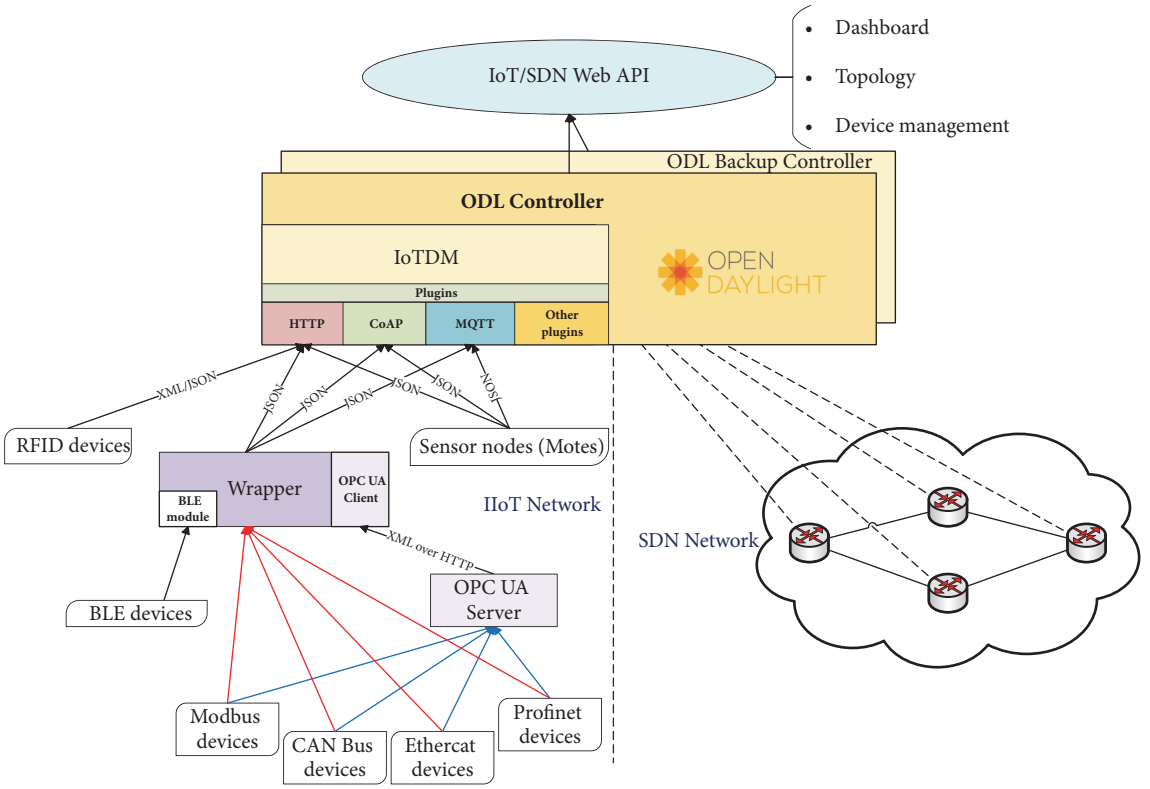


Figure 4: Industrial IoT software architecture solution.

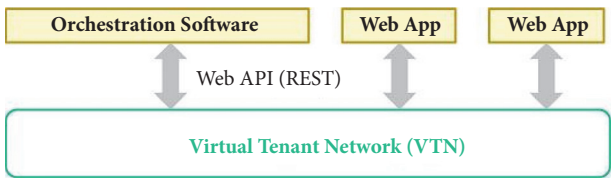


Figure 5: VTN orchestration Software.

etc. The decision comes from the fact that according to the work in [24], a single controller has more than enough powerful to manage the traffic of a set of medium-size IT and IoT networks in an industrial scenario. Authors in [27] show how a single NOX (https://github.com/noxrepo/nox-classic/wiki) controller can handle at least 30Kb flows per second with 10 ms of install delay. Hence, in this scenario, only two controllers are considered, but the number of controllers could vary because ODL allows a flexible deployment of controllers, depending on the needs. ODL uses a software functionality called Virtual Tenant Network (VTN) for the orchestration of a cluster of controllers (see Figure 5).

Another possibility of SDN controller deployment, not considered in this work, is the use of a dedicated SDN controller for each type of network: one controller for the IT network and one controller for the IoT network. This solution has some advantages such as modularity, more efficient management of applications in the northbound and southbound, or a decrease of bottleneck risk. However, this deployment shows some disadvantages: (i) more hardware, that is, at least two high-powerful computers to allocate each controller and two more for allocating backup controllers; (ii) to use a nonunified interface: the number of user interfaces will be equal to the number of controllers running in the deployment; (iii) to replicate NBI and SBI applications if they operate in both networks: the use of a single application for managing more than one network is very usual. For instance, NBI applications for visualization and device management with dashboard are needed in every network orchestration.

The deployment and the communication among all actors in the architecture proposed are explained in detail in the next subsections, from the physical to the application layer, that is, from the southbound to the northbound SDN architecture.

5.1. Southbound Layer. The southbound is the physical/logical connection between devices (networks) placed in the industrial scenario and the SDN controller (ODL and IoTDM), installed in a virtual or local machine. Three networks are identified: IT network, IoT network and noncompliant IoT networks, or industrial machinery networks. Conventional IT networks are directly connected to the ODL controller via wired or wireless connection, transmitting and receiving monitoring and management data with the widely used Openflow and NETCONF protocols and even with SNMP protocol if it is required, because ODL has a plugin to support it. The IoT network, composed of Wireless Sensor Networks (WSN) or a set of sensor nodes, is connected to ODL through IoTDM via wired or wireless connection (LoRa (https://www.lora-alliance.org/) and Sigfox (https://www.sigfox.com/) coud be used as wireless network). Other networks noncompliant with the communication protocols HTTP, CoAP, or MQTT and the data formats JSON or XML will need a gateway or *wrapper* to communicate with ODL through IoTDM.

RFID is compatible with IoTDM. RFID readers are the gateways of the RFID network deployments, sending and receiving the data to ODL and IoTDM in JSON or XML format, through HTTP. Most of IoT sensors (motes) in the market are also compatible with IoTDM. Motes acting as gateways in the sensor network deployments store and send data collected directly to ODL through IoTDM in JSON format using CoAP or MQTT protocol. TelosB, TinyOS, Beagle Bone, or Tmote Sky use CoAP while Raspberry Pi and Arduino can use both.

On the other hand, those devices operating with BLE need a *wrapper* to communicate with IoTDM. The wrapper is a SW+HW entity that can be implemented in the same machine where ODL together with IoTDM is being executed or in a low cost solution near the BLE devices, for instance, a Raspberry Pi 3 Compute Module IO Board V3 with one or more Compute Modules 3 (https://www.raspberrypi.org/products/compute-module-io-board-v3/). Raspberry Pi 3 is widely used in industrial scenarios due to their robustness. In any of the two options, a BLE receptor is required (via USB or GPIO) as gateway between the BLE network and IoTDM, and a parser from BLE data to JSON and vice versa must be working in the wrapper. An open-source solution of this parser can be found in [28].

The industrial machinery networks, with devices operating with technologies and communication protocols like Modbus, CAN Bus, and Ethercat cannot operate directly with IoTDM due to their data format and communication protocols incompatibility. As we stated in the previous section, this machinery uses as intermediate step in their communications a OPC UA client-server architecture. Each technology needs an OPC UA server to convert the data to JSON format and to communicate with an OPC UA client. The OPC UA client is, at the end, the entity that must communicate with ODL and IoTDM. Note that OPC-UA server-clients are not typically open-source software. Then, two open source options are proposed in this work for enabling interoperability among industrial machinery and ODL with IoTDM:

(i) An OPC UA client-server solution for each technology, installed in an intermediate machine. This machine could be the wrapper. This must parse, adapt, and restructure the received data from OPC UA clients in order to shape a valid JSON or XML understood by IoTDM. The communication procedure of this option is plotted in Figure 4 with blue lines. In [26] a list of open source solutions can be found.

(ii) A parser for each technology which addresses separately the data conversion to XML or JSON. They could be installed in an intermediate machine (wrapper). The communication procedure of this option is plotted in Figure 4 with red lines. There are some open-source solutions that can be used: modbus TCP [29] for Modbus; ecatmod for EtherCAT [30] and libcanardbc [31] for CAN bus.

Table 3: Methods implemented in the HubPlugin class.

Method	Description
getConfig	Returns to the controller the hub configuration
getDevices	Returns to the controller the device list managed by the hub.
getDeviceConfig	Returns to the controller a concrete device configuration
getDeviceData	Returns to the controller the data given by a concrete device
getDeviceSchema	Returns to the controller the configuration scheme from a concrete device
setConfig	Updates the hub configuration
setDeviceConfig	Updates a concrete device configuration

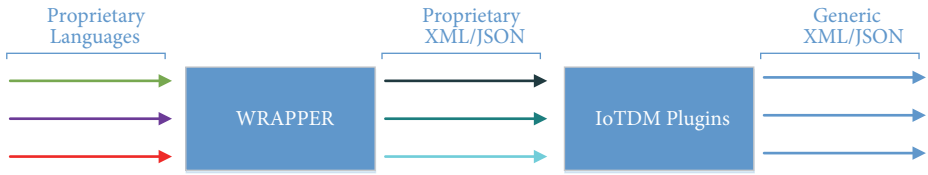


Figure 6: Data flow (upload) to reach a generic JSON/XML data format.

In any of the two options an intermediate machine is needed. In Figure 4 the wrapper is plotted as a single machine, but a set of machines could be required if the number of networks to gateway or the amount of data to process is high.

Note that the software solution proposed for industrial machinery will change in a near future because the OPC Foundation is working on a new extension to make OPC UA compatible with IoT applications, the OPC UA PubSub [32].

In short, those devices that cannot communicate directly with IoTDM make use of a wrapper to shape their data to be understood by IoTDM. The communication among the actors of this scenario is illustrated in Figure 6. The wrapper joins the functionalities of the parsers described above as a single bridge that receives the data from different technologies and parses and sends the converted data to ODL with IoTDM.

5.2. Controller Layer. ODL with IoTDM works in the controller layer. It can be run in a dedicated enterprise server or in the cloud. The latter could simplify the implementation and reduce the hardware costs. The decision about where the SDN platform must be allocated depends on the computation resources in the IT network factory, the size of the IT IoT and industrial machinery networks and the amount of data traffic to manage, the security policies in the enterprise, etc. ODL and IoTDM can be executed in Linux and Windows, but the former is better in terms of performance and compatibility. When ODLwith IoTDM is executed, it keeps awaiting incoming device petitions and/or data. The data received by IoTDM with specific JSON structure must be parsed to generic JSON structure, needed to be understood by any application in the northbound layer. Thanks to the open ODL functionality, this parser has been developed in this work. Here is where the vendor lock-in problem is completely broken.

The parser has been developed with an abstract class, *HubPlugin*, which has been defined in the controller. It includes some methods to be implemented for each new technology to adapt. This lets us isolate the communication mechanisms and the data formats of each vendor. The plugin not only provides the data in the required JSON format but also sets the communication protocol with the physical *hub* (HTTP, CoAP, or MQTT). In this software architecture a *hub* is understood as a technology involved in the physical layer (RFID, BLE, Profibus, Modbus, etc.). Each *HubPlugin* implementation must include a set of methods, summarized in Table 3.

The data in ODL with IoTDM are internally processed and stored in the Model-Driven SAL (MD-SAL) [33]. This is a set of infrastructure services defined in ODL with the aim of providing a common and generic support to applications, plugins for developers and infrastructure services for data storage, RPC, service routing and notification for publish/subscribe services. Figure 7 shows the module hierarchy of the parser implemented in the controller layer. The services are described as follows:

(i) Controller Manager is responsible for the northbound interface implementation. It also includes the controller core, adding the following tasks: (i) communication with ODL data base (MDS-SAL) and (ii) dynamic plugins instantiation (specific format to generic format translators) for enabled hubs (technologies).

(ii) Controller Provider is in charge of giving the base element executed by ODL when it starts, initiating the Controller Manager.

(iii) Data Gathering polls every configured time to all activated plugins, allowing the data storage.

(iv) ODL Data Base is the specific MD-SAL data store from ODL.

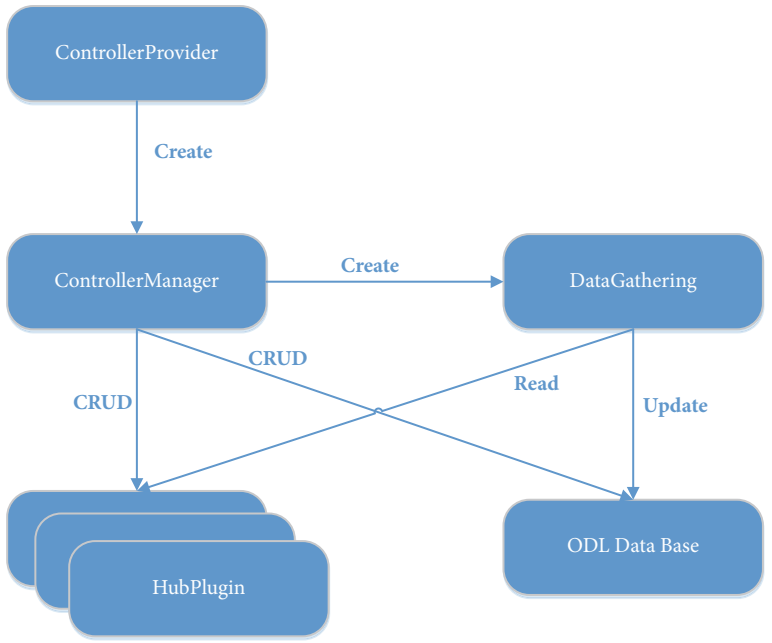


Figure 7: Modules hierarchy implemented in ODL.

(v) Hub Plugin is the abstract class which defines the behavior of each plugin and hub involved in the scenario.

5.3. Northbound Layer. The Northbound layer is the highest layer in the SDN architecture. It is composed of user applications. These integrate the so-called Northbound APIs. The main functionality of this layer is to give the users final applications with different purposes, for instance, data and device management in the network, device visualization in graphic interfaces, host tracker, flow programmer, or static routing.

Northbound applications communicate with the physical network in the Southbound through the SDN controller, which takes decisions about how to proceed. An example of Northbound API is given in Section 6, where a web application is developed with the aim of managing an IIoT network, composed of devices from different vendors. Other examples of Northbound APIs are shown in Figure 8. There is a wide variety, with different purposes: cloud orchestration tools based on OpenStack (https://www.openstack.org/), VMware (https://www.vmware.com/), or CloudStack (https://cloudstack.apache.org/) for instantiation of virtual machines in network elements; services to increase network security such as firewalls to filter traffic according to certain criteria based on ports, IP addresses, or services; network planning tools such as Net2Plan (http://net2plan.com/), to generate topologies, to collect statistics or to execute network optimization algorithms; applications of accounting and load balancing; applications focused on network operation and management, some of them based on CRUD services (Create, Read, Update, and Delete), or dashboards to show the topology and the current status of every device in the network.

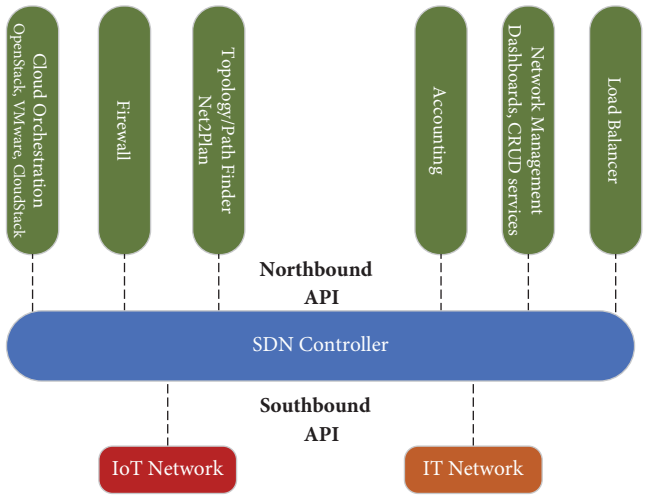


Figure 8: Northbound APIs examples.

6. IIoT Pilot Scenario Using ODLwith IoTDM and Plugins

In contrast to the works reviewed in Section 2, this work shows a pilot to test the proposed open source SDN architecture solution. This pilot provides light to the orchestration of an industrial scenario using SDN, testing the management of a set of lamps and sensors installed in an industrial warehouse, together with the IT network operating in the same factory. This scenario could be real in a medium-high factory, and it could be extended for managing any industrial technology, just following the software and hardware requisites described in Section 5.

Following the three layered architectures in Figure 1, extended in Figure 4, the southbound consists of a set of

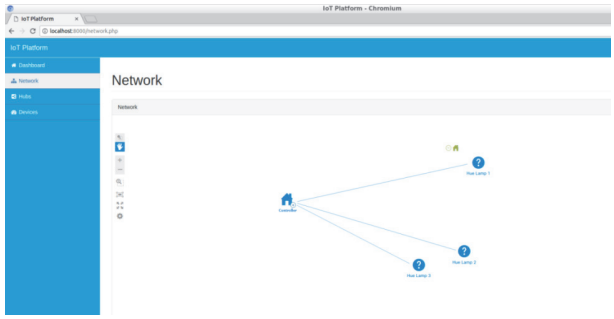


Figure 9: Dashboard of web interface for orchestrating IIoT networks.

smart lamps (Philips Hue (https://www.philips.co.uk/c-p/8718291775027/hue-personal-wireless-lighting/specifications)) and a set of intelligent sensors used in industrial scenarios for monitoring temperature, humidity, etc. (B+B SmartWorx (http://advantech-bb.com)). They work as an IIoT network. In the controller layer ODL with IoTDM, version Carbon 0.6.3 is executed in a local machine running Ubuntu- Linux. In the northbound a simple web application has been developed to test the communication among all actors in the architecture. The user-devices interactions are done in a background communication between the web interface and the controller layer. A snapshot of the web application is shown in Figure 9. It shows the visual management of those devices connected to the IIoT network and allows users to create, modify, and/or delete entities related to the deployed scenario. The web tool offers four main functionalities:

(i) Dashboard: main application screen, showing general information about the platform as the number of devices (hubs) to manage.

(ii) Network: showing the topology deployed.

(iii) Hubs: screen to manage hubs via CRUD.

(iv) Devices: screen to visualize and configure devices.

The pilot has been tested for orchestrating the single IIoT network, the IT network and both together with a single controller.

7. Conclusions

On the one hand, this work has identified the main problems that could arise in the I4.0 adoption for a medium-high factory: the management of thousands of IoT devices of different technologies and vendors, the heterogeneity of communication protocols and data formats, the need of manage the traditional industrial machinery networks and IT networks together with new IoT infrastructures, and the risks of huge amount of data traffic in the I4.0 infrastructure, which could derive in undesirable packet delays and network congestion. After that, an open-source software solution architecture based on ODL together with IoTDM has been proposed to orchestrate the whole I4.0 infrastructure, enabling interoperability and management of IIoT devices from different vendors, including conventional industrial machinery and IT networks. The software architecture solution includes the use of different system of plugins in the southbound and in the controller layer. In the latter, the plugin is based on oneM2M standard that parse the JSON/XML format from IoTDM to a generic JSON/XML format understood by any application in the northbound. The system of plugins has been developed with modularity and scalability premises, giving an open software solution that allows anyone to be able to develop a new plugin for each technology involved in an industrial scenario. The software architecture presented in this work provides scalability and exportation to other industrial scenarios, breaking the vendor lock-in barriers and generating a vendor-agnostic solution.

This work gives light to the practical difficulties appearing in introducing SDN in industrial contexts for I4.0 adoption. It reviews and provides solution to critical aspects from the controller architectural choices, specific IoT interfaces, and the difficulties for covering different communication protocols popular in industrial contexts.

Data Availability

The paper proposes an open-source software architecture solution for "orchestrating" and Industrial IoT scenario. All software components are free (available) in the references the authors have included throughout the paper. Only two software components are not open-source (available) because they are still working on them. But the authors give all details about their structure: Section 5: controller layer: middleware parser and Section 6: northbound application. The authors are planning to upload the source code in the github repository of their research group: https://github.com/girtel/.

Disclosure

This publication reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work has been partially cofunded by the Spanish National Project ONOFRE-2 (ref. no. TEC2017-84423-C3-1-P), H2020 EU Project (ref. no. H2020-ICT-2016-2 (ID 761727) topic: ICT-7-2016), and Erasmus+ Program of EU (ref. no. 575853-EPP-1-2016-1-ES-EPPKA2-SSA).

References

[1] P. Gerbert, M. Lorenz, M. Rüßmann et al., *Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries*, Boston Consulting Group, 2015.

[2] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[3] S. Jeschke, C. Brecher, T. Meisen, D. Özdemir, and T. Eschert, "Industrial Internet of Things and Cyber Manufacturing Systems," in *Industrial Internet of Things*, Springer Series in Wireless Technology, pp. 3–19, Springer International Publishing, Cham, 2017.

[4] P. P. Ray, "A survey of IoT cloud platforms," *Future Computing and Informatics Journal*, vol. 1, no. 1-2, pp. 35–46, 2016.

[5] "5H2020 – UNIFY-IoT Project, Report on IoT platform activities," http://www.unify-iot.eu/wp-content/uploads/2016/10/D03_01_WP02_H2020_UNIFY-IoT_Final.pdf.

[6] R. Yang, Z. Wen, J. Xu, and M. Rovatsos, "Fog Orchestration for IoT Services: Issues, Challenges and Directions," *IEEE Internet Computing*, vol. 21, no. 2, pp. 16–24, 2016.

[7] D. Wilusz and J. Rykowski, "Orchestration of distributed hetero-geneous sensor networks and Internet of Things," *Informatyka Ekonomiczna*, vol. 3, no. 33, 2014.

[8] N. Matsuda, K. Takagi, S. Horiuchi, H. Aoki, and A. Akutagawa, "IoT network implemented with NFV," *NEC Technical Journal*, vol. 10, no. 3, pp. 35–40, 2016.

[9] N. Omnes, M. Bouillon, G. Fromentoux, and O. Le Grand, "A programmable and virtualized network & IT infrastructure for the internet of things: How can NFV & SDN help for facing the upcoming challenges," in *Proceedings of the 2015 18th International Conference on Intelligence in Next Generation Networks, ICIN 2015*, pp. 64–69, February 2015.

[10] N. Bizanis and F. A. Kuipers, "SDN and virtualization solutions for the internet of things: a survey," *IEEE Access*, vol. 4, pp. 5591–5606, 2016.

[11] A. Agarwal, G. Misra, and K. Agarwal, "The 5th Generation Mobile Wireless Networks- Key Concepts, Network Architecture and Challenges," *American Journal of Electrical and Electronic Engineering*, vol. 3, no. 2, pp. 22–28, 2015.

[12] Y. Li, X. Su, J. Riekki, T. Kanter, and R. Rahmani, "A SDN-based architecture for horizontal Internet of Things services," in *Proceedings of the ICC 2016 - 2016 IEEE International Conference on Communications*, pp. 1–7, Kuala Lumpur, Malaysia, May 2016.

[13] R. Vilalta, A. Mayoral, D. Pubill et al., "End-to-end SDN orchestration of IoT services using an SDN/NFV-enabled edge node," in *Proceedings of the 2016 Optical Fiber Communications Conference and Exhibition, OFC 2016*, USA, March 2016.

[14] R. Vilalta, A. Mayoral, R. Casellas, R. Martinez, and R. Muñoz, "SDN/NFV Orchestration of Multi-technology and Multi-domain Networks in Cloud/Fog Architectures for 5G Services, in Proc. of OptoElectronics and Communications Conference (OECC," in *Proceedings of the OptoElectronics and Communications Conference (OECC)*, Niigata, Japan, 2016.

[15] Y. Ma, Y. Chen, and J. Chen, "SDN-enabled network virtualization for industry 4.0 based on IoTs and cloud computing," in *Proceedings of the 2017 19th International Conference on Advanced Communication Technology (ICACT)*, pp. 199–202, Pyeongchang, Kwangwoon Do, South Korea, Feburary 2017.

[16] K. Sood, S. Yu, and Y. Xiang, "Software-defined wireless networking opportunities and challenges for internet-of-things: a review," *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 453–463, 2015.

[17] S. Bera, S. Misra, and A. V. Vasilakos, "Software-Defined Networking for Internet of Things: A Survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1994–2008, 2017.

[18] X. Li, D. Li, J. Wan, C. Liu, and M. Imran, "Adaptive Transmission Optimization in SDN-Based Industrial Internet of Things With Edge Computing," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1351–1360, 2018.

[19] W. Cerroni, C. Buratti, S. Cerboni et al., "Intent-based management and orchestration of heterogeneous openflow/IoT SDN domains," in *Proceedings of the 2017 IEEE Conference on Network Softwarization (NetSoft)*, pp. 1–9, Bologna, Italy, July 2017.

[20] J. wan, S. Tang, Z. Shu et al., "Software-Defined Industrial Internet of Things in the Context of Industry 4.0," *Proceedings of IEEE Sensors Journal*, vol. 16, no. 20, 2016.

[21] A. L. Stancu, S. Halunga, A. Vulpe, G. Suciu, O. Fratu, and E. C. Popovici, "A comparison between several Software Defined Networking controllers," in *Proceedings of the 12th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services, TELSIKS 2015*, pp. 223–226, Serbia, October 2015.

[22] R. Khondoker, A. Zaalouk, R. Marx, and K. Bayarou, "Feature-based comparison and selection of Software Defined Networking (SDN) controllers," in *Proceedings of the World Congress on Computer Applications and Information Systems (WCCAIS '14)*, January 2014.

[23] S. Rowshanrad, V. Abdi, and M. Keshtgari, "Performance evaluation of sdn controllers: Floodlight and OpenDaylight," *IIUM Engineering Journal*, vol. 17, no. 2, pp. 47–57, 2016.

[24] "EPCglobal, EPC Radio-Frequency Identify Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz - 960 MHz," https://www.gs1.org/sites/default/files/docs/epc/uhfc1g2_1_2_0-standard-20080511.pdf.

[25] *Control Area Network standard (road vehicles)*, 2018, https://www.iso.org/standard/63648.html.

[26] "Open source OPC UA servers and clients," https://github.com/open62541/open62541/wiki/List-of-Open-Source-OPC-UA-Implementations.

[27] A. Tavakoli, M. Casado, T. Koponen, and S. Shenker, "Applying NOX to the Datacenter," in *Proceedings of the 8th ACM Workshop on Hot Topics in Networks (HotNets-VIII)*, vol. 2009, New York, NY, USA.

[28] "Open source BLE to JSON parser," https://github.com/mfgCode/ble-packet-parser.

[29] "Open source modbus-tcp parser," https://www.npmjs.com/package/modbus-tcp.

[30] "Open source ecatmod parser," https://www.npmjs.com/package/ecatmod.

[31] "Open source libcanardbc parser," https://github.com/Polyconseil/libcanardbc.

[32] "OPC Fundation Publish-Subscribe release," https://opcfoundation.org/news/press-releases/opc-foundation-announces-opc-ua-pubsub-release-important-extension-opc-ua-communication-platform/.

[33] "ODL controller: Model Driven SAL," https://wiki.opendaylight.org/view/OpenDaylight_Controller.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 6934825, 8 pages
https://doi.org/10.1155/2018/6934825

Research Article

Pipeline Leak Aperture Recognition Based on Wavelet Packet Analysis and a Deep Belief Network with ICR

Xianming Lang,[1,2] Zhiyong Hu,[3] Ping Li,[1] Yan Li,[2] Jiangtao Cao,[1] and Hong Ren[4]

[1]*School of Information and Control Engineering, Liaoning Shihua University, Fushun 113001, China*
[2]*School of Automation, Northwestern Polytechnical University, Xi'an 710072, China*
[3]*School of Petrochemical Engineering, Liaoning Shihua University, Fushun 113001, China*
[4]*CNPC Northeast Refining & Chemical Engineering Co. Ltd. Shenyang Company, Shenyang 110167, China*

Correspondence should be addressed to Zhiyong Hu; huzhiyong024@163.com

Received 5 April 2018; Accepted 27 June 2018; Published 16 August 2018

Academic Editor: Houbing Song

The leakage aperture cannot be easily identified, when an oil pipeline has small leaks. To address this issue, a leak aperture recognition method based on wavelet packet analysis (WPA) and a deep belief network (DBN) with independent component regression (ICR) is proposed. WPA is used to remove the noise in the collected sound velocity of the ultrasonic signal. Next, the denoised sound velocity of the ultrasonic signal is input into the deep belief network with independent component regression (DBN_{ICR}) to recognize different leak apertures. Because the optimization of the weights of the DBN with the gradient leads to a local optimum and a slow learning rate, ICR is used to replace the gradient fine-tuning method in conventional DBN for improving the classification accuracy, and a Lyapunov function is constructed to prove the convergence of the DBN_{ICR} learning process. By analyzing the acquired ultrasonic sound velocity of different leak apertures, the results show that the proposed method can quickly and effectively identify different leakage apertures.

1. Introduction

Because of aging pipelines, corrosion, or welding defects, small leaks and slow leaks occur frequently; such leaks represent risks to the environment and can cause financial losses [1–4]. The pressure drop produced by a small leakage is low and difficult to detect; however, small leak and slow leak are main forms of leakage of long distance oil pipelines during the service period. Determining how to identify the small leakage aperture of the pipeline in time has become a popular topic of study in the management of the integrity of a pipeline [5, 6]. Thus, it is important to estimate the leakage aperture, to assist in the development of the pipeline repair plan and in the evaluation of the leakage area.

In a Kneser liquid, the transmission speed of an ultrasonic wave changes as the liquid pressure changes at a definite temperature [7]. In this study, the sound velocity of an ultrasonic signal of a pipeline is used to identify different leakage apertures. In practical engineering works, the general signals collected by data acquisition equipment include noises in the external environment, and the purpose of signal denoising is to distinguish the high-frequency signals from the interference caused by the high-frequency noise and to remove the high-frequency interference signal, as well as retain the useful information in the signal. The weak signal must be guaranteed to not be filtered out in the process of signal analysis, whereas the original signal can be reduced as much as possible. Wavelet packet analysis (WPA) presents a powerful ability for denoising a signal from the measured signals [8]. However, this method requires selection of a suitable wavelet basis function to achieve better denoising effect. When the measured signals are decomposed using empirical mode decomposition (EMD), because of the influence of the end effect, it is very difficult to accurately reconstruct the signal [9]. The local mean decomposition (LMD) method is more relaxed than EMD in terms of decomposition conditions, as the end effect is reduced and the over envelope phenomenon is avoided in the decomposition process [6, 10]. However, the end effect still affects the signal reconstruction. Recently,

variable mode decomposition (VMD) has been proposed as an adaptive signal decomposition method, which is an entirely nonrecursive signal decomposition method. VMD not only has a good separation effect of signal and noise, but can also effectively suppress modal aliasing; nevertheless, the end effect also affects the signal reconstruction [11, 12].

Theoretically, a deep belief network (DBN) [13] is composed of multiple restricted Boltzmann machines (RBMs), in contrast with a shallow learning model, such as ANN and SVM. Recently, the DBN is becoming a useful tool for classification [14, 15]. The most significant difference between deep learning methods and shallow learning methods is that the former can present features from the original feature set automatically, instead of selecting feature manually. Two types of optimization methods of DBN are the adjustment of the depth structure and the optimization of the related parameters [16, 17]. The supervised learning algorithm of DBN is based on the backpropagation algorithm with a gradient; as a result, the weight adjustment can easily fall into a local optimum and slow the learning speed, thereby affecting the classification accuracy. Determining how to select the optimal weights and avoid the gradient of the fine adjustment method is the key to improve the classification accuracy. To overcome these difficulties, it is necessary to find a learning algorithm without the gradient. Independent component regression (ICR) is a layer-to-layer supervised parameter regression model without the gradient [18, 19]. Therefore, a weight optimization method based on $\mathrm{DBN}_{\mathrm{ICR}}$ is proposed, to avoid the local optimum brought by the gradient algorithm, and the classification accuracy of DBN is further improved.

The rest of this paper is organized as follows. Section 2 introduces a method of aperture identification. In Section 3, the experimental results for pipeline are analyzed and discussed. Finally, Section 4 draws the main conclusions.

2. Leak Apertures Identification Method

In this section, WPA and $\mathrm{DBN}_{\mathrm{ICR}}$ are first introduced, and then, a combination of WPA and $\mathrm{DBN}_{\mathrm{ICR}}$ is proposed for different leakage apertures identification of a pipeline. Afterwards, an illustration is provided of the aperture identification process based on the time domain sound velocity of ultrasonic signal through WPA and $\mathrm{DBN}_{\mathrm{ICR}}$.

2.1. Principle of Wavelet Packet Analysis-Based Denoising. WPA [20–22] is an effective time-frequency analysis technique for nonstationary signals, multiresolution analysis of the wavelet packet transform decomposes signals into low frequency signals and high-frequency signals. Furthermore, the process continues to decompose the following layers until reaching the preset level; as a result, the wavelet packet analysis has obviously better ability to achieve accurate local analysis. Moreover, wavelet packet analysis has better characteristics for further segmentation and refinement of the frequency band broadened with the increase of scale. Therefore, WPA is a precise analysis method with the characteristics of broad high-frequency bandwidth and narrow

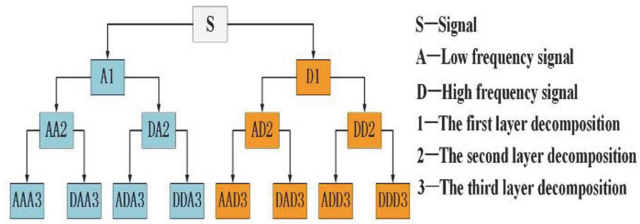


Figure 1: Wavelet packet decomposition of the signal S.

low frequency bandwidth. When the signal is decomposed by wavelet packets, a variety of wavelet basis functions can be adopted. For example, the signal S is decomposed by wavelet packets using three-layer decomposition [23], as shown in Figure 1.

The signal S is decomposed by the wavelet packet decomposition tree and can be represented as

$$\mathrm{S} = \mathrm{A1} + \mathrm{AAD3} + \mathrm{DAD3} + \mathrm{DD2} \tag{1}$$

2.2. Deep Belief Networks with Independent Component Regression. First, the independent component regression (ICR) is presented. Next, DBN with ICR and its learning algorithm are presented.

2.2.1. Independent Component Regression. Independent component analysis (ICA) [24] is a statistical and computational technique for revealing the hidden factors of signals; the goal of ICA is to decompose the observed data linearly into a statistical independent component. Denote the process observation samples as $(x_i, y_i)_{i=1,2,\cdots,N}$; the samples include independent variables $x_i \in \mathfrak{R}^{\mathrm{c}\times 1}$ and dependent variable $\mathrm{y}_i \in \mathfrak{R}^{\mathrm{p}\times 1}$. After the independent components have been estimated from ICA, the ICR model can be obtained.

According to the ICR algorithm, it is assumed that the measured process variables $x_i \in \mathfrak{R}^{\mathrm{c}\times 1}$ can be expressed as linear combinations of $r(\leq \mathrm{c})$ unknown independent components $s_i \in \mathfrak{R}^{\mathrm{r}\times 1}$; the ICR model is given by

$$x_i = \mathrm{A}_i s_i + \mathrm{e}_i \tag{2}$$

where $\mathrm{A}_i \in \mathfrak{R}^{\mathrm{c}\times \mathrm{r}}$ is the mixing matrix and $\mathrm{e}_i \in \mathfrak{R}^{\mathrm{c}\times 1}$ is the residual vector.

Thus, the independent components can be estimated as

$$\widehat{s}_i = \mathrm{W}_i x_i \tag{3}$$

where $\mathrm{W}_i \in \mathfrak{R}^{\mathrm{r}\times \mathrm{c}}$ is the separating matrix.

$$y_i = \mathrm{Q}_i^{\mathrm{T}} \widehat{s}_i \tag{4}$$

Thus, the linear regression matrix can be expressed as shown:

$$\mathrm{Q}_i = \left(\widehat{s}_i^{\mathrm{T}} \widehat{s}_i\right)^{-1} \widehat{s}_i^{\mathrm{T}} y_i \tag{5}$$

Therefore, the desired model between x_i and y_i can be obtained by the ICR model, which is given as

$$y_i = \mathrm{Q}_i^{\mathrm{T}} \mathrm{W}_i x_i = T_{ICR_i} x_i \tag{6}$$

where $\mathrm{Q}_i^{\mathrm{T}} \mathrm{W}_i = T_{ICR_i}$.

2.2.2. Deep Belief networks with Independent Component Regression Learning Process. Please refer to the drawing [25–27] for more information on DBN. ICR not only is a method of regression, but is also regarded as a supervised learning algorithm. DBN with ICR supervised fine-tuning starts from the classifier layer, and an alternative ICR method is repeatedly used to model every two hidden layers from the top layer to the bottom layer. Therefore, the ICR algorithm can be used to replace the gradient-based supervised learning; the goal of $\mathrm{DBN_{ICR}}$ is to overcome the low-accuracy and time-consuming nature of DBN.

For the classifier layer and the last hidden layer, the detailed training steps are as follows:

(a) First, the state of the last hidden layer is extracted as the independent variable, and the classifier layer is used as the dependent variable.

It is assumed that the classifier matrix is p dimensional; the last hidden layer state matrix is q dimension; the N sample observation matrices are as follows:

$$Y_s = \begin{bmatrix} y_{s_{11}} & \cdots & y_{s_{1p}} \\ \vdots & \ddots & \vdots \\ y_{s_{N1}} & \cdots & y_{s_{Np}} \end{bmatrix}, \quad H_L = \begin{bmatrix} h_{L_{11}} & \cdots & h_{L_{1q}} \\ \vdots & \ddots & \vdots \\ h_{L_{N1}} & \cdots & h_{L_{Nq}} \end{bmatrix} \tag{7}$$

where Y_s is from the N samples and H_L is the last hidden layer state matrix, obtained by the hidden layer of the last RBM.

(b) Independent components $\hat{s}_L$ are extracted from the observation sample matrix H_L.

$$\hat{s}_L = W_L H_L \tag{8}$$

(c) The linear regression Q_L can be performed on $\hat{s}_L$ and the output matrix Y_s.

$$\mathrm{Q}_L = \left(\hat{s}_L^{\mathrm{T}} \hat{s}\right)^{-1} \hat{s}_L^{\mathrm{T}} Y_s \tag{9}$$

(d) Therefore, the ICR model can be expressed by

$$\hat{Y}_s = Q_L^{\mathrm{T}} \hat{s}_L = Q_L^{\mathrm{T}} W_L H_L \tag{10}$$

where the output weight matrix optimized by the ICR model is described by

$$W_{out} = Q_L^{\mathrm{T}} W_L \tag{11}$$

Note that the number of the independent components c_L is very important for the accuracy of ICR.

$$error_{c_L} = \left\| Y_s - \hat{Y}_s \right\| \tag{12}$$

where c_L is determined by minimizing the absolute error of prediction; the corresponding c_L is optimal.

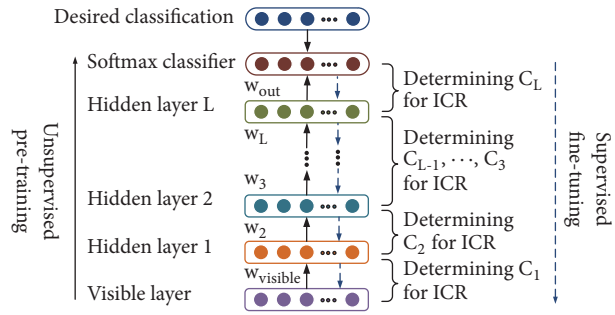


Figure 2: Architecture of $\mathrm{DBN_{ICR}}$.

Thus, the ICR fine-tuning process is completed between the output layer and the last hidden layer. Next, ICR is repeatedly conducted in every two hidden layers, starting form h_{L-1} and h_{L-2} to h_1 and x_{input}. As a result, the remaining weight matrices optimized by ICR are $W = (W_{out}\ W_L\ W_{L-1}\ \cdots\ W_2\ W_{visible})$.

The contrastive divergence (CD) algorithm is used to train each RBM from down layer to top layer first, and the fixed weights (initialization weights) after unsupervised training are determined. Next, the actual output is used to build the layer-layer ICR model and fine tune the initialization weights layer by layer; in the process of establishing the ICR model, the independent variable comes from the state variable of the RBM after the unsupervised training is completed.

The architecture of the $\mathrm{DBN_{ICR}}$ algorithm is shown in Figure 2.

2.3. Convergence Analysis. For the proposed $\mathrm{DBN_{ICR}}$, weight parameters are crucial to the convergence of $\mathrm{DBN_{ICR}}$. Thus, a theoretical proof for the convergence of $\mathrm{DBN_{ICR}}$ is described in this section. According to the learning process of $\mathrm{DBN_{ICR}}$, the whole dynamics transmission error can be described in [28]

$$\begin{aligned} \dot{E}_{error} &= \dot{y} - \dot{\hat{y}} \\ &= -E_{error} + F\left(x_{visible}, \overline{W}\right) - F'\left(x_{visible}, W\right) \end{aligned} \tag{13}$$

where $E_{error}(0) = 0$, and $F(\cdot)$ and $F'(\cdot)$ are the ideal deep belief network architecture function and the obtained deep belief network architecture function of $\mathrm{DBN_{ICR}}$ for the training samples, respectively. $W = (W_{out}\ W_L\ W_{L-1}\ \cdots\ W_2\ W_{visible})$ is the final weight derived from $\mathrm{DBN_{ICR}}$, and $\overline{W}$ is the ideal weight of $\mathrm{DBN_{ICR}}$ for the training samples.

According to [29], assuming that

$$\left\| F\left(x_{visible}, \overline{W}\right) - F'\left(x_{visible}, W\right) \right\| \le E_f \tag{14}$$

where $\| \cdot \|$ is the Euclidean distance, $E_f > 0$. Because $F(\cdot)$ is bounded, according to the architectural of deep belief networks, $F'(\cdot)$ is also bounded. Therefore, this assumption is achievable.

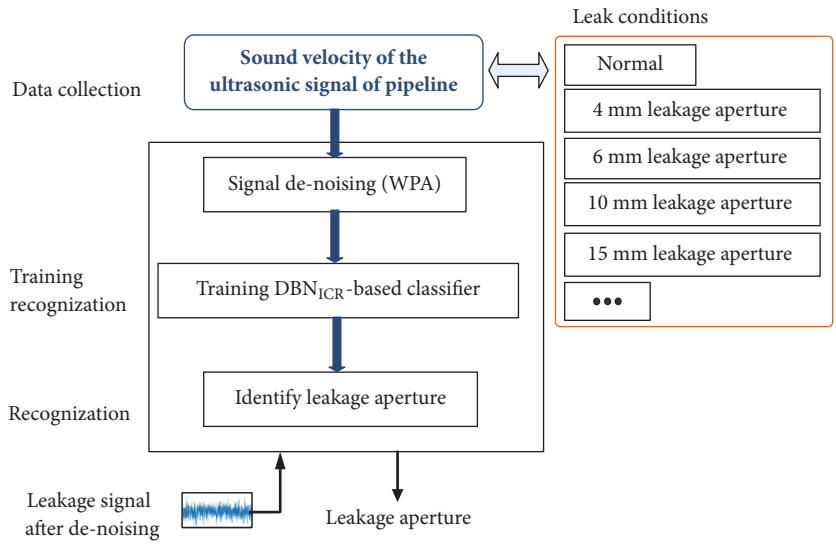


Figure 3: Schematic of the method to recognize pipeline leakage apertures.

Theorem 1. *Considering a stable system by (13) and (14), if DBN_{ICR} is used to recognize the different leak apertures, then the classification error E_{error} can converge to a finite vector: $E_{error} < E_m$ (E_m is a small positive number). Moreover, with the increase of samples and time, E_{error} is uniformly and ultimately bounded, and it approaches 0 when the training samples are adequate.*

Proof. Given the Lyapunov function

$$V\left(E_{error}\left(t\right)\right) = \frac{1}{2}E_{erroer}^{\mathrm{T}}\left(t\right)E_{error}\left(t\right) \tag{15}$$

The derivative of $V(E_{error}(t))$ is described as

$$\begin{aligned}\dot{V}\left(E_{error}\left(t\right)\right) &= E_{error}^{\mathrm{T}}\left(t\right)\dot{E}_{error}\left(t\right) = E_{error}^{\mathrm{T}}\left(t\right)\\ &\cdot\left(-E_{error}\left(t\right) + F\left(x_{input}\left(t\right), \overline{W}\right)\right.\\ &\left.- F'\left(x_{input}\left(t\right), W\right)\right) = -E_{error}^{\mathrm{T}}\left(t\right)^2 + E_{error}^{\mathrm{T}}\left(t\right)\\ &\cdot\left(F'\left(x_{input}\left(t\right), W\right) - F\left(x_{input}\left(t\right), \overline{W}\right)\right)\\ &\leq -E_{error}^{\mathrm{T}}\left(t\right)^2 + \left\|E_{error}\left(t\right)\right\|E_m = -\left\|E_{error}\left(t\right)\right\|\\ &\cdot\left(E_{error}\left(t\right) - E_m\right)\end{aligned} \tag{16}$$

To further analyze (16), it is discussed according to the following two cases:

(1) If $E_{error}(t) \geq E_m$, then $\dot{V}(E_{error}(t)) \leq 0$, and $\dot{V}(E_{error}(t))$ is negative semidefinite.

(2) If $E_{error}(t) < E_m$, then $\dot{V}(E_{error}(t)) > 0$, and $\dot{V}(E_{error}(t))$ is positive definite. However, as time goes on or the number of samples increase, there must be two types of situations:

$\|E_{error}(t)\| = E_m$ or $\|E_{error}(t)\| > E_m$; both situations are just same as that of item (1).

Thus, $\dot{V}(E_{error}(t))$ is negative semidefinite.

The training error sum AE is expressed by

$$AE = \sum_{i=1}^{N}\left(E_{error}^{\mathrm{T}}\left(t\right)E_{error}\left(t\right)\right) \tag{17}$$

According to the Lyapunov stability theorem,

$$\lim_{t\longrightarrow\infty} AE = 0 \tag{18}$$

where $t \longrightarrow \infty$ indicates that the number of the training samples is enough large and that all the training samples have been put into DBN_{ICR}.

Therefore, the convergence of DBN_{ICR} with respect to weight parameters is guaranteed theoretically. □

2.4. Leak Apertures Recognition Based on WPA and DBN_{ICR}. It has been shown that a deep belief network can achieve lower error rate compared to traditional methods in fault patterns classification [30]. Hence, using the advantages of both WPA and DBN_{ICR}, we propose a hybrid leakage aperture identification method. The sound velocity of ultrasonic signals can be denoised using WPA. These signals corresponding to different leakage apertures may be different; however, it is hard to differentiate the different apertures through pattern recognition without feature extraction based on prior knowledge.

Therefore, DBN_{ICR} is applied to identify the different leakage apertures via signals after WPA denosing. The schematic of the proposed method is shown in Figure 3.

3. Field Experiment and Analysis

Acquired sound velocities of the ultrasonic signals are first processed by WPA and then classified by DBN_{ICR}. In addition, a comparative study between the proposed and existing pipeline leakage aperture methods is performed.

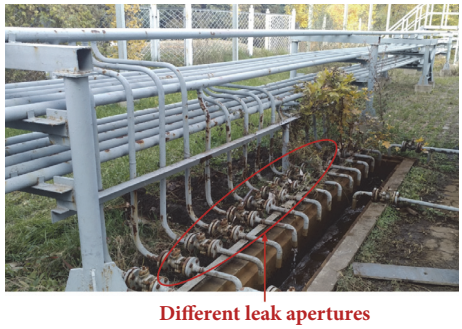


Figure 4: Different leak apertures.

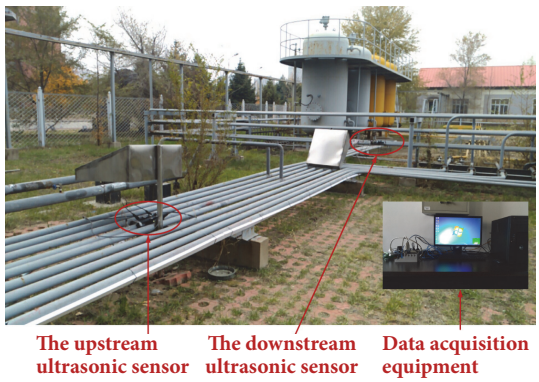


Figure 5: Experiment apparatus.

3.1. Field Experiment. Because of safety and cost problems, water instead of oil was used to simulate pipeline leaks in the experiment. According to the leakage experimental protocols for examining the aperture of different sizes, as shown in Figure 4, the sound velocities of upstream or downstream ultrasonic signals were chosen to perform the test. The length of the pipeline segment is 2,800 m with an inner diameter of 50 mm, and the leakage apertures are 4 mm, 6 mm, 10 mm, and 15 mm. The operation conditions were as follows: (a) water was transported at 12 m^3/h; (b) different apertures of valves were installed to emulate leaks; (c) the leakage flow and the energy were added by the upstream pump (the lift of the pump was 120 m); (d) leaked water was stored by a tank. The experiment apparatus is shown in Figure 5. The WPA and DBN_{ICR} algorithms are tested in MATLAB environment. We acquired a database from ultrasonic equipment by National Instruments DAQ-9184, at a sampling rate of 100 Hz. All the methods are implemented in MATLAB R2014a on a PC with an Intel Pentium processor (2.90 GHz) and 6 GB RAM.

3.2. Sound Velocity of Ultrasonic Processing and Aperture Recognition. The sound velocities of the ultrasonic signals from different apertures were acquired by an ultrasonic sensor at the end of the pipeline; these velocities were used as the database to verify the proposed algorithm. The acquired signals are shown in Figure 6.

According to the measured sound velocities of the ultrasonic signals, they are decomposed by WPA of best tree with

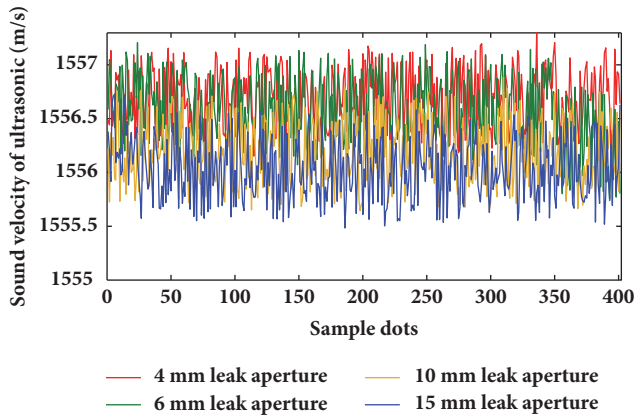


Figure 6: Ultrasonic signals under different leakage apertures.

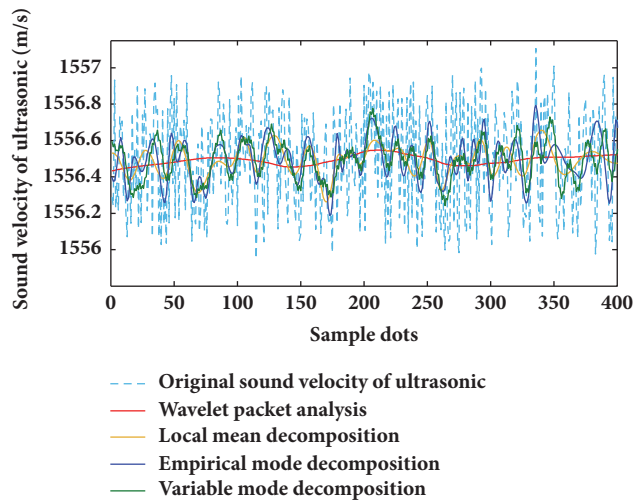


Figure 7: Pipeline inlet sound velocities of the ultrasonic signals of the 4 mm leakage aperture.

db3 in the three-layer model. Using the best tree structure, the sound velocity of the ultrasonic signal is reconstructed. To validate denoised signals, which are reconstructed signals obtained via the wavelet packet, the results of the signal denoised with VMD, EMD, and LMD are as shown in Figure 7.

Figure 7 shows that the wavelet packet analysis presents a powerful ability for denoising the signal from the measured sound velocities of the ultrasonic signal, compared with other methods, such as EMD, LMD, and VMD. When the measured sound velocities of the ultrasonic signals are reconstructed with EMD, because of the influence of end effect, it is very difficult to accurately remove the noise. The LMD method is more relaxed than EMD method in terms of decomposition conditions as the end effect is reduced and the over envelope phenomenon is avoided in the reconstruction process. However, the end effect still affects the extraction for removing the noise. VMD can adaptively extract the intrinsic modes of original sound velocities of the ultrasonic signal, but the end effect also affects the reconstruction signal.

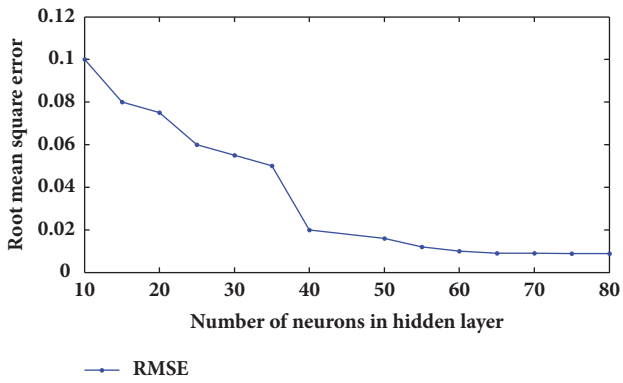


Figure 8: Relation curve between the number of neurons in the hidden layer and the classification error.

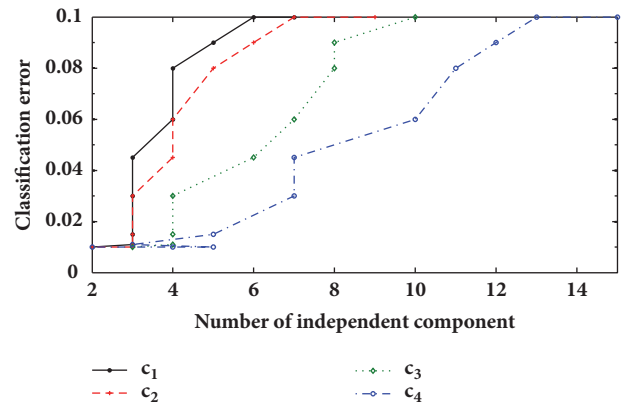


Figure 9: Relation curve between the number of independent components and the classification error.

Therefore, WPA is used to denoise the sound velocities of the ultrasonic signals at the end of the pipeline; the sound velocities of the ultrasonic signals of normal condition and for leakage apertures of 4 mm, 6 mm, 10 mm, and 15 mm are collected by ultrasonic equipment. Five cases are created, and 1,000 samples from each case are chosen for training the $\mathrm{DBN}_{\mathrm{ICR}}$ and the DBN. Thus, 5,000 samples of sound velocities of the ultrasonic signals are selected. Moreover, 4,000 samples are chosen as training samples, and the others are used as the testing samples.

In this simulation, the denoised sound velocities of the ultrasonic signals are the inputs of the $\mathrm{DBN}_{\mathrm{ICR}}$ and DBN, and the five feature vectors (absolute mean value, effective value, kurtosis, plus factor, and peak value factor) are input into the least squares twin support vector machine (LSTSVM) [31], the least squares support vector machine (LSSVM), the support vector machine (SVM), and the back propagation neural network (BPNN). The feature vectors of 4,000 groups (800 of each condition) are chosen as training samples and the others are used as testing samples. This paper adopts the "One-against-All" algorithm (OAA) of multiclassification.

We follow the experience in the architecture selection of DBN: the number of neurons of the next layer is smaller than that of the previous layer so that the process of the DBN can be a feature extraction process. In this study, the three hidden layers of $\mathrm{DBN}_{\mathrm{ICR}}$ are selected by simulation. Moreover, to determine the number of neurons of the hidden layer, a trail-and-error approach is chosen to demonstrate the relation between the number of neurons in the hidden layer and the classification error. The relation curve between the number of neurons in the hidden layer and the classification error is shown in Figure 8. Figure 8 reveals that the best number of neurons in hidden layer 3 is 65, and the corresponding root mean square error (RMSE) is 0.01; the RMSE is given as

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2} \tag{19}$$

where y_i is the desired value, $\hat{y}_i$ is the output value of $\mathrm{DBN}_{\mathrm{ICR}}$, and N is the number of testing samples.

According to the above analysis, the architecture of $\mathrm{DBN}_{\mathrm{ICR}}$ and and DBN is selected as 400-200-100-65-5. To improve the accuracy of fine-tuning, the number of independent components must be set as $c = \{c_4 \;\; c_3 \;\; c_3 \;\; c_1\}$; the trail-and-error approach is used to determine the optimal number of independent components, and the relation curve between number of independent components and the classification error is shown in Figure 9.

Figure 9 reveals that the best numbers of independent components are $\{c_4 = 4 \;\; c_3 = 3 \;\; c_3 = 2 \;\; c_1 = 2\}$, and the corresponding classifier error is 0.01. Thus, the $\mathrm{DBN}_{\mathrm{ICR}}$ is constructed with the optimal number of independent components. After unsupervised learning and supervised learning, we use $\mathrm{DBN}_{\mathrm{ICR}}$ to recognize different leakage apertures. To efficiently demonstrate the proposed method of $\mathrm{DBN}_{\mathrm{ICR}}$, the input and output for DBN are the same as those for $\mathrm{DBN}_{\mathrm{ICR}}$; moreover, 4,000 samples results using LSTSVM, LSSVM, SVM, and BPNN are obtained for comparison with the simulation. Practically, DBN is the traditional DBN.

The DBN's weights are initialized randomly, the biases are initialized to zero, the maximum iterations are 1, the batch size is 1, and the learning rate is 0.4. In LSTSVM, we set the slack variable $a_1 = a_2$ to 0.01, and the kernel parameter $\sigma_1 = \sigma_2$ to 1. The OAA method is used to accomplish multi- classification. We employ LS-SVMlab to implement the multiclassifier of LSSVM, where the slack variable α is 1 and the kernel parameter σ is 1. We also employ Libsvm to implement the multiclassifier of SVM, where the slack variable α is 10 and the kernel parameter σ is 0.1. This paper chooses a three-layer BPNN for which the middle layer node number is 100, using 100 iterations, the learning rate of 0.5, and the minimum error of 1×10^{-5}. A comparison of the testing results with LSTSVM, LSSVM, SVM and BPNN is shown in Table 1.

The experimental results are listed in Table 1. Table 1 reveals that $\mathrm{DBN}_{\mathrm{ICR}}$ has the best results in terms of average training time. Note that, although the accuracy of BPNN is as good as that of DBN, the time domain signals of sound velocities of the ultrasonic signals without feature selection by experience were input into DBN and $\mathrm{DBN}_{\mathrm{ICR}}$. Table 1 also reveals that the LSTSVM improves the accuracy

Table 1: Simulation results of different methods in different leakage apertures.

methods	Neurons number	Recognition accuracy (%)	Average running-time(s)	
			Training time	Testing time
DBN_{ICR}	400-200-100-65-5	98.98	42.32	10.25
DBN	400-200-100-65-5	98.7	58.25	15.63
LSTSVM	-	98.58	16.23	4.11
LSSVM	-	98.42	20.61	6.75
SVM	-	98.31	22.15	7.04
BPNN	400-100-5	98.91	15.13	4.41

and the average running time than LSSVM and SVM; the accuracy of LSSVM is the same as that of SVM, but the average running time of LSSVM is less than SVM. Compared with DBN, DBN_{ICR} has improved in recognition accuracy of different leakage apertures and program running time, primarily because of the fine-tuning of ICR of DBN_{ICR}.

3.3. Results and Analysis. The classification rate is calculated as the ratio of the number of correctly classified test samples to the total number of test samples. The proposed method and the LSTSVM-based, LSSVM-based, SVM-based, and BPNN-based methods are used to identify the different leakage apertures. 5,000 trials are performed, where 80% of samples are randomly selected for training and other samples are used for testing.

In the simulation with the proposed method, the average accuracies of testing are 98.98%; i.e., all the different apertures are effectively recognized. Using the LSTSVM-based, LSSVM-based, SVM-based, and BPNN-based methods, however, the average testing accuracy is 98.58%, 98.42%, 98.1%, and 98.91%, respectively. This result implies that the proposed method obtains higher recognition accuracies and shows better robustness than the other methods in distinguishing the different apertures.

(1) DBN can process the sound velocities of ultrasonic signals of a pipeline in the time domain to recognize the different leakage apertures directly, without feature extraction and feature selection by prior knowledge. Thus, the intelligence of leak detection and leak aperture recognition is enhanced.

(2) In this paper, we have not studied the leak location, and the architecture selection of DBN. At present, the architecture of deep learning networks including the number of hidden layers and the number of neurons in each hidden layer is selected through empirical or experimental methods. Thus, this approach requires much work and may affect the accuracy or speed by architecture selection. The architecture adaptive selection is still a difficult problem to solve for deep neural networks; we will study this problem in the future.

4. Conclusions

In this paper, the method for leak aperture recognition of pipeline based on WPA and DBN_{ICR} was proposed. To effectively extract more valuable leakage information, WPA was applied to refine the measured sound velocity of the ultrasonic signal to design an original set. To achieve the desirable performance of leak aperture recognition and remove the requirement for manual feature selection, DBN_{ICR} was used as classifier. To investigate the effectiveness of the proposed method, it was tested on sound velocity of the ultrasonic data of an experimental pipeline to recognize the different leak apertures. The results showed that the proposed method can reliably recognize the different leakage apertures.

Data Availability

Because all data of the experiment will be used as the patent application of the project, we cannot share the experimental data.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 61673199.

References

[1] S. Datta and S. Sarkar, "A review on different pipeline fault detection methods," *Journal of Loss Prevention in the Process Industries*, vol. 41, pp. 97–106, 2016.

[2] G. Bolzon, T. Boukharouba, G. Gabetta, M. Elboujdaini, and M. Mellas, *Integrity of pipelines transporting hydrocarbons*, Springer, The Netherlands, 2011.

[3] M. Henrie, P. Carpente, and R. E. Nicholas, *Pipeline Leak Detection Handbook*, Gulf Professional Publishing, 2016.

[4] A. Martini, M. Troncossi, and A. Rivola, "Automatic Leak Detection in Buried Plastic Pipes of Water Supply Networks by Means of Vibration Measurements," *Shock and Vibration*, vol. 2015, Article ID 165304, 2015.

[5] J. Liu, H. Su, Y. Ma, G. Wang, Y. Wang, and K. Zhang, "Chaos characteristics and least squares support vector machines based online pipeline small leakages detection," *Chaos, Solitons & Fractals*, vol. 91, pp. 656–669, 2016.

[6] J. Sun, Q. Xiao, J. Wen, and F. Wang, "Natural gas pipeline small leakage feature extraction and recognition based on LMD envelope spectrum entropy and SVM," *Measurement*, vol. 55, no. 9, pp. 434–443, 2014.

[7] D. Wang, Z. Song, Y. Wu, and Y. Jiang, "Ultrasonic wave based pressure measurement in small diameter pipeline," *Ultrasonics*, vol. 63, pp. 1–6, 2015.

[8] J. Hu, L. Zhang, and W. Liang, "Detection of small leakage from long transportation pipeline with complex noise," *Journal of Loss Prevention in the Process Industries*, vol. 24, no. 4, pp. 449–457, 2011.

[9] C. Guo, Y. Wen, P. Li, and J. Wen, "Adaptive noise cancellation based on EMD in water-supply pipeline leak detection," *Measurement*, vol. 79, pp. 188–197, 2016.

[10] J. Sun, Q. Xiao, J. Wen, and Y. Zhang, "Natural gas pipeline leak aperture identification and location based on local mean decomposition analysis," *Measurement*, vol. 79, pp. 147–157, 2016.

[11] Z. Li, J. Chen, Y. Zi, and J. Pan, "Independence-oriented VMD to identify fault feature for wheel set bearing fault diagnosis of high speed locomotive," *Mechanical Systems and Signal Processing*, vol. 85, pp. 512–529, 2017.

[12] K. Dragomiretskiy and D. Zosso, "Variational mode decomposition," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 531–544, 2014.

[13] J. Li, X. Fan, G. Chen, Z. Gao, M. Chen, and L. Li, "A DNN for small leakage detection of positive pressure gas pipelines in the semiconductor manufacturing," in *Proceedings of the 2016 IEEE International Conference of Online Analysis and Computing Science, ICOACS 2016*, pp. 384–388, China, May 2016.

[14] Z. Zhang and J. Zhao, "A deep belief network based fault diagnosis model for complex chemical processes," *Computers & Chemical Engineering*, vol. 107, pp. 395–407, 2017.

[15] H. Shao, H. Jiang, H. Zhang, and T. Liang, "Electric Locomotive Bearing Fault Diagnosis Using a Novel Convolutional Deep Belief Network," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 3, pp. 2727–2736, 2017.

[16] S. Kamada and T. Ichimura, "An adaptive learning method of Deep Belief Network by layer generation algorithm," in *Proceedings of the 2016 IEEE Region 10 Conference, TENCON 2016*, pp. 2967–2970, Singapore, November 2016.

[17] T. Ichimura and S. Kamada, "Adaptive learning method of recurrent temporal deep belief network to analyze time series data," in *Proceedings of the 2017 International Joint Conference on Neural Networks, IJCNN 2017*, pp. 2346–2353, USA, May 2017.

[18] Y. Zhang and Y. Zhang, "Optimized independent components for parameter regression," *Chemometrics and Intelligent Laboratory Systems*, vol. 104, no. 2, pp. 214–222, 2010.

[19] Z. Ge, Z. Song, and P. Wang, "Probabilistic combination of local independent component regression model for multimode quality prediction in chemical processes," *Chemical Engineering Research and Design*, vol. 92, no. 3, pp. 509–521, 2014.

[20] T. L. T. Da Silveira, A. J. Kozakevicius, and C. R. Rodrigues, "Automated drowsiness detection through wavelet packet analysis of a single EEG channel," *Expert Systems with Applications*, vol. 55, pp. 559–565, 2016.

[21] D. Lei, L. Yang, W. Xu, P. Zhang, and Z. Huang, "Experimental study on alarming of concrete micro-crack initiation based on wavelet packet analysis," *Construction and Building Materials*, vol. 149, pp. 716–723, 2017.

[22] J. Li, X. Cui, H. Song, Z. Li, and J. Liu, "Threshold selection method for UWB TOA estimation based on wavelet decomposition and kurtosis analysis," *EURASIP Journal on Wireless Communications and Networking*, vol. 2017, no. 1, 2017.

[23] X. Lang, P. Li, Y. Li, and H. Ren, "Leak Location of Pipeline with Multibranch Based on a Cyber-Physical System," *Information*, vol. 8, no. 4, p. 113, 2017.

[24] C. Tong, T. Lan, and X. Shi, "Soft sensing of non-Gaussian processes using ensemble modified independent component regression," *Chemometrics and Intelligent Laboratory Systems*, vol. 157, pp. 120–126, 2016.

[25] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[26] L. Nie, D. Jiang, S. Yu, and H. Song, "Network traffic prediction based on deep belief network in wireless mesh backbone networks," in *Proceedings of the 2017 IEEE Wireless Communications and Networking Conference, WCNC 2017*, USA, March 2017.

[27] S. Jeschke, C. Brecher, H. Song, and D. B. Rawat, *Industrial internet of things cybermanufacturing systems*, Springer International Publishing Switzerland, 2017.

[28] H. Han and J. Qiao, "A self-organizing fuzzy neural network based on a growing-and-pruning algorithm," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 6, pp. 1129–1143, 2010.

[29] Y. Li, S. Tong, and T. Li, "Observer-based adaptive fuzzy tracking control of MIMO stochastic nonlinear systems with unknown control directions and unknown dead zones," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 4, pp. 1228–1241, 2015.

[30] L. Zhang, H. Gao, J. Wen, S. Li, and Q. Liu, "A deep learning-based recognition method for degradation monitoring of ball screw with multi-sensor data fusion," *Microelectronics Reliability*, vol. 75, pp. 215–222, 2017.

[31] X. Lang, P. Li, Z. Hu, H. Ren, and Y. Li, "Leak Detection and Location of Pipelines Based on LMD and Least Squares Twin Support Vector Machine," *IEEE Access*, vol. 5, pp. 8659–8668, 2017.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 5713913, 10 pages
https://doi.org/10.1155/2018/5713913

Research Article

LAB: Lightweight Adaptive Broadcast Control in DSRC Vehicular Networks

Linsheng Ye,[1] Linghe Kong,[1] Kayhan Zrar Ghafoor,[1] Guihai Chen,[1] and Shahid Mumtaz[2]

[1]*Shanghai Jiao Tong University, China*
[2]*Instituto de Telecomunicacoes, Portugal*

Correspondence should be addressed to Linghe Kong; linghe.kong@sjtu.edu.cn

Received 26 April 2018; Revised 16 July 2018; Accepted 5 August 2018; Published 13 August 2018

Academic Editor: Dongyao Jia

The Industrial Internet of Things (IIoT) is the use of Internet of Things (IoT) technologies in manufacturing. The vehicular ad hoc networks (VANETs) are a typical application of IIoT. Benefiting from Dedicated Short-Range Communication (DSRC) technology, vehicles can communicate with each other through wireless manner. Therefore, road safety is able to be greatly improved by the broadcast of safety messages, which contain vehicle's real-time speed, position, direction, etc. In existing DSRC, safety messages are broadcasted at a fixed frequency by default. However, traffic conditions are dynamic. In this way, there are too many transmission collisions when vehicles are too dense and the wireless channel is underused when vehicles are too sparse. In this paper, we address broadcast congestion issue in DSRC and propose lightweight adaptive broadcast (LAB) control for DSRC safety message. The objectives of LAB are to make full use of DSRC channel and avoid congestion. LAB meets two key challenges. First, it is hard to adopt a centralized method to control the communication parameters of distributed vehicles. Furthermore, the vehicle cannot easily acquire the channel conditions of other vehicles. To overcome these challenges, channel condition is attached with safety messages in LAB and broadcast frequency is adapted according to neighboring vehicles' channel conditions. To evaluate the performance of LAB, we conduct extensive simulations on different roads and different vehicle densities. Performance results demonstrate that LAB effectively adjusts the broadcast frequency and controls the congestion.

1. Introduction

Internet of Things (IoT) is an ecosystem of connected physical objects such as home appliances, vehicles, and robots, so that they have ability to access the Internet and communicate with each other [1]. Industrial Internet of Things (IIoT) is an application of the IoT to the manufacturing industry, which is believed to promote the development of manufacturing industry [2]. IIoT is also the foundation of industry 4.0 [3, 4]. Plenty of modern technologies are integrated into IIoT, such as cloud computing [5], wireless networks [6], artificial intelligence [7], autonomous vehicles [8], and data analysis, which makes it more powerful. The vehicular ad hoc networks (VANETs) [9] are a typical application of IIoT and make vehicles have the ability to communicate with each other. In recent years, Dedicated Short-Range Communication (DSRC) [10] is an emerging technology in VANETs, which provides the communication capability among vehicles [11]. It is reported that connected vehicles can help to avoid 74% of car crashes, which would save tens of thousands of lives and billions of dollars every year [12]. In addition, the US Federal Communication Commission (FCC) allocated 75MHz licensed spectrum at 5.9 GHz for DSRC, which can be used exclusively for vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications [13]. The DSRC spectrum is divided into 7 channels and the control channel (CCH) is exclusively reserved for safety communications [14].

Periodical broadcast of vehicles' safety messages is a key factor of road safety improvement. To let all neighboring vehicles within the transmission range know the condition of a vehicle, this vehicle broadcasts its safety messages to all single-hop vehicles on control channel. These safety messages provide detailed information like speed, position, direction,

acceleration, brake status, etc. [15]. The safety messages are broadcasted periodically, usually at a fixed frequency. The default periodicity is 0.1 seconds [16], as the recommendation of industry.

DSRC uses carrier sense multiple access with collision avoidance (CSMA/CA) at MAC layer as the fundamental multiaccess scheme [17]. When the density of vehicles within an area is too high, the channel is full of safety messages and collision rate increases rapidly. In this case, the vehicles cannot decode received collided safety messages and it becomes hard for the vehicle to know the status of neighbors. On the other hand, when the density of vehicles is relatively low, the channel is underused. To make full use of the channel resource, it is better if vehicles broadcast more safety messages. Therefore, broadcasting at fixed frequency does not fit in dynamic traffic circumstance. This issue becomes the motivation of our work.

To tackle this issue, there exist two challenges caused by the DSRC broadcast mechanism. First, vehicles are distributed to and independent of each other. So it is hard to use a centralized method to control the broadcast periodicity of all vehicles. Furthermore, since a vehicle does not know the channel conditions of other vehicles, it is not easy to set an accurate periodicity with only individual information.

In this paper, we propose a novel method, named *lightweight adaptive broadcast* (LAB) *control*, for DSRC safety message to make full use of channel and avoid channel congestion. LAB uses *channel detector* to assess channel condition and this information will be attached along with the safety message. *Broadcast manager* analyses neighbors' channel conditions and adjusts broadcast frequency (broadcast frequency represents the transmission frequency of safety message in this paper) correspondingly. In this way, the broadcast frequency is reduced when vehicle density is high and the broadcast frequency is increased when vehicle density is low. Furthermore, the challenges are overcome in a distributed way, because vehicles can acquire neighbors' channel conditions and make the decision of periodicity by themselves.

The contributions of this work are as follows:

(i) We propose and study a problem in DSRC that the broadcast frequency of safety messages should be carefully adjusted in dynamic traffic circumstance.

(ii) We propose a novel lightweight adaptive broadcast control (LAB) framework for DSRC safety message. LAB is a completely distributed scheme and vehicles broadcast their safety messages adaptively and efficiently under any conditions.

(iii) We conduct extensive simulations to evaluate LAB. Performance results demonstrate that LAB is an effective broadcast scheme. It works well on different vehicle densities and average received safety messages are up to 3 times compared with standard DSRC method. Moreover, it converges to global optimal state within 10s.

The rest of paper is organized as follows. In Section 2, we review the related work. Section 3 is problem statement of this paper. Section 4 presents the design details of our proposed scheme LAB. We evaluate performance of LAB in Section 5. Section 6 concludes our paper.

2. Related Works

In literature, plenty of solutions are proposed to control congestion in DSRC, which aim at enhancing the performance of vehicular networks.

Fei Ye et al. [18] proposed the congestion control and power control strategy that maximizes the efficiency, after analysing the broadcast efficiency under Rayleigh fading channel.

Soufiene Djahel et al. [19] proposed an algorithm that consists of three phases: assign priority to the safety messages; detect congestion; adjust transmit power and safety message transmission rate.

Lv Humeng et al. [20] proposed a distributed safety message frequency control algorithm adjusting the broadcast frequency according to the current network condition.

Mohamed Salah et al. [21] presented a congestion control algorithm based on the concept of dynamic priorities-based scheduling, to provide a reliable and safe communications system for VANETs.

M. A. Benatia et al. [22] proposed a Markov chain model to control congestion, which consists of four steps: priority assignment, buffer monitoring, congestion detection phase, and beacon transmission rate adjustment.

Bilal Munir Mughal et al. [23] proposed a conceptual view of a congestion control scheme which adjusts transmission rate and transmission power simultaneously for optimal congestion control.

Gaurav Bansal et al. [24] proposed a method which is based on LIMERIC [13] algorithm and achieves weighted fair message rate congestion control.

Tessa Tielert et al. [25] presented a design methodology for congestion control. Furthermore, a resulting rate adaption oriented protocol named PULSAR is proposed with the description and evaluation.

Gaurav Bansal et al. [26] proposed a congestion control method called EMBARC which adjusts the safety message transmission rate based on both channel load and vehicular dynamics.

In this work, we focus on a solution which can not only control channel congestion, but also make full use of channel resource and ensure fairness.

3. Problem Statement

In order to ensure road safety, vehicles equipped with DSRC devices broadcast their safety messages carrying their safety status among neighboring vehicles. By default, broadcast frequency is a fixed value, which cannot make full use of the channel and may lead to congestion. So it is necessary to propose a tailored adaptive broadcast control for DSRC safety message.

3.1. Notations. We summarize the notations of this paper as follows. We use R to denote a list of channel busy rates, and

Table 1: Summary of notations.

Notation	Definition
R	a list of vehicles' channel busy rates
r_i	channel busy rate of vehicle i
r	channel busy rate
r^*	optimal channel busy rate
v_i	vehicle i
s_i	safety message i
f	broadcast frequency
f'	adjusted broadcast frequency
F_{low}	frequency lower bound
F_{high}	frequency upper bound
$twnd$	time window
T_{idle}	idle time in a $twnd$
α	step size

r_i is the item of R. f denotes the broadcast frequency. $twnd$ denotes the time window. Other key notations in this paper are listed in Table 1.

3.2. Problem Formulation. Our objectives are to make full use of channel, avoid channel congestion, and guarantee fairness. To formulate these three objectives, we define r as the ratio of channel busy state to express the channel condition; use T_{idle} to denote idle time of channel in a time window $twnd$; use r^* to denote optimal channel busy rate. To make full use of channel, T_{idle} should be minimized; to avoid channel congestion, it should be guaranteed that the average of neighboring vehicles' channel busy rates approaches r^*; to guarantee fairness, the standard deviation of r should be minimized, so that each vehicle can have same chance to broadcast its safety messages. Therefore, the objectives can be formulated as

$$\begin{aligned} &\min T_{idle} \\ &\min \sigma(r) \\ &\overline{r} = r^* \end{aligned} \tag{1}$$

To make it adaptive, broadcast frequency should be adjusted according to real-time channel condition. Furthermore, to prevent the broadcast frequency from getting too high or too low, we should limit it within a range. It will not get more benefits from too high broadcast frequency while it is energy wasted. Meanwhile, for basic requirement of safety message broadcasting, broadcast frequency should have a minimum limit. In this way, we use F_{high} to denote frequency upper bound and F_{low} to denote lower bound. Based on the above analysis, we formulate the problem as

$$\begin{aligned} &\min T_{idle} \\ &\min \sigma(r) \\ &\overline{r} = r^* \end{aligned}$$

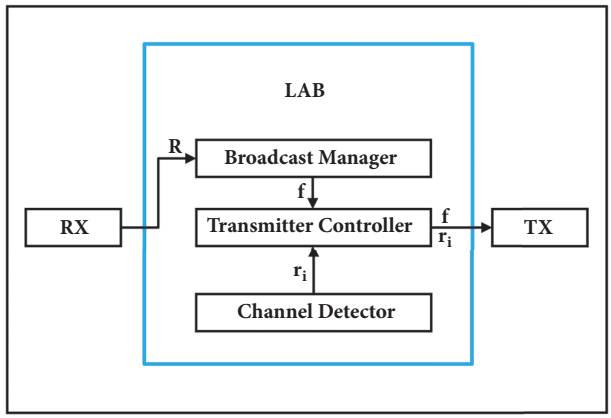


Figure 1: Overview of LAB.

$$\begin{aligned} &F_{low} < f < F_{high} \\ &0 \le r_i \le 1 \end{aligned} \tag{2}$$

4. Design of LAB

In this section, we introduce the design of our proposed lightweight adaptive broadcast (LAB) control in detail. Firstly, we present the overview on LAB. Then, we describe the channel detection mechanism. At last, we introduce the details of broadcast frequency control strategy.

4.1. Design Overview. To solve the problem formulated in (2), we design LAB, whose overview is shown in Figure 1. LAB has three core modules: *broadcast manager*, *transmitter controller*, and *channel detector*.

Channel detector is able to detect channel state and channel condition is conveyed to transmitter by *transmitter controller*. When transmitter gets channel condition, it will attach channel condition to safety messages, which contain safety information and other information. The format of the LAB frame is shown in Figure 2.

Broadcast manager will get the messages from receiver and fetch the carried contents. Then *broadcast manager* will analyse the channel conditions of nearby vehicles according to received messages, because channel condition is attached in these messages. At last, *broadcast manager* will make an adjustment for broadcast frequency and send the adjustment request to *transmitter controller*, to change broadcast frequency.

Transmitter controller controls the behavior of transmitter and transfers channel condition to transmitter.

Because the decision made by *broadcast manager* is based on channel condition, the objectives of making full use of channel and avoiding channel congestion are satisfied. At the same time, the objective of fairness is also satisfied for the consideration of nearby vehicles' channel conditions. The deeper analysis can be found in Section 4.3.

In the following subsections, we introduce the details of our design.

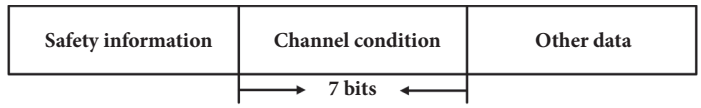


Figure 2: Packet format of safety message.

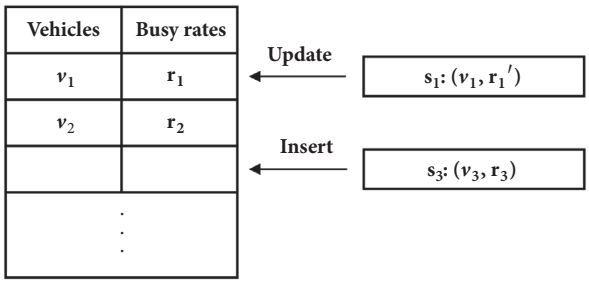


Figure 3: Busy rate table.

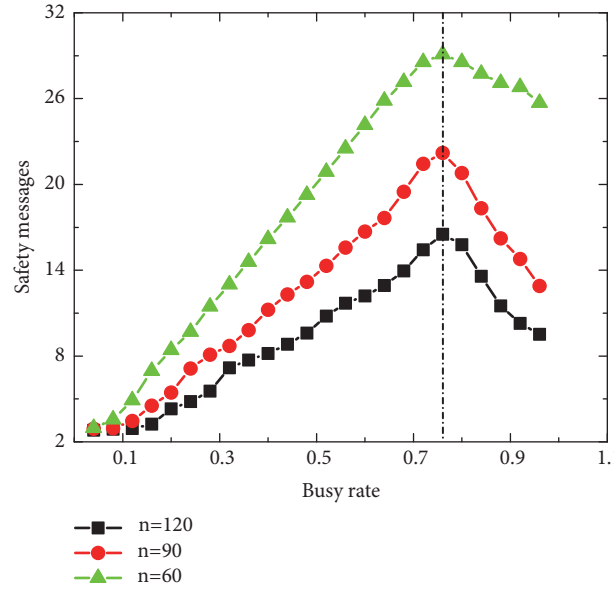


Figure 4: Safety messages on different channel busy rates.

4.2. Channel Detection. Channel condition can be quantified by many indicators. *Channel detector* chooses the channel busy rate, which is the ratio of channel when it is in a busy state. Busy state means channel is not idle and it may be transmitting, receiving, etc. *Channel detector* periodically detects channel state to quantify channel busy rate.

To quantify the busy state of the channel, a typical way is to conduct multiple sampling in a time slot. Because the busy state of the channel is dynamic, it is important to choose a suitable time slot. *Channel detector* detects the channel state every 10ms and maintains a time window (*twnd*). The length of time window is set to 1s by default. Every time it finds channel in a busy state within *twnd*, *channel detector* will record it. At the end of *twnd*, *channel detector* will send the channel busy rate to transmitter and start a new *twnd*.

It is worth noting that only 7 bits are appended to safety message, for conveying busy rate, as shown in Figure 2. The reason is that *channel detector* has 100 sampling points for each *twnd* and there are only 100 possible values for busy rate. Therefore, it contributes to LAB's lightweight features because it almost does not increase the communication overhead.

4.3. Broadcast Frequency Control. LAB adjusts broadcast frequency of transmitter according to neighboring vehicles' channel conditions. Whenever the receiver of vehicle receives a safety message from a neighbor, the content carried by this message will be fetched by *broadcast manager*.

Broadcast manager maintains a busy rate table and a time window (*twnd*). Every time a vehicle receives a safety message, *broadcast manager* will record channel busy rate into busy rate table, which indicates the channel condition of the vehicle that sends this message. As shown in Figure 3, if a safety message carries channel busy rate of a new vehicle, *broadcast manager* will insert a new entry into busy rate table; or it will update corresponding entry. Moreover, this busy rate table only records channel busy rates in current *twnd* and it will be reset at the end of *twnd*. Reset mechanism ensures that busy rate table only maintains the vehicles within transmission range in each *twnd*. The frequency adjustment decisions should be based on these vehicles. The overhead of reset is small because there are only hundreds of vehicles in transmission range at most typically.

One goal of LAB is to let vehicles receive valid safety messages as many as possible. We find that there exists an optimal busy rate r^*, in which the number of valid safety messages received by vehicle reaches the maximum value. It is easy to prove the existence of r^*. When the busy rate is close to zero, the channel is nearly idle and the vehicle receives few safety messages. On the contrary, when the busy rate is close to 1, the channel is too busy and full of safety messages, which means that most of messages are collided and the vehicle receives few valid safety messages. From the previous analysis, we can conclude the existence of an optimal channel busy rate r^*. We simulate some communication scenarios to verify this conclusion, as shown in Figure 4. We place n continuously moving vehicles at the crossroads in ns-3 [27]. n changes in each scenario and vehicles broadcast their safety messages in different frequencies. We estimate the number of packets received by each vehicle and corresponding channel busy rate. Figure 4 shows the relation between channel busy rate and average safety messages received from neighbors. We find that when busy rate reaches to about 0.76, vehicle gets most safety messages from neighbors in each scenario.

At the end of current *twnd*, *broadcast manager* will analyse maintained busy rate table and adjust broadcast frequency

Input: Current broadcast frequency: f, Frequency lower bound: F_{low}, Frequency upper bound: F_{high}
Output: Adjusted broadcast frequency: f'
(1) $f' = f + \lceil \alpha(r^* - r) \rceil$;
(2) **if** $f' < F_L$ **then**
(3) $f' = F_L$;
(4) **if** $f' > F_H$ **then**
(5) $f' = F_H$
(6) **return** f'

Algorithm 1: LAB: broadcast frequency control.

correspondingly. Frequency control algorithm should meet the objectives and should not break the limitations shown in (2). Therefore, we propose a control algorithm shown as Algorithm 1.

LAB adopts gradient descent method to adjust broadcast frequency:

$$f' = f + \lceil \alpha (r^* - r) \rceil \quad (3)$$

In the above equation, when the frequency converges to a value where the neighbors' average busy rate is equal to r^*, vehicle receives maximum safety messages.

It is worth mentioning that frequency should not break upper bound and lower bound. The reason for setting upper bound of broadcast frequency is that vehicles benefit less from broadcasting too many safety messages per second and it is energy wasted. Meanwhile the reason for setting lower bound of broadcast frequency is that vehicles must ensure a minimum broadcast frequency to spread their safety states.

In this way, vehicles can make full use of the channel to broadcast their safety messages. At the same time, the objective of fairness is guaranteed because LAB is a negative feedback process, whose causal chain is as follows:

$$f_i \uparrow \Longrightarrow r_{neighbors} \uparrow \Longrightarrow f_i \downarrow \quad (4)$$

Therefore, all vehicles have the same rights to broadcast their safety messages, which means the broadcast frequencies of single-hop neighbors are in a roughly same level.

On the contrary, a positive feedback will be trigged if control algorithm only uses vehicle's own channel busy rate to control frequency. The causal chain is shown as follows:

$$f_i \uparrow \Longrightarrow r_{neighbors} \uparrow \Longrightarrow f_{neighbors} \downarrow \Longrightarrow r_i \downarrow \Longrightarrow f_i \uparrow \quad (5)$$

Whenever *broadcast manager* decides to adjust broadcast frequency, it sends adjustment request to *transmitter controller* that controls the behavior of transmitter directly. *Transmitter controller* will adjust broadcast frequency of transmitter according to *broadcast manager*'s decision. At last, transmitter will broadcast safety messages in specified frequency. Furthermore, these safety messages carry vehicle's safety information got from safety system of vehicle and channel condition got from *channel detector*.

5. Evaluation

To validate the performance of LAB, we use ns-3 [27] to simulate vehicle's broadcasting on different scenarios. We generate vehicles' mobility traces using SUMO [28]. LAB is compared with standard DSRC method, unfair control method which adopts the vehicle's own channel busy rate to control frequency, and a distance based control method (denoted as DIST by us) proposed by Fallah et al. [29]. We create three types of roads in SUMO, which are crossroads, box road, and straight road. All roads in this simulation are bidirectional with three lanes in each direction. We use two-ray ground propagation loss model in ns-3 setting. We run the simulations 100s for each case. The data rate is set to 3Mbps, which is the default data rate of broadcast in DSRC [30]. The settings of other parameters are summarized in Table 2. The following subsections are the details of evaluation results.

Table 2: Simulation setting.

Parameters	Values
Number of lanes	3
Lane width	3.2 m
Transmission range	300 m
Data rate	3 Mbps
Communication frequency	5.9 GHz
Packet length	200 Bytes
DSRC broadcast frequency	10 Hz
F_{low}	5 Hz
F_{high}	30 Hz
r^*	0.76
$twnd$	1s
α	10

5.1. Performance on Different Densities. We test performance of LAB on different densities. Vehicles are moved in a crossroad being 1km in each direction and vehicle number ranges from 20 to 380.

Firstly, we test average safety messages received by each vehicle. The result is shown as Figure 5. We can easily find that LAB is better than standard DSRC and it receives more safety messages, because LAB controls broadcast frequency adaptively. Furthermore, the performance of LAB is similar to unfair control method. When vehicle number is larger than 220, DIST performs a bit better than LAB because DIST can reduce the transmission range.

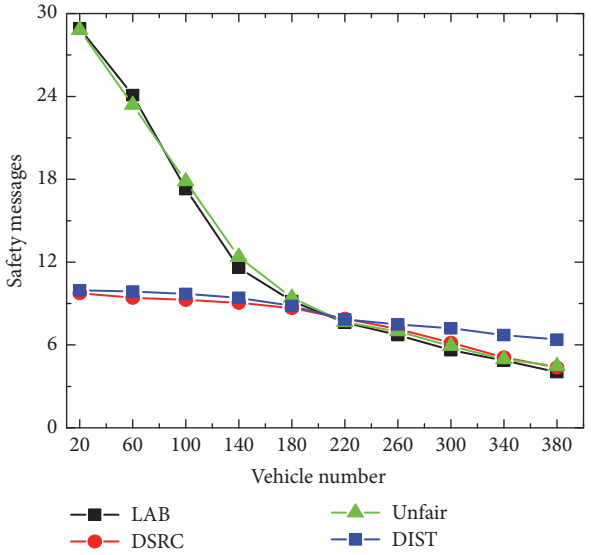


Figure 5: Safety messages on different vehicle densities.

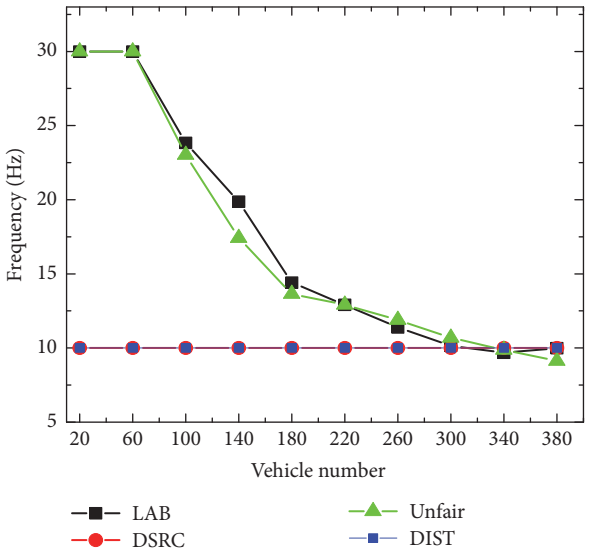


Figure 6: Broadcast frequency on different vehicle densities.

Figure 6 shows the average broadcast frequency on different vehicle densities. LAB's broadcast frequency drops from upper bound to lower bound with the increase of vehicle density while standard DSRC and DIST have no change. The curve of unfair method is also similar to LAB. The first two points of LAB are at frequency upper bound 30Hz because the channel is not fully utilized at that time.

It is worth mentioning that although the global performance is roughly the same between LAB and unfair method, we can find the difference between them regarding fairness in next section.

5.2. Fairness. Then, we test whether each vehicle is treated fairly in LAB and it is compared with unfair control method.

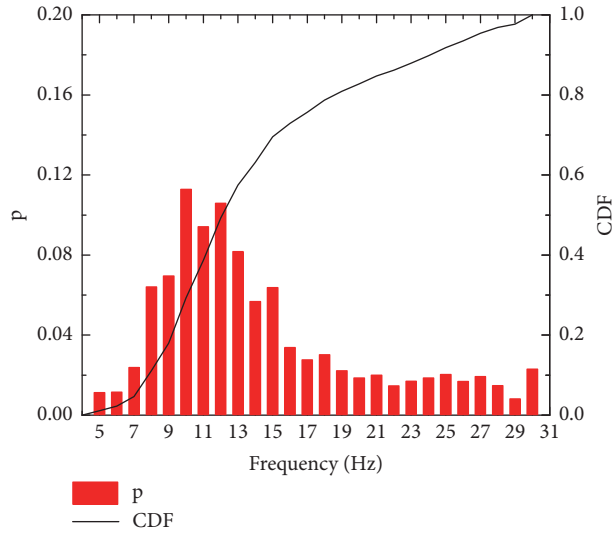


Figure 7: CDF of LAB method.

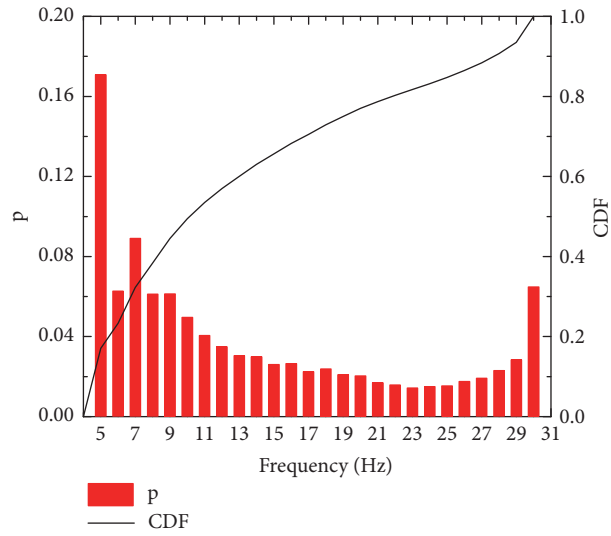


Figure 8: CDF of unfair method.

In this case, 180 vehicles are moved in a crossroad being 1km in each direction.

Figure 7 shows LAB's cumulative distribution function (CDF) and probability distribution of each vehicle's broadcast frequency. Figure 8 is for unfair method.

As shown in Figure 7, frequency of each vehicle mainly appears at median of all vehicles' frequencies. However, in Figure 8, frequency mainly appears at F_{high} and F_{low}. In this way, it can be concluded that LAB is more fair, which is the most important advantage compared with unfair method.

5.3. Time Trend. In this subsection, we check the performance changes of LAB over time, compared with standard DSRC method and unfair control method. In this test, 100 vehicles are moved in a crossroad being 1km in each direction.

Firstly, we test changes of average received safety messages over time. The result is shown as Figure 9. We find that

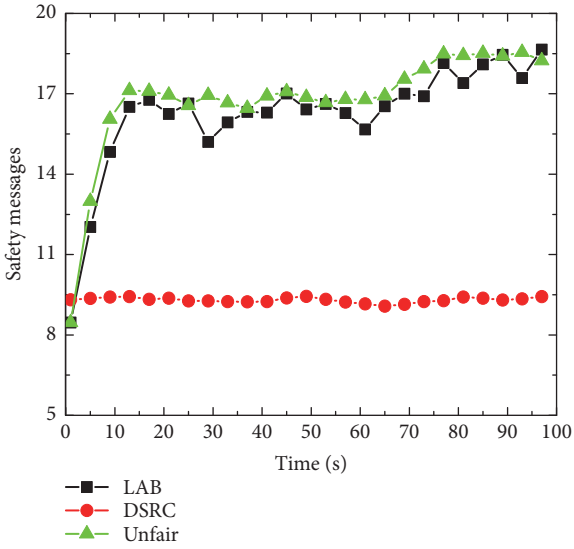


Figure 9: Time trend of safety messages.

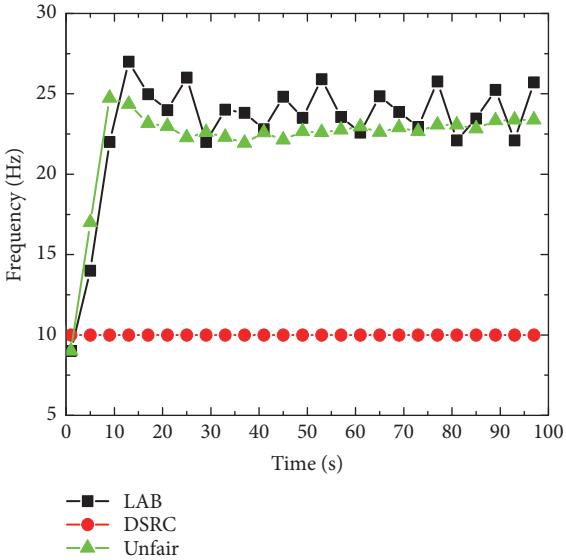


Figure 10: Time trend of broadcast frequency.

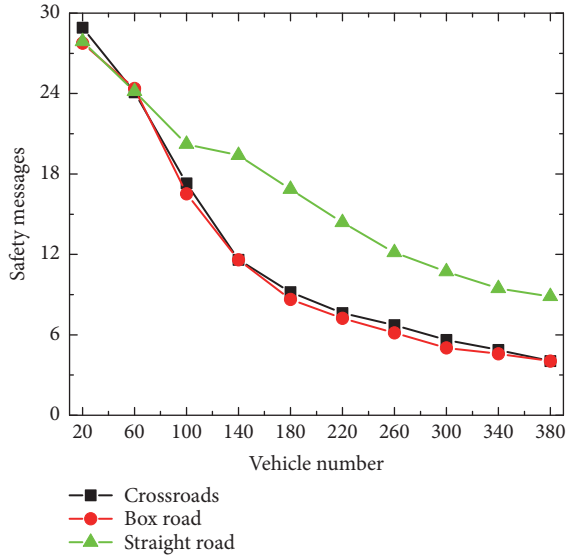


Figure 11: Safety messages on different roads.

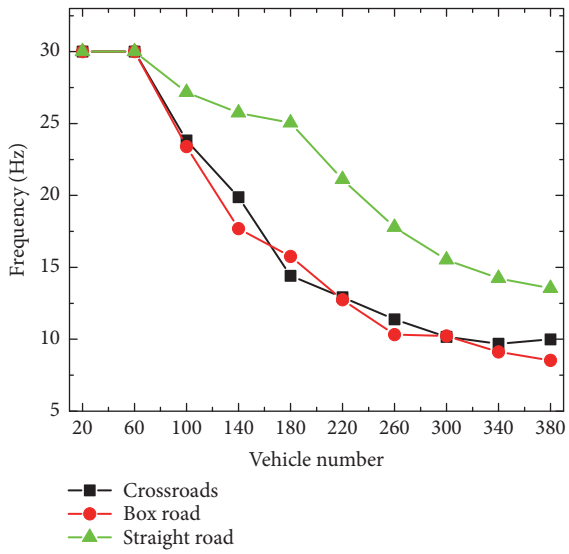


Figure 12: Frequency on different roads.

received safety messages of LAB increase to a stable level in 10s.

Then we test the time trend of average broadcast frequency as shown in Figure 10. We can also find that average broadcast frequency of LAB converges to the optimal state in 10s.

From previous results, we know that LAB is able to adapt to a changing environment and quickly converge to the optimal state.

5.4. Performance on Different Roads. In this subsection, we test performance of LAB on different roads. In this test, vehicle number ranges from 20 to 380.

The results are shown in Figures 11 and 12. We can find that the general trend on these three roads is roughly the same. Furthermore, performance on straight road is better than box road and crossroads because the intersection of them is more likely to congest.

5.5. Performance on Different Packet Lengths. In this subsection, we check the performance changes of LAB on different packet lengths. In this test, 100 vehicles are moved in a crossroad being 1km in each direction and the packet length changes from 50 Bytes to 700 Bytes.

The result is shown as in Figures 13 and 14. We find that the bigger the packet length, the fewer the safety messages that the vehicles can receive. It is because when packet length increases, it costs channel more time to send a packet. Therefore, broadcast frequency drops too.

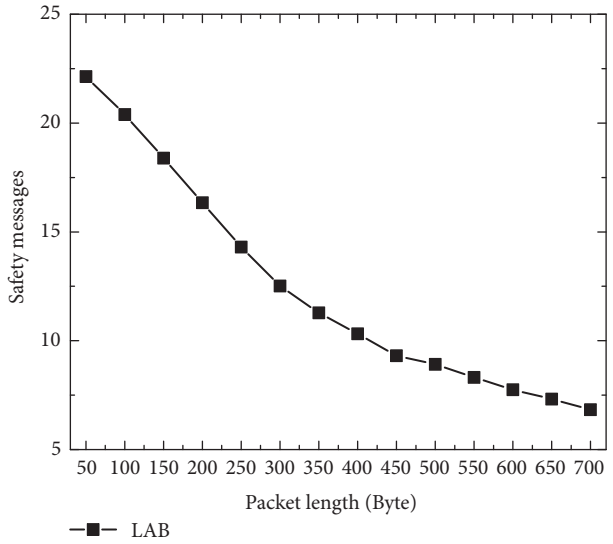


Figure 13: Safety messages on different packet lengths.

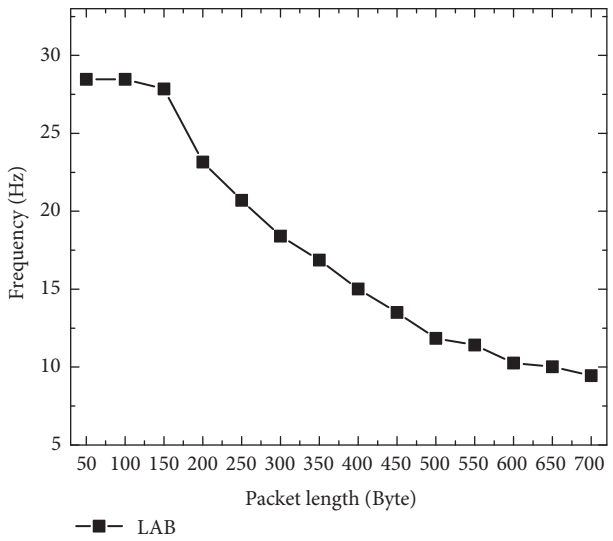


Figure 14: Frequency on different packet lengths.

5.6. Performance on Different Transmission Power. In this subsection, we check the performance changes of LAB on different transmission power. In this test, 100 vehicles are moved in a crossroad which is 1km in each direction and the transmission power changes from 6dBm to 28dBm.

We evaluate the frequency and average received packet number in Figures 15 and 16. We find that when the transmission power increases, broadcast frequency and the number of safety messages converge into a stable state.

5.7. Time Trend of Different Time Window. In this subsection, we check the time trend of LAB on different time window (*twnd*). In this test, 100 vehicles are moved in a crossroad which is 1km in each direction.

In Figures 17 and 18, the safety messages and frequency are evaluated, respectively. Four curves shown in figures

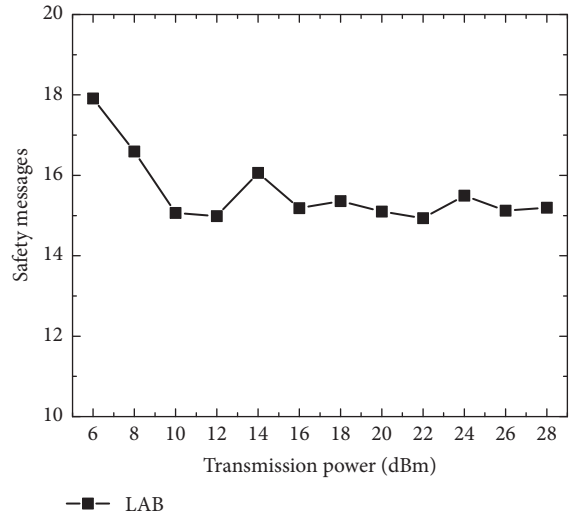


Figure 15: Safety messages on different transmission power.

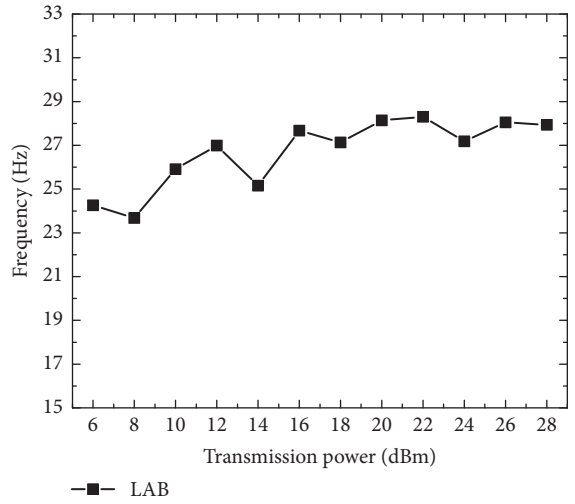


Figure 16: Frequency on different transmission power.

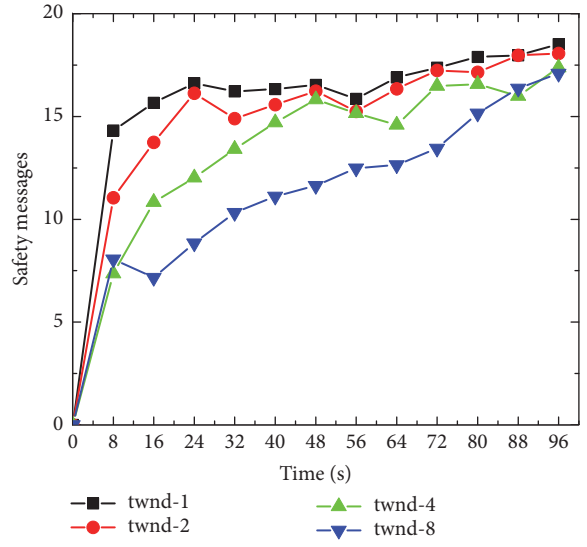


Figure 17: Time trend of safety messages.

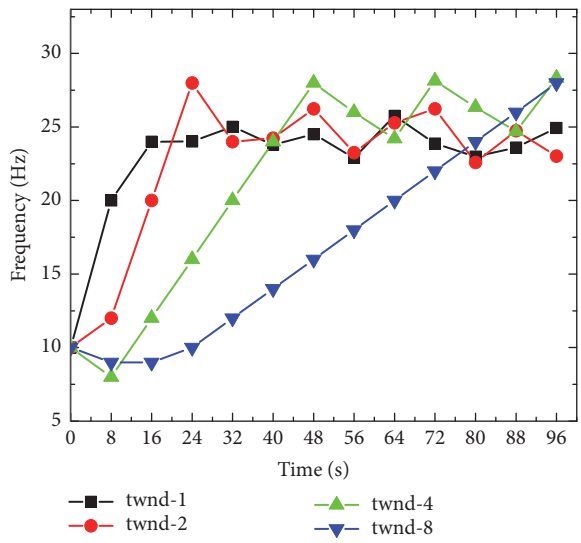


Figure 18: Time trend of broadcast frequency.

represent 4 sizes of *twnd*, which are 1s, 2s, 4s, and 8s. We find that the convergence speed drops when *twnd* becomes larger. Therefore, choosing 1s as the default setting of *twnd* is proper, which is tradeoff between overhead and sensibility.

6. Conclusion

In this paper, we propose lightweight adaptive broadcast control (LAB) for DSRC safety message. LAB consists of *broadcast manager*, *transmitter controller*, and *channel detector*. To adjust broadcast frequency, *broadcast manager* will analyse channel conditions of neighboring vehicles and adjust frequency correspondingly. If channel is congested, broadcast frequency will be reduced till it gets to lower bound. If channel is idle, broadcast frequency will increase till it gets to upper bound. Furthermore, channel condition is assessed by *channel detector* and LAB will broadcast its channel condition along with safety message. We conduct extensive simulations to evaluate the performance of LAB and simulation results validate efficiency of our proposed scheme.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This work is partly supported by National Key Research and Development Program grant (2016YFE0100600), NSFC (61672349, 61672353čň61472252), and China 973 Project (2014CB340303).

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] D. Serpanos and M. Wolf, "Industrial internet of things," in *Internet-of-Things (IoT) Systems*, pp. 37–54, Springer, 2018.

[3] X. Li, D. Li, J. Wan, A. V. Vasilakos, C.-F. Lai, and S. Wang, "A review of industrial wireless networks in the context of Industry 4.0," *Wireless Networks*, vol. 23, no. 1, pp. 23–41, 2017.

[4] Q. Liu, J. Wan, and K. Zhou, "Cloud manufacturing service system for industrial-cluster-oriented application," *Journal of Internet Technology*, vol. 15, no. 3, pp. 373–380, 2014.

[5] P. Mell and T. Grance, "The NIST definition of cloud computing," *Cloud Computing and Government: Background, Benefits, Risks*, pp. 171–173, 2011.

[6] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen, "A survey of energy efficient network protocols for wireless networks," *Wireless Networks*, vol. 7, no. 4, pp. 343–358, 2001.

[7] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*, Malaysia; Pearson Education Limited, 2016.

[8] L. Kong, M. K. Khan, F. Wu, G. Chen, and P. Zeng, "Millimeter-wave wireless communications for IoT-cloud supported autonomous vehicles: Overview, design, and challenges," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 62–68, 2017.

[9] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (VANETS): status, results, and challenges," *Telecommunication Systems*, vol. 50, no. 4, pp. 217–241, 2012.

[10] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

[11] F. Bai, D. D. Stancil, and H. Krishnan, "Toward understanding characteristics of dedicated short range communications (DSRC) from a perspective of vehicular network engineers," in *Proceedings of the 16th Annual Conference on Mobile Computing and Networking (MobiCom '10)*, pp. 329–340, ACM, September 2010.

[12] X. Chen, L. Kong, X. Liu, L. Rao, F. Bai, and Q. Xiang, "How cars talk louder, clearer and fairer: Optimizing the communication performance of connected vehicles via online synchronous control," in *Proceedings of the 35th Annual IEEE International Conference on Computer Communications, IEEE INFOCOM 2016*, USA, April 2016.

[13] G. Bansal, J. B. Kenney, and C. E. Rohrs, "LIMERIC: a linear adaptive message rate algorithm for DSRC congestion control," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4182–4197, 2013.

[14] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular Ad Hoc network," *Journal of Network and Computer Applications*, vol. 37, no. 1, pp. 380–392, 2014.

[15] F. Martelli, M. Elena Renda, G. Resta, and P. Santi, "A measurement-based study of beaconing performance in IEEE 802.11p vehicular networks," in *Proceedings of the IEEE Conference on Computer Communications, INFOCOM 2012*, pp. 1503–1511, USA, March 2012.

[16] Z. Tong, H. Lu, M. Haenggi, and C. Poellabauer, "A Stochastic Geometry Approach to the Modeling of DSRC for Vehicular Safety Communication," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 5, pp. 1448–1458, 2016.

[17] T. V. Nguyen, F. Baccelli, K. Zhu, S. Subramanian, and X. Wu, "A performance analysis of CSMA based broadcast protocol in

VANETs," in *Proceedings of the IEEE INFOCOM*, pp. 2805–2813, Turin, Italy, April 2013.

[18] F. Ye, R. Yim, J. Zhang, and S. Roy, "Congestion control to achieve optimal broadcast efficiency in VANETs," in *Proceedings of the 2010 IEEE International Conference on Communications, ICC 2010*, South Africa, May 2010.

[19] S. Djahel and Y. Ghamri-Doudane, "A robust congestion control scheme for fast and reliable dissemination of safety messages in VANETs," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '12)*, pp. 2264–2269, Shanghai, China, April 2012.

[20] H. Lv, X. Ye, L. An, and Y. Wang, "Distributed beacon frequency control algorithm for VANETs (DBFC)," in *Proceedings of the 2nd International Conference on Intelligent Systems Design and Engineering Applications (ISDEA '12)*, pp. 243–246, January 2012.

[21] M. S. Bouassida and M. Shawky, "On the congestion control within VANET," in *Proceedings of the 2008 1st IFIP Wireless Days, WD 2008*, UAE, November 2008.

[22] M. A. Benatia, L. Khoukhi, M. Esseghir, and L. M. Boulahia, "A markov chain based model for congestion control in VANETs," in *Proceedings of the 27th International Conference on Advanced Information Networking and Applications Workshops, WAINA 2013*, pp. 1021–1026, Spain, March 2013.

[23] B. M. Mughal, A. A. Wagan, and H. Hasbullah, "Efficient congestion control in VANET for safety messaging," in *Proceedings of the 2010 International Symposium on Information Technology, ITSim'10*, pp. 654–659, Malaysia, June 2010.

[24] G. Bansal and J. B. Kenney, "Achieving weighted-fairnessin message rate-based congestion control for DSRC systems," in *Proceedings of the IEEE 5th International Symposium on Wireless Vehicular Communications (WiVeC '13)*, pp. 1–5, Dresden, Germany, June 2013.

[25] T. Tielert, D. Jiang, Q. Chen, L. Delgrossi, and H. Hartenstein, "Design methodology and evaluation of rate adaptation based congestion control for vehicle safety communications," in *Proceedings of the IEEE Vehicular Networking Conference (VNC '11)*, pp. 116–123, November 2011.

[26] G. Bansal, H. Lu, J. B. Kenney, and C. Poellabauer, "EMBARC: Error model based adaptive rate control for vehicle-to-vehicle communications," in *Proceedings of the 10th ACM International Workshop on Vehicular Inter-Networking, Systems, and Applications, VANET 2013*, pp. 41–50, Taiwan, June 2013.

[27] "ns-3," https://www.nsnam.org/.

[28] "SUMO – Simulation of Urban MObility," http://sumo.dlr.de.

[29] Y. P. Fallah, H. ChingLing, S. Raja, and H. Krishnan, "Congestion control based on channel occupancy in vehicular broadcast networks," in *Proceedings of the IEEE 72nd Vehicular Technology Conference Fall (VTC-Fall '10)*, pp. 1–5, Ottawa, Canada, September 2010.

[30] J. Yin, T. Elbatt, G. Yeung et al., "Performance evaluation of safety applications over DSRC vehicular ad hoc networks," in *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks (VANET '04)*, pp. 1–9, October 2004.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 8589343, 13 pages
https://doi.org/10.1155/2018/8589343

Research Article

TTEthernet Transmission in Software-Defined Distributed Robot Intelligent Control System

Caibing Liu,[1] Fang Li ,[2] Guohao Chen,[1] and Xin Huang[3]

[1]*South China University of Technology, Guangzhou 510641, China*
[2]*School of Computer Science & Engineering, South China University of Technology, Guangzhou 510641, China*
[3]*Guangzhou Start to Sail Industrial Robot Co., Ltd., Guangzhou 510641, China*

Correspondence should be addressed to Fang Li; cslifang@scut.edu.cn

Received 11 April 2018; Accepted 12 June 2018; Published 17 July 2018

Academic Editor: Dongyao Jia

With the integration of new technologies such as smart technologies and cloud computing in the industrial Internet of Things, the complexity of industrial IoT applications is increasing. Real-time performance and determinism are becoming serious challenges for system implementation in these Internet of Things systems, especially in critical security areas. This paper provides a framework for a software-defined bus-based intelligent robot system and designs scheduling algorithms to make TTEthernet play the role of scheduling in the framework. Through the framework, the non-real-time and uncertainties problem of distributed robotic systems can be solved. Moreover, a fragment strategy was proposed to solve the problem of large delay caused by Rate-Constrained traffic. Experimental results indicate that the improved scheme based on fragmentation strategy proposed in this paper can improve the real-time performance of RC traffic to a certain extent. Besides, this paper made a performance test and comparison experiments of the improved scheme in the simulation software to verify the feasibility of the improved scheme. The result showed that the delay of Rate-Constrained traffic was reduced and the utilization rate of network was improved.

1. Introduction

Industrial Internet of Things (IoT) integrates heterogeneous networks such as the Internet, wireless sensor networks [1], and fieldbus networks in traditional industrial networks. With the integration of new technologies such as smart technologies and cloud computing in the industrial Internet of Things, the complexity of industrial IoT applications is increasing. The performance requirements of the system are also constantly increasing [2]. Especially in critical security areas, real-time performance and determinism are very important [3–5]. In terms of "real-time", real-time performance means that the control instruction can reach the control execution node at a specified time delay. In terms of "determination", the determinism refers to the change degree of the delay of the data packet transmitted by the control system, which is the jitter of the data packet [6].

Take a distributed intelligent robot system as an example. The design becomes complicated because the system needs distributed deployment and network integration. Besides, functional requirements are constantly increasing and the sensors and actuators are always from different manufacturers [7]. Also, open architecture is required in achieving the modern robot system for modularity, flexibility, reusability, and reconfigurability. Moreover, because the application of component-based development methods and software bus is implemented in the open architecture, real-time performance and determinism are becoming serious challenges for system implementation in these Internet of Things systems [8–11].

Many researchers have conducted many meaningful studies on real-time performance in IoT. Oliver et al. [12] investigated the computational complexity of incremental scheduling problems and profile performance metrics. Jielin et al. [13] proposed a multiobjective nondominated sorting genetic algorithm based on the mapping matrix. Fuchs [14] et al. used a mixed physical layer and the new real-time Ethernet standard without deteriorating timing in real-time. Danielis et al. [15] summarized the different industrial Ethernet protocols for real-time data transmission. Wollschlaeger et al. [16] investigated the working conditions

of time-sensitive Ethernet. Sestito et al. [17] optimized data extraction and classification of traffic-related features for real-time Ethernet anomaly detection. Carvajal et al. [18] explored the trade-offs of three different solutions for real-time communication. Fuchs et al. [19] proposed a test and online monitoring method for access delay, path delay, and synchronization quality.

Software-defined network (SDN) is an innovative architecture of Emulex network [20]. It is an implementation of network virtualization. Its core technology, OpenFlow, achieves separation of control and data flows. SDN enables flexible control of network traffic, making the network more intelligent as a conduit. SDN is an influential technology for the IoT [21]. Shu et al. [22, 23] described countermeasure techniques that can be used to prevent, mitigate, or recover from some of these attacks.

Increasing number of nodes increases the time delay and jitter of the transmitted data. The nodes of the system include bus nodes, robot nodes, and various device nodes. Specific nodes include depth cameras, gesture recognition devices [24], and recording devices. Especially when it comes to multirobot control, it does not meet the requirement of real-time performance of system from the perspective of task scheduling. Therefore, looking for a suitable real-time network to ensure the "hard" real-time performance of data transmission is particularly important [25–27].

The communication control buses used in traditional robot control systems, such as RS485 and CAN buses, have disadvantages like low transmission rates, short transmission distances, and poor real-time performance [28]. These disadvantages are performance bottlenecks in distributed systems. Real-time Ethernet overcomes the problems of low speed and short transmission distance of the traditional control bus. Moreover, it has certain real-time performance and reliability. It has quickly become one of the important control buses in the safety-critical field [29, 30].

TTEthernet, a new real-time Ethernet technology led by TTech company, has become more and more popular in recent years [31]. TTEthernet is a standard protocol that is compatible with legacy Ethernet and defines three types of data flow: time-triggered (TT), Rate-Constrained (RC), and Best-Effort (BE) Ethernet [32]. Through a global clock synchronization algorithm, a certain accuracy of the synchronization clock is maintained. Therefore, TT traffic achieves real-time performance and certainty to some extent. TTEthernet is based on the exchange of Ethernet technology, which can be extended to any topology and has high scalability and flexibility [33, 34]. Moreover, TTEthernet offers redundant channels and fault tolerance mechanisms that make it able to serve key security areas [35].

Tămaş-Selicean et al. [36] optimized the packaging of messages, the assignment of frames to virtual links, the routing of virtual links, and the time-triggered static schedule in frames so that all frames are schedulable and end-to-end delay of the RC messages is minimized. Aristova et al. [37] discussed Ethernet evolution history and packet assembly in the OSI model, analyzed the limitations of blocking Ethernet applications in industrial automation, and described innovative solutions such as power over Ethernet and embedded Ethernet switches. Selvatici et al. [38] suggested that there is a long-term transition for network. They provided an essential element for the coexistence of network, allowing for transparent communication from the network to another network. Yang et al. [39] proposed a universal control system security model for industrial real-time Ethernet interference. The added security module can effectively prevent hackers from malicious attacks and ensure the security of real-time Ethernet. Tamas-Selicean et al. [40] proposed an optimization algorithm to offline static schedule of TT messages. The result is that the deadline for TT and RC messages becomes more reasonable, and the end-to-end delay of RC traffic is minimized.

Adopting a service-oriented software-defined architecture offers the benefits of loose coupling and flexibility for the system, but at the same time the whole system is not in real-time [41]. To reduce this cost, this paper proposes time-triggered Ethernet as the system's transmission protocol to make the system real-time and deterministic. Time-triggered Ethernet overcomes the problem of low-rate, short transmission distance, and other issues of traditional control bus. Three kinds of traffic of Ethernet meet different real-time requirements of the applications. TT traffic application deployment can be used for high real-time demanding control tasks, BE traffic can be used for general non-real-time applications, and RC traffic is suitable for applications that have certain QoS requirements but do not demand real-time performance.

This paper provides an architecture for a bus-based software-defined intelligent robot system and designs scheduling algorithms to make TTEthernet play the role of scheduling in the architecture. Through the framework, the non-real-time and uncertainties problem of distributed robotic systems can be solved. Moreover, a fragment strategy was proposed to solve the problem that rate limits traffic and thus there was a large delay.

The remaining sections are as follows: Section 2 is TTEthernet-based distributed robot control system framework. Section 3 details the construction of distributed robot control system model based on TTEthernet. Section 4 introduces scheduling technology of the system. Section 5 details the experiments. Section 6 gives a conclusion of this paper.

2. Framework

Figure 1 shows the hierarchy architecture of the software-defined TTEthernet-based distributed robot control system, which intuitively presents the application of TTEthernet in the entire system framework. There are three layers in the framework, including service layer, bus layer, and bus management layer. Service layer is an integration of application modules. As a software-defined architecture, data and control are separated in the system. Collaborative control service coordinates the services of various sensors and provides data to the robot. There are three kinds of data services in this layer, including data acquisition service, data transmission service, and data processing service. The data service can collect and calculate information needed by the system, which can save computing resources and the hardware cost can be reduced

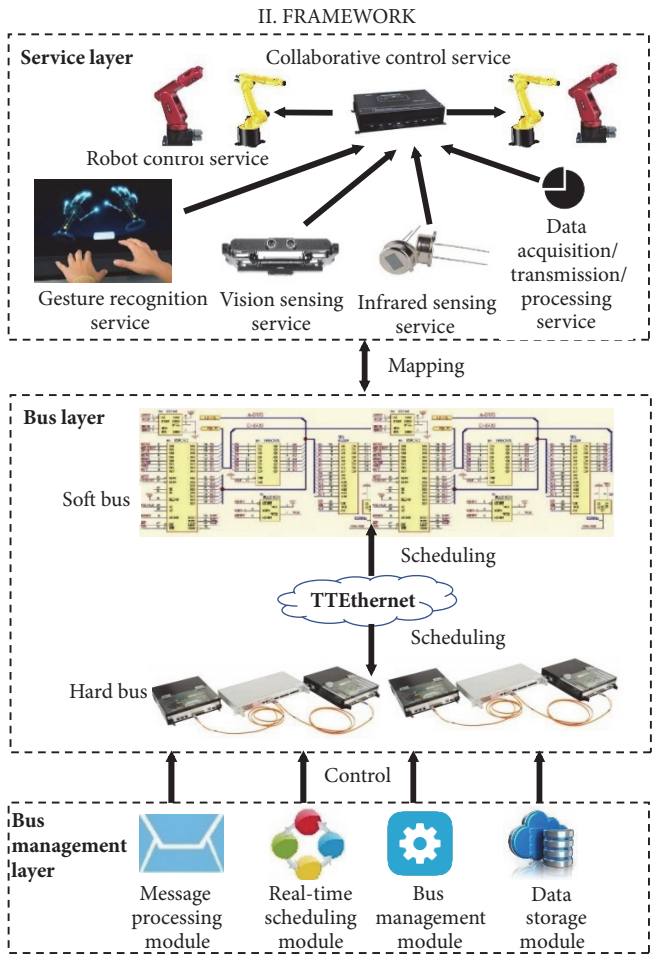


Figure 1: Architecture of software-defined distributed robot control.

accordingly. Therefore, software-defined data processing may provide a more flexible framework. Bus layer includes soft bus and hard bus. TTEthernet is very important because it schedules the bus resource. Bus management layer includes message processing module, real-time scheduling module, bus management module, and data storage module.

To prevent hotspot access from causing excessive load on the single-bus node, [42] proposed a distributed robot control system based on multibus nodes and used a load balancing algorithm to solve the problem of overload. Because of the service-oriented performance of the original control system, its loose coupling characteristics make the system expand into a multinode distributed system with advantages. The early built services can be accessed to the multibus node control system without any changes.

With the widespread use of various sensors in human-robot interaction scenes, the way people interact with robots becomes more and more abundant, such as voice interaction, gesture interaction, and force feedback interaction. A scene is set up based on hybrid sensors to control multiple robots. In the foreseeable future it can even be realized. Therefore, it is more and more important to study the distributed robot control system.

Although the introduction of multiple nodes (including bus nodes, robot nodes, and sensor nodes) has expanded the functions and application scope of the overall system, the introduction of multiple nodes in the system will cause the problem of degrading the real-time performance of message delivery. Because the original system adopts a TCP/UDP-based transmission protocol, the message transmitted by the system adopts a store-and-forward mechanism in the switch. For the high real-time control process, the forwarding of the message is likely to occur in the switch due to a long queue time. The time lag of the arrival of the control command at the control unit and the disadvantages of this type are fatal for the application in the safety-critical field.

Therefore, the proposed structure highlights the leading role of the message in the entire system.

An important feature of the improved service layer is that services can be combined with each other to build new task-solving processes. Service composition can solve the problem of low development efficiency and high repetition rate. Developers do not repeat coding; they just need to combine the existing service modules. Through this control system, both the old robot application and the new application can seamlessly interact with each other on this platform. Software

Table 1: Service description document structure.

Element	Definition
<portType>	Services related operations, implementation functions
<message>	The message used by the service
<types>	The data type used by the service
<binding>	Message bus and protocol bounded to the Service
<QoS>	The quality of service provided by the service

developed in different periods is available in this platform.

Table 1 describes the structure of the service description document. <types> defines the data type that service will use, whereas data types are defined with XML Schema. In doing so, XML is platform independent. <message> defines the data elements of an operation; each message can be composed of one or more components. The components in the message can be seen as the formal parameters required to write a function in the procedure. <binding> defines the protocol and format details used to provide the service. <portType> describes the operations that the service can perform and details about the parameters and messages involved. <QoS> describes the service quality of a service.

The bus layer provides various message encapsulations to the service layer. Two layers have mappings. Application services send a specific message to the bus according to their own needs. Google protocol buffers are a protocol defined by Google's internal messaging. Because the message is platform independent and based on binary, it is ideal for platforms that require high performance. According to the published test results, Google protocol buffers are 3 to 10 times smaller than traditional XML serialized messages, with 20-100 times better performance than XML on serialization and deserialization. These performance improvements are achieved with the cost of readability like the XML format. However, for the proposed system, it is worth sacrificing some readability for better performance.

Each message consists of three parts: the length of the header, the body of the header, and the body of the message. The length of the header is a 4-byte digital representation of the unsigned integer, which is read as small-end storage when fetched. The body of the header indicates the description attribute provided by the message packet. Among headers, in addition to the keyword "type" and "msgLength", the rest of the keywords are not necessary. Relevant fields are filled according to the role of the message. According to the service registration process, firstly, the service sends a message to the bus whose type is SERVICE_REGIESTER, and then the bus returns the registration service result with the message type SERVICE_RESULT.

The bus layer requires maintaining communications between the software bus and TTEthernet and regular communications between TTEthernet and hard bus. Finally, the service layer can greatly reduce the complexity of the development of the upper application, enabling service providers to focus on writing service-specific function codes and dramatically increase production efficiency.

Since the bus management layer of the prototype system requires frequent calls to the data storage module, the bus management layer and data storage module of the prototype system should be integrated in consideration of the improvement of overall system efficiency. Because each bus independently assumes all the functions in the distributed bus control system architecture, each bus node has a corresponding data storage module. Data redundancy improves, which also improves the data safety. The bus control layer of the prototype system needs to make frequent calls to the data storage layer for the registration information of the service, the routing information of the message, and the management information of the bus. So consider the efficiency of the overall system and integrate the bus control layer and data storage layer of the prototype system. And because, in the distributed bus control system structure, each bus independently assumes all the functions, each bus node has a corresponding data storage module equivalent to data for redundant backup, which also improves data security.

3. Model Building

3.1. Model Complementation. Figure 2 is a schematic diagram of the TTEthernet terminal node simulation model. It can be seen from the figure that the terminal node simulation model mainly includes four parts: a traffic application part, a sending part, a receiving part, and a scheduling and time synchronization part.

Figure 3 shows the relationship of the scheduler, sender, and receiver. The scheduling and time synchronization part is composed of the scheduling trigger module and the time synchronization module. System scheduling trigger module mainly depends on the scheduler. In scheduler part, the scheduler mainly consists of three modules: timer, oscillator, and period. The timer module is the core module of the scheduler. Timer maintains an event queue, which is used to trigger other events in the terminal node related to time modules (such as TTApp). Event triggered on the terminal node needs to be registered in the timer, like the TT sending queue, receiving queue, and so on. Oscillator is a local clock module that provides a time reference for the timer. Period provides the number of periods to the timer. That is, if the next period arrives (TT period), it will send a message to the timer to update the period. The timer will update the parameter according to this message and handle it accordingly (for example, discard the events of the previous period and issue a false warning).

The time synchronization module consists of a Sync module. Sync forwards the clock synchronization signal from the master node to the timer and forwards the clock synchronization signal to the synchronization master node during the synchronization period.

The processing of the timer is as follows:

(1) Initialize the event list in timer module, oscillator module, and period module, and get the global clock signal.

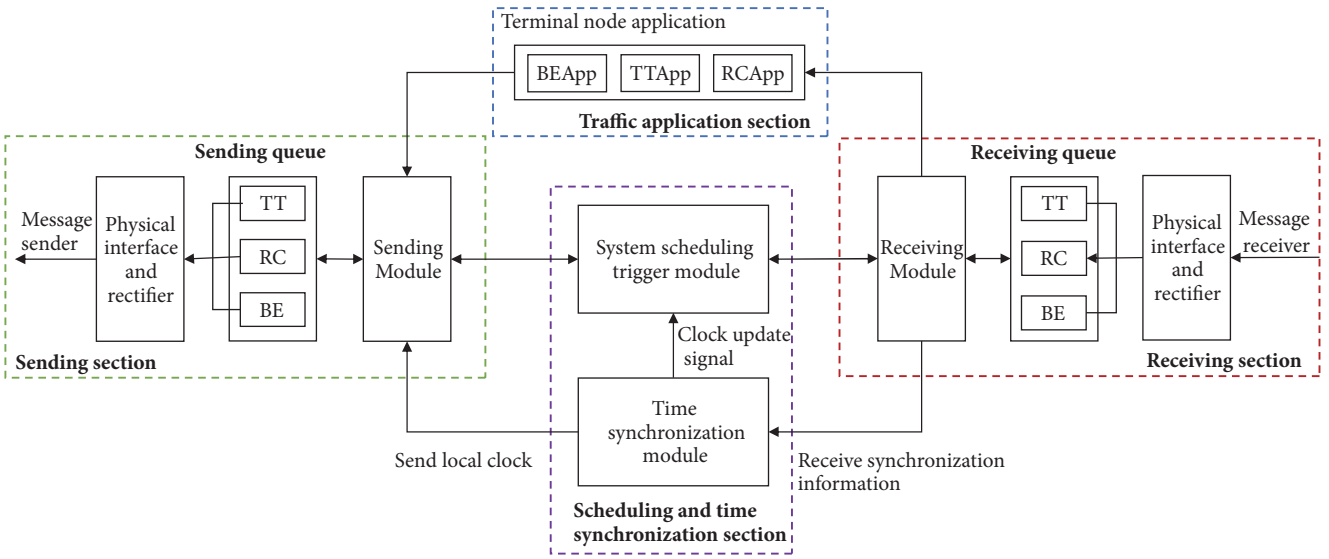


Figure 2: TTEthernet terminal node simulation model.

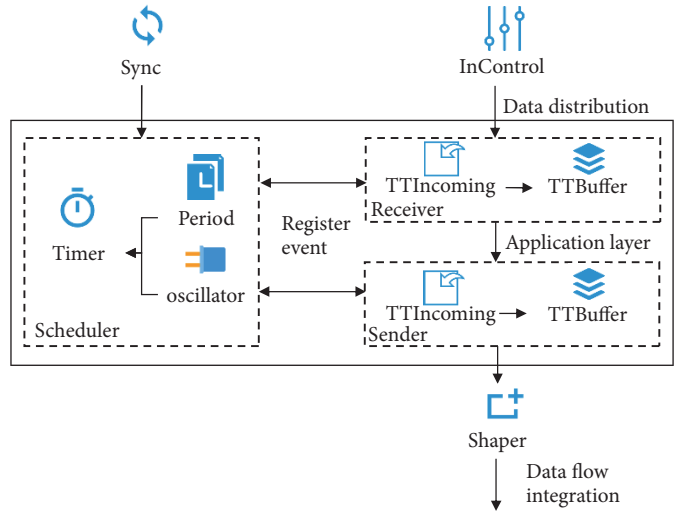


Figure 3: Relationship of scheduler, sender, and receiver.

(2) Monitor trigger message cyclically; if trigger message is received, go to step C. Otherwise, continue monitoring.

(3) Start to search the event list and send the event trigger message with the trigger time shorter than or equal to the current time to the corresponding module receiving unit. If the trigger time is less than the current time, output the event alarm signal. Otherwise, go to (2) to continue monitoring.

Take TT message as an example; in the receiver part, InControl module is used for receiving data, and the main function is to complete the diversion of different data messages. It is in the data link layer in the physical receiving layer. As can be seen in the figure; InControl sends the received TT data to the receiver that handles the TT data.

Receiver has two modules, TTIncoming and TTBuffer. The main function of TTIncoming is to ensure that the TT message can be received within the time range specified in the system schedule. If it exceeds the received time range, the message needs to be discarded and returns the alarm. Otherwise, TTIncoming sends an event to the scheduler. The event is that TT messages are forwarded to TTBuffer based on the time specified in the system's schedule. Scheduler triggers TTIncoming to send a message to the TTBuffer queue based on the time of the registration event. TTBuffer cyclically registers sending events in the scheduler according to the configuration cycle from initialization. Finally, when the TTBuffer sending event is triggered, the scheduler triggers the TTBuffer to send the TT message from the queue to the application layer.

The processing of receiver is as follows:

(1) Initialize TTIncoming and TTBuffer, and then TTBuffer registers sending trigger events according to the system's time schedule to the scheduler.

(2) TTIncoming monitors the receiving port, and if the received message is within the allowed time range, send a trigger event to the scheduler registration and go to c. Otherwise, discard the message and alarm, and then continue monitoring the port.

(3) Scheduler sends a trigger message to TTIncoming; TTIncoming forwards the received message to TTBuffer and goes to (4).

(4) TTBuffer puts the message forwarded by TTIncoming to the message queue. If TTBuffer receives the trigger message sent from the scheduler, it pops the first node in the message queue and forwards it to the corresponding APP according to the configuration file parameters. If no trigger message is received, then wait for the next scheduler trigger message to arrive.

In sending part, the functions of TTIncoming and TTBuffer are the same as those described in the receiver. Shaper module is in the link layer to deal with different data streams module. The main function of Shaper module is to integrate the three data streams and forward the data to the physical link without causing conflict according to the priority of the data stream.

The processing of the sending module is as follows:

(1) Initialize TTIncoming and TTBuffer, and TTBuffer registers sending trigger events according to the system's time schedule to the scheduler.

(2) TTIncoming monitors the received port from the application layer and registers a sending event to the scheduler if the received message is within the allowed time range and goes to (3). Otherwise, discard the message and alarm, and continue monitoring the port.

(4) Scheduler sends a trigger message to TTIncoming, and TTIncoming forwards the received message to TTBuffer and goes to (4).

(5) TTBuffer puts the message forwarded by TTIncoming into the message queue. If the TTBuffer receives the trigger message from the scheduler, then it pops the message in the message queue and forwards it to the Shaper module in the link layer and goes to (5). If no trigger message is received, then wait for the next scheduler trigger message to arrive.

(6) According to the priorities in the queue, the Shaper module forwards the messages in the queue to the physical ports from a high priority to a low priority until the message queue is empty.

As shown in Figure 4, the RC traffic application needs to periodically schedule its scheduler registration to the system because the RC traffic needs to periodically send RC traffic data according to the RC parameters (sending interval time).

The processing of RC traffic application is as follows:

(1) Initialize the application based on the system network configuration, set the rate of the RC traffic and the maximum size of the sent data packet, and register the initial triggering event in the scheduler.

(2) The news arrives. If the event is scheduler-triggered, go to (4). Otherwise, go to (3).

(3) Discard the message, go to b and wait for the next message to arrive.

(4) Generate eligible data packets according to the configuration parameters.

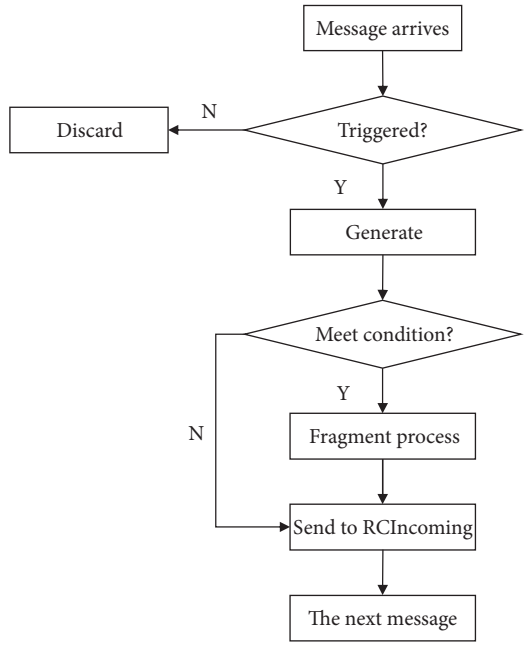


Figure 4: Process RC application.

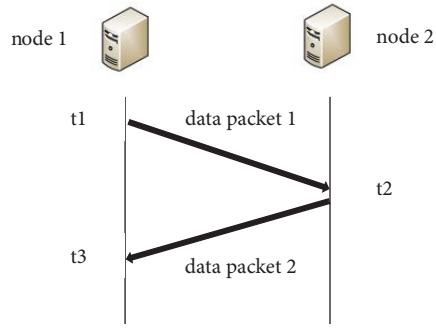


Figure 5: Network end-to-end time delay.

(5) Determine whether the current sending condition meets the condition of the fragmentation. If the condition is satisfied, go to (6). Otherwise, go to (7).

(6) Fragment the message to be sent according to the fragmentation policy.

(7) Send all messages in the buffer to RCIncoming in the sending module.

(8) According to the parameters of the rate limit, register a new trigger message in the scheduler, return to (2), and wait for the next message to arrive.

3.2. Performance Index. The network performance indicators in the experiment are as follows:

(1) Time delay: it is the time from the data sending the node to the data receiving node. The experimental time delay used in this paper is the one-way transmission delay from the sending node to the receiving node. As shown in Figure 5, the end-to-end time delay of packet 1 is: $Delay_{12} = t_2 - t_1$. The delay of packet 2 is $Delay_{21} = t_3 - t_2$.

(2) Network jitter: it is the transmission delay difference between the largest packet and the smallest packet over a measurement interval. The factors that affect jitter are generally related to network congestion. In simulation experiments, the closer T to zero, the smaller the network jitter and the more stable the overall network operation. *Tjitter* is calculated as follows: *Tjitter=Tdelay- TlastDelay*, where *Tdelay* is the current data stream transmission delay, and *TlastDelay* is the last data stream transmission delay.

(3) Network channel utilization: it includes the proportion of the total time that the channel is sent to the overall channel. It can be obtained through the OMNeT ++ API.

(4) Network packet loss: it is the loss of data units that cannot reach the receiving node in the transmission process. It is caused by the physical nature of physical links or network-driven defects.

4. Scheduling Technology

Due to the large RC data packet, the transmission cannot be completed before the arrival of the next TT data packet. The timely blocking algorithm is triggered and the RC data packet will delay sending. Because TT data traffic occupies data link bandwidth periodically, it is much likely that an RC data packet is clogged in the node's buffer zone and cannot be sent. In this case, if an RC traffic burst occurs in an application sending the RC packet, a large part of the traffic burst will be discarded by the buffer due to the size limitation of the buffer, resulting in a packet loss problem. To solve this problem, this paper presents a fixed size RC packet segmentation strategy. Figure 6 shows the flow chart of the fragment strategy algorithm. A specific description of a fixed size RC packet segmentation policy is as follows:

(1) Determine whether the packet is greater than the maximum transmission unit virtual link limit when RC data packet arrives. If the packet is bigger than the maximum transmission unit limit, the packet is divided into several data packets by the maximum transmission unit.

(2) Add all groups to the preparation queue. If the preparation queue is empty, then the process is over, or jump to the next step.

(3) Take a packet from preparation queue and obtain the time *Ttt_arrival* next TT data stream arrives. Calculate packet sending end time, and the formula is as follows:

$$\begin{aligned} &SendDuration \\ &= \frac{(msize + INTERFRAME + ((PREAMBLE + SFD) * 8))}{txRate} \\ &SendTime = TotalTime + SendDuration \end{aligned} \quad (1)$$

where *SendDuration* is the time required to transmit the packet, *SendTime* is the end of the packet transmission, *msize* is the number of bits for sending packets. INTERFRAME, PREAMBEL, and SFD are the required fields for the network interface layer whose sizes are 96 bits, 56 bits, and 8 bits, respectively. *TotalTime* is the system's current time.

(4) Determine if *Ttt_arrival* is greater than *SendDuration*; if it is, then the packet can be directly sent to the RC packet buffer queue. If not, go to the fifth step.

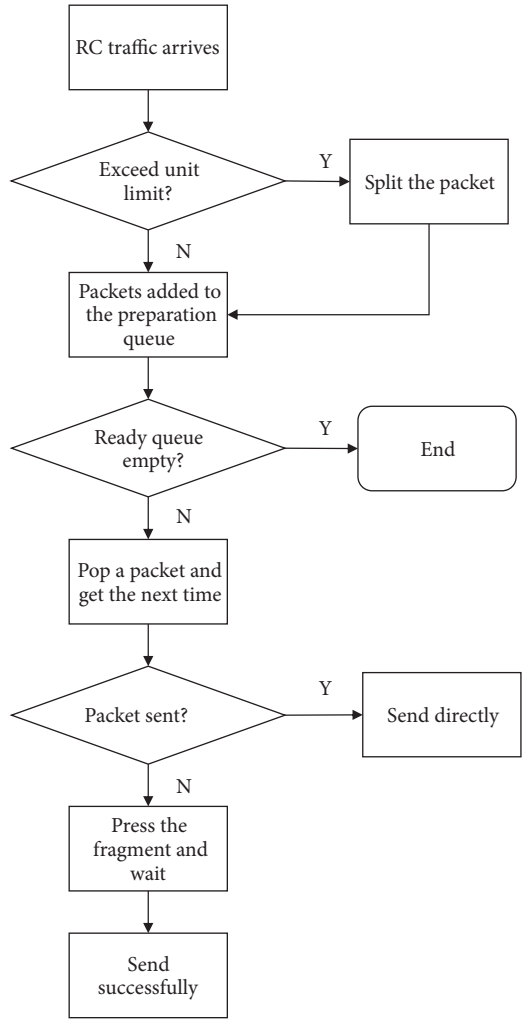


Figure 6: Process of fragment strategy algorithm.

(5) Split the packet. The current link can be used to transfer the largest unit as the basis of the fragment to avoid the problem of information redundancy.

$$\begin{aligned} &MAXUNIT \\ &= \left(TotalTime - T_{tt\_arrival} - T_{safety\_margin}\right) * txRate \\ &N = \left\lceil \frac{msize}{MAXUNIT} \right\rceil \end{aligned} \quad (2)$$

MAXUNIT is the maximum size of fragments that can be transmitted, and N is the number of fragments that are grouped. *Tsafety_margin* is used to guarantee that the fragment can complete the transmission.

(6) Divide the packet into N fragments according to the formula in the fifth step and place it into the buffer queue to wait for the transmission.

(7) After all the fragments have been sent, the execution process of a fixed size RC packet will go to the second step.

Table 2: TT traffic application.

Send node	Application type	Destination	Period (ms)	Traffic size (byte)
co_controller	TTSource	Robot 1	100	72
co_controller	TTSource	Robot 2	100	72
co_controller	TTSource	Robot 3	100	72
co_controller	TTSource	Robot 4	100	72
sensor 3	TTSource	co_controller	50	80

Table 3: RC traffic application.

Send node	Application type	Destination	Period (ms)	Bandwidth distribution slots (us)	Traffic size (byte)
Robot 1	RCSource	co_controller	0.8	350	390
Robot 2	RCSource	co_controller	0.8	350	390
Robot 3	RCSource	co_controller	0.8	350	390
Robot 4	RCSource	co_controller	0.8	350	390
Sensor 1	RCSource	co_controller	0.8	350	390
Sensor 2	RCSource	co_controller	0.8	350	390
Robot 3	RCSource	co_controller	0.8	350	390
Robot 4	RCSource	co_controller	0.8	350	390

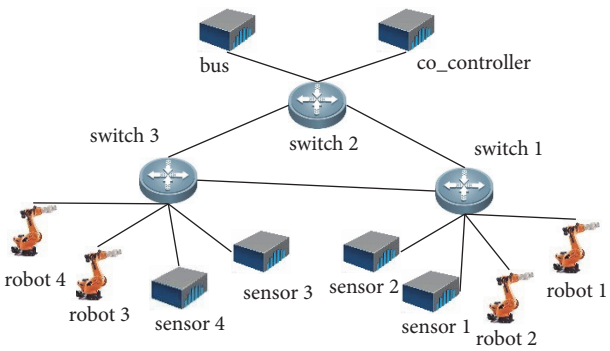


Figure 7: Simulation experiment network topology.

5. Experiment

5.1. System Performance Verification Experiment. The first experiment is a TTEthernet distributed robot system simulation verification. The main purpose is to verify whether TTEthernet-based distributed robot system can meet the real-time and deterministic requirements. The simulation topology is shown in Figure 7. There were four robots, sensors, cocontrollers, and bus nodes used to represent the TTEthernet host node. Robot 1, robot 2, sensor 1, and sensor 2 are connected to switch 1. Similarly, robot 3, robot 4, sensor 3, and sensor 4 are connected to switch 3. Bus nodes and cooperative controller are connected to switch 2. Three TTEthernet switches form the core switching network. All channels have a link length of 100Mb/s and a length of 20 meters. Four time-triggered applications are deployed in the coordinated controller. One control instruction is sent to all the four robots. That is, a total of four TT data streams are sent from the coordinated controller. The time spent from the origin device to the destination device is 20us. The entire network has a cluster period of 100ms. The switch schedules the received messages in 100us. The size of the time-triggered data is 72 Bytes. In addition, sensor 3 deployed a time-triggered traffic and was sent to the cooperative controller. The switch transmits the traffic in 200us, which sends the monitored operating status of the whole system to the coordinated controller. The detailed TT traffic application is shown in Table 2.

In addition to the TT message sending time in the cluster cycle, the remaining time channels can transmit RC messages and normal Ethernet messages. The rate limit message has higher priority than the normal Ethernet message if not in the time-triggered period; the switch will forward the rate limit message with priority. Therefore, in this experiment, a rate limiting application was deployed in sensor 1-sensor 4. The allocated bandwidth is 350us, the traffic is 390 Bytes, and message generation interval is 0.8ms. In addition, four robots are also deployed with a RC traffic, the bandwidth is 350us, the message size is 390 Bytes, and message generation interval is 0.8ms. Detailed RC traffic applications are shown in Table 3.

The four robots nodes deployed an application that provide BE service. The sending data time interval of this application is in the range of 100μs and 500μs obeying uniform distribution. The data size obeys uniform distribution in the range of 46 Bytes and 1500 Bytes. Detailed RC traffic applications are shown in Table 4.

The second experiment mainly verifies real-time performance and other related indicators about RC traffic fragmentation strategy, including time delay, jitter, network packet

Table 4: BG traffic application.

Send node	Application type	Destin-nation	Period (ms)	Traffic size (byte)
Robot 1	BGSource	bus	uniform(100,500)	uniform(46, 1500)
Robot 2	BGSource	bus	uniform(100,500)	uniform(46, 1500)
Robot 3	BGSource	bus	uniform(100,500)	uniform(46, 1500)
Robot 4	BGSource	bus	uniform(100,500)	uniform(46, 1500)
Robot 1	BGSource	Robot 2	uniform(100,500)	uniform(46, 1500)
Robot 2	BGSource	Robot 3	uniform(100,500)	uniform(46, 1500)
Robot 3	BGSource	Robot 4	uniform(100,500)	uniform(46, 1500)
Robot 4	BGSource	Robot 1	uniform(100,500)	uniform(46, 1500)

Table 5: Time delays and average jitter of three types of traffic.

Time delay (us)	TT traffic (200us)	TT traffic (100us)	RC traffic	BE traffic
max time delay	213.827	109.529	405.009	109593
min time delay	195.182	108.538	100.000	19.700
average time delay	201.205	109.037	142.004	797.909
average jitter	3.153	0.231	29.162	118.035

loss rate, and data link utilization. The fragmentation strategy was adapted based on the TTEthernet simulation model at HAW Hamburg University in Germany [43]. This experiment mainly compares two fragmentation strategies and the non-fragmentation strategy under the application environment of three different RC sending data sizes.

The time distribution of the time-triggered traffic is mainly in the interval of 100us to 200us, and the delay distribution of RC traffic is mainly distributed in the interval of 200 to 600, while the delay distribution of BE traffic is dispersed over the entire time axis. In conclusion, the time-triggered traffic has the advantage of low average latency over the remaining two traffic types, followed by RC traffic, and the BE traffic is the worst.

Table 5 shows the statistics of the maximum delay and the minimum delay for three types of traffic. The jitter time is the time delay difference of two consecutive data packets, reflecting the stability and certainty of the transmission of the different traffic data streams. The time-triggered traffic (100us) is the time-triggered traffic from the coordinated controller to the robot node. The average delay of triggered traffic is 109.037us with the average jitter of 0.231us, which is the least jitter among the three traffic types that meets the requirements of the motion control. The requirements are that the maximum delay is not higher than 1ms and the maximum time certainty is not higher than 1us [6]. Moreover, as the monitoring signal, the time-triggered traffic is 200us; the average delay is 201.205us with the average jitter of 3.153us, which follows the control, and has met the time certainty of monitoring. Due to TTEthernet's regulation, the average delay and average jitter of RC traffic and BE traffic are lower than the time-triggered traffic. RC traffic can transmit video and other data that is not deterministic to the data flow. Transfer bus service management data.

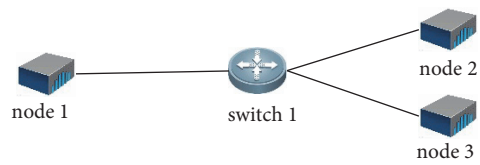


Figure 8: Simulation experiment network topology.

Through the analysis of the experimental data, it can be verified that the TTEthernet distributed robot control system meets the robot control requirements. From the above data, it can be obtained that, in time-triggered Ethernet simulation, time-triggered traffic has an advantage of lower latency and higher certainty. Time-trigger traffic is suitable for real-time and deterministic data. The BE data can be used to transfer application data with low requirements of real-time performance. RC traffic is a compromise between time-triggered and BE data.

5.2. Fragmentation Strategy Verification Experiment. The second experiment, the fragmentation strategy verification experiment, mainly verifies the relevant indicators such as real-time performance of the RC traffic fragment strategy, including time delay, jitter, network packet loss rate, and data link utilization. Figure 8 shows the topological structure of the simulation application, which contains three end nodes and one switch node. Each terminal node is connected with the switch node; the data link data transfer rate is 100Mbits/s. The simulation time of the experiment is 10s. In this experiment, two TT cycles, C1 and C2, are defined, the period of C1 is 100us, the period of C2 is 1000us, and there is a bias of 30us (that is, every C2 period starts from ($i * 1000 + 30$, $i = 2...$) us). If the least common multiple of C1 and C2 as a

Table 6: TT traffic application.

Terminal node	Application type	Destination	Period	Traffic size (Byte)
Node 1	TTSource	Nodes 2, 3	C_1	46
Node 1	TTSource	Node 2	C_2	46
Node 2	TTSource	Node 1	C_2	46

Table 7: Comparison of three environmental parameters.

scene	Application type	Traffic size (Byte)	SI(us)	VL(us)
1	RCSource/RCSource_frag	Uniform(46,357)	5000	1000
2	RCSource/RCSource_frag	Uniform(357,1232)	5000	1000
3	RCSource/RCSource_frag	Uniform(1232,1500)	5000	1000

SI: sending interval; VL: virtual link.

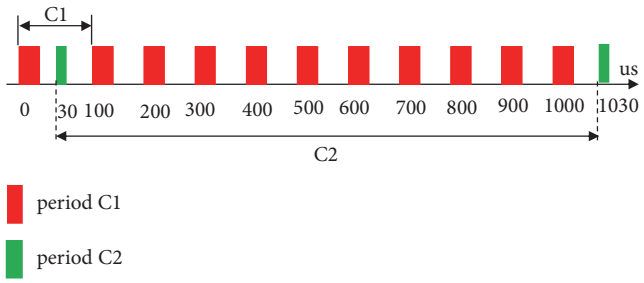


Figure 9: Network TT cycle integration.

Table 8: Average time delay of RC traffic in three scenarios.

Average time delay (s)	scene 1	scene 2	scene 3
RCSource	4.73288E-05	#NaN	#NaN
RCSource_frag	4.52014E-05	0.000354399	0.00062752

complete period from the perspective of a data link is taken, the TT traffic occupancy in the data link can be obtained as shown in the figure. In addition, the configuration file also defines that the length of buffer queues should be no longer than 2500. For bursts of large data streams, it is likely that due to the limitation of buffer queue size a higher packet loss rate will result.

Figure 9 shows the result of integrating two TT cycles into the whole network. It can be noticed that the smallest interval in the network occurs in the interval of 1000us increments (0us, 30us and 1000us, 1030us). If the data size transmitted in the data link of the TT cycle is the smallest data unit (46 Bytes), i.e., the smallest Ethernet frame (64 Bytes), the maximum data frame transmitted in this interval is about 375 Bytes in size (which should be less than 375 Bytes in actuality). Similarly, in the interval (100us, 200us) and other increments of 1000us, the maximum size of the data is about 1250 Bytes (actually less than 1250 Bytes).

Table 6 describes the application of TT traffic deployed in the terminal node in this lab. TT data stream in the distributed robot control system mainly controls signals such as real-time performance and high deterministic signal.

This experiment mainly verifies the performance of RC traffic subcontracting strategy, so this paper compares the performance of the three applications. The application of the RC traffic is deployed on Node 2, and the RCSource and RCSource_frag correspond to the RC traffic application without fragmentation strategy and with fragmentation strategy, respectively. In addition to the two applications with or without fragmentation strategy, the other parameters are the same. Table 7 lists the three application scenarios for comparison. The main difference is that the traffic is differentiated according to the load. The other parameters are the same. The traffic volume of the three environments follows a uniform distribution. The parameters are designed according to the maximum and minimum transmittable data frame sizes in the integrated TT cycle mentioned above.

Table 8 shows the average delay of RC traffic in three scenarios. It can be seen from the data that in all three scenarios the average delay of RC traffic with fragmentation strategy is better than that of the original model. It can be seen from the simulation results that the maximum experimental latency with fragmentation strategy under scene 1 is 107.37us, the minimum is 19.59us, and the difference is 87.78. The maximum experimental delay without fragmentation strategy was 135.03us, the minimum was 19.59us, and the difference was 115.44. In scene 1, from the perspective of average delay, it can be concluded that the performance of the fragmented strategy is improved by 4.5% compared with the nonfragmented strategy.

The appearance of #NaN in the table indicates that the simulation platform does not detect the related traffic; that is, in scene 2 and scene 3, the RC application data stream without the fragment strategy fails to reach the target node. It can be found in the log file through the simulation platform that, as the data generated by the RC application at the beginning of the TT period gap cannot be successfully sent (i.e., greater than the maximum transmittable data frame size, about 1250 Bytes), the data frame has been clogged in buffer queue. Consequently, the following data frames cannot be sent.

From the data of scene 2 and scene 3, it can be found that the RC application of the original model in TTEthernet is not robust to the larger RC data frames, resulting in a waste of

Table 9: Average jitter of RC traffic in three scenarios.

Average time delay (s)	scene 1	scene 2	scene 3
RCSource	2.08913E-05	#NaN	#NaN
RCSource_frag	1.8761E-05	0.000307254	0.000649226

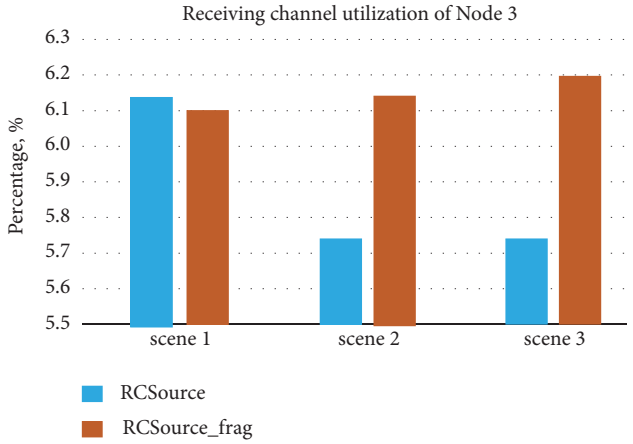


Figure 10: Receiving channel utilization of Node 3.

data link bandwidth. This is also the reason for the proposed RC data fragmentation strategy.

Table 9 shows the average jitter of RC traffic in three scenarios. It can be seen from the data that, in all three scenarios, the average jitter time of the RC traffic application with fragmentation strategy is due to the RC application of the original model. In scene 1, the average RC jitter of the original model is 20.89μs, that of the RC application with fragmentation strategy is 18.76μs, and that of the RC model with fragmentation strategy is 10.2% higher than the jitter performance of the original model. Similarly, due to the problem of congestion in the buffer queue, the RC traffic without the original model RC application is successfully received.

The utilization of Node 3 receiving channel in the simulation network is shown in Figure 10. Node 3 is the only node that receives the RC traffic in the simulation network. The channel utilization of original model and improved model can be analyzed by examining the receiving channel utilization of the node. It can be obtained from scene 1 that the channel utilization of the original model is 6.12% and the channel utilization of the improved model is 6.10%, which is considered to be consistent. In scene 2 and scene 3, the channel utilization rate of the original model dropped sharply to 5.76% due to the failure that sends the RC traffic. In scenarios 2 and 3, the channel utilization of the improved model is 6.12% and 6.20%, respectively. The reason for the higher channel utilization in scene 3 lies in the fact that the size of the data to be sent is concentrated on larger data frames (1232 Bytes, 1500 Bytes). Moreover, due to the use of substrategy, channel utilization is improved.

Table 10: Packet loss in simulation network in three scenarios.

Packet loss	scene 1	scene 2	scene 3
RCSource	0	0	0
RCSource_frag	0	0	662

Finally, we acquire the packet loss by examining the simulation network, which is shown in Table 10. For scene 1 and scene 2, the result indicates that both models have zero packet loss. However, in scene 3, packet loss occurs in applications which use the fragment strategy, accounting for 16.02% of the total data frames sent. The reason for packet loss is that a large data frame is divided into several data frames due to the fragment strategy. If the buffer queue length is short, data packet loss occurs. Therefore, the use of fragment strategies needs to consider the size of the buffer and transmit the data with fewer certainty requirements.

6. Conclusion

This paper provides a framework for a bus-based intelligent robot system and designs scheduling algorithms to make TTEthernet play the role of scheduling in the framework. TTEthernet is proposed as the system's transmission protocol which has improved the real-time and deterministic performance in the system. TTEthernet overcomes the problems of low speed and short transmission distances and has certain real-time and reliability performance.

A bus-based intelligent robot system framework is put forward and a scheduling algorithm is designed. The framework can achieve real-time performance and deterministic performance for distributed robot systems. In addition, because of the problem of rate limiting traffic which causes a large delay, a fragment strategy is proposed to solve this problem.

Experimental results indicate that the improved scheme based on fragmentation strategy proposed in this paper can improve the real-time performance of RC traffic to a certain extent but due to network node buffer queue restrictions will lead to packet loss of fragmentation strategy when there is a large data flow of the problem; the application of certainty is not applicable. In addition, the original simulation model is not optimized for the virtual link routing. For the simulation of large networks, there can be an improvement in the TTEthernet virtual link routing problem. These issues need further analysis and resolution.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is partially supported by National Natural Science Foundation of China (no. 61602182), Science and Technology

Planning Project of Guangzhou, China (no. 201604046029), Guangdong Natural Science Funds for Distinguished Young Scholar (2017A030306015), Pearl River S&T Nova Program of Guangzhou (201710010059), Guangdong Special Projects (2016TQ03X824), and the Fundamental Research Funds for the Central Universities (no. 2017JQ009).

References

[1] X. Li, D. Li, J. Wan, A. V. Vasilakos, C.-F. Lai, and S. Wang, "A review of industrial wireless networks in the context of Industry 4.0," *Wireless Networks*, vol. 23, no. 1, pp. 23–41, 2017.

[2] G. Du, P. Zhang, and D. Li, "Human-manipulator interface based on multisensory process via kalman filters," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 10, pp. 5411–5418, 2014.

[3] M. FRUSTACI, P. PACE, G. ALOI, and G. FORTINO, "Evaluating critical security issues of the IoT world: Present and Future challenges," *IEEE Internet of Things Journal*, 2017.

[4] M. Asplund and S. Nadjm-Tehrani, "Attitudes and Perceptions of IoT Security in Critical Societal Services," *IEEE Access*, vol. 4, pp. 2130–2138, 2016.

[5] H. B. Salameh, S. Almajali, M. Ayyash, and H. Elgala, "Security-aware channel assignment in IoT-based cognitive radio networks for time-critical applications," in *Proceedings of the 4th International Conference on Software Defined Systems, SDS 2017*, pp. 43–47, Spain, May 2017.

[6] M. Felser, "Real-time ethernet - Industry prospective," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1118–1129, 2005.

[7] G. Du and P. Zhang, "Online serial manipulator calibration based on multisensory process via extended kalman and particle filters," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 12, pp. 6852–6859, 2014.

[8] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *Journal of the ACM*, vol. 20, no. 1, pp. 46–61, 1973.

[9] M. S. Branicky, S. M. Phillips, and W. Zhang, "Scheduling and feedback co-design for networked control systems," in *Proceedings of the 41st IEEE Conference on Decision and Control*, pp. 1211–1217, Las Vegas, Nev, USA, December 2002.

[10] S. Dai, H. Lin, and S. S. Ge, "Scheduling-and-control codesign for a collection of networked control systems with uncertain delays," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 1, pp. 66–78, 2010.

[11] W. Jie and L. Wei-dong, "Control and scheduling co-design of networked control system," in *Proceedings of the 2011 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, pp. 1–4, Xi'an, Shaanxi, China, September 2011.

[12] R. Serna Oliver, S. S. Craciunas, and G. Stoger, "Analysis of Deterministic Ethernet scheduling for the Industrial Internet of Things," in *Proceedings of the 2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, CAMAD 2014*, pp. 320–324, Greece, December 2014.

[13] J. Li and M. Chen, "Multiobjective Topology Optimization Based on Mapping Matrix and NSGA-II for Switched Industrial Internet of Things," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1235–1245, 2016.

[14] S. Fuchs, H.-P. Schmidt, and S. Witte, "Test and on-line monitoring of real-time Ethernet with mixed pysical layer for Industry 4.0," in *Proceedings of the 21st IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2016*, Germany, September 2016.

[15] P. Danielis, J. Skodzik, V. Altmann et al., "Survey on real-time communication via ethernet in industrial automation environments," in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, pp. 1–8, Barcelona, Spain, September 2014.

[16] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, 2017.

[17] G. S. Sestito, A. C. Turcato, A. L. Dias et al., "A Method for Anomalies Detection in Real-Time Ethernet Data Traffic Applied to PROFINET," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 2171–2180, 2018.

[18] G. Carvajal, C. W. Wu, and S. Fischmeister, "Evaluation of communication architectures for switched real-time ethernet," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 63, no. 1, pp. 218–229, 2014.

[19] S. Fuchs, A. Gercikow, and H. Schmidt, "Monitoring of real-time behavior of industrial ethernet for industry 4.0," in *Proceedings of the 2017 International Electrical Engineering Congress (iEECON)*, pp. 1–4, Pattaya, Thailand, March 2017.

[20] X. Li, D. Li, J. Wan, C. Liu, and M. Imran, "Adaptive Transmission Optimization in SDN-Based Industrial Internet of Things With Edge Computing," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1351–1360, 2018.

[21] J. Wan, S. Tang, Z. Shu et al., "Software-Defined Industrial Internet of Things in the Context of Industry 4.0," *IEEE Sensors Journal*, vol. 16, no. 20, pp. 7373–7380, 2016.

[22] Z. Shu, J. Wan, D. Li, J. Lin, A. V. Vasilakos, and M. Imran, "Security in Software-Defined Networking: Threats and Countermeasures," *Mobile Networks and Applications*, vol. 21, no. 5, pp. 764–776, 2016.

[23] Z. Shu, J. Wan, J. Lin et al., "Traffic Engineering in Software-Defined Networking: Measurement and Management," *IEEE Access*, vol. 4, pp. 3246–3256, 2016.

[24] G. Du and P. Zhang, "A markerless human-robot interface using particle filter and kalman filter for dual robots," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 4, pp. 2257–2264, 2015.

[25] H. Fischer, P. Vulliez, J.-P. Gazeau, and S. Zeghloul, "An industrial standard based control architecture for multi-robot real time coordination," in *Proceedings of the 14th IEEE International Conference on Industrial Informatics, INDIN 2016*, pp. 207–212, France, July 2016.

[26] A. C. Kapoutsis, S. A. Chatzichristofis, L. Doitsidis et al., "Real-time adaptive multi-robot exploration with application to underwater map construction," *Autonomous Robots*, vol. 40, no. 6, pp. 987–1015, 2016.

[27] P. Das, S. Pradhan, H. Tripathy, and P. Jena, "Improved real time A*-fuzzy controller for improving multi-robot navigation and its performance analysis," *International Journal of Data Science*, vol. 2, no. 2, p. 105, 2017.

[28] Z. Lukasik, B. Pniewska, R. Pniewski, and Z. Łukasik, *Rozproszony system sterowania robotem mobilnym*, IEEE, 2009.

[29] D. Kim, Y. Doh, and Y.-H. Lee, "Table driven proportional access based real-time Ethernet for safety-critical real-time systems," in *Proceedings of the Pacific Rim International Symposium on Dependable Computing, PRDC 2001*, pp. 356–363, Republic of Korea, December 2001.

[30] S. YANG, "Safety Critical Real-Time Networks Based on Ethernet Technology," *Journal of Software* , vol. 16, no. 1, p. 121, 2005.

[31] A. Ademaj, H. Kopetz, P. Grillinger, K. Steinhammer, and M. Prammer, "Integration of Predictable and Flexible In-Vehicle Communication using Time-Triggered Ethernet," in *Proceedings of the SAE 2006 World Congress & Exhibition*.

[32] W. Steiner, G. Bauer, B. Hall, M. Paulitsch, and S. Varadarajan, "TTEthernet dataflow concept," in *Proceedings of the 2009 8th IEEE International Symposium on Network Computing and Applications, NCA 2009*, pp. 319–322, USA, July 2009.

[33] A. Loveless, "On TTEthernet for integrated Fault-Tolerant spacecraft networks," in *Proceedings of the AIAA SPACE Conference and Exposition, 2015*, USA, September 2015.

[34] T. Steinbach, H.-T. Lim, F. Korf, T. C. Schmidt, D. Herrscher, and A. Wolisz, "Tomorrow's in-car interconnect? a competitive evaluation of IEEE 802.1 AVB and time-triggered ethernet (AS6802)," in *Proceedings of the 76th IEEE Vehicular Technology Conference, VTC Fall 2012*, Canada, September 2012.

[35] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, "Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges," *IEEE Access*, vol. 6, pp. 6505–6519, 2018.

[36] D. Tămaş–Selicean, P. Pop, and W. Steiner, "Design optimization of TTEthernet-based distributed real-time systems," *Real-Time Systems*, vol. 51, no. 1, 2014.

[37] N. I. Aristova, "Ethernet in industrial automation: Overcoming obstacles," *Automation and Remote Control*, vol. 77, no. 5, pp. 881–894, 2016.

[38] M. Selvatici, M. Ruggeri, and L. Dariz, "Advanced gateway services for real-time in-vehicle ethernet network," in *Proceedings of the 2015 IEEE International Conference on Industrial Technology, ICIT 2015*, pp. 1826–1831, Spain, March 2015.

[39] J. C. Yang et al., "Research towards IoT-oriented universal control system security model," *Journal of Communications*, 2012.

[40] D. Tamas-Selicean, P. Pop, and W. Steiner, "Synthesis of communication schedules for TTEthernet-based mixed-criticality systems," in *Proceedings of the the eighth IEEE/ACM/IFIP international conference*, p. 473, Tampere, Finland, October 2012.

[41] G. L. Du, P. Zhang, and X. Liu, "Markerless humanmanipulator interface using leap motion with interval Kalman filter and improved particle filter," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 2, pp. 694–704, 2016.

[42] P. F. Jin, VxWorks-based Software Bus and Its Autonomous Recovery Technology,.

[43] T. Steinbach, H. Dieumo Kenfack, F. Korf, and T. Schmidt, "An Extension of the OMNeT++ INET Framework for Simulating Real-time Ethernet with High Accuracy," in *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, Barcelona, Spain, March 2011.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 1857202, 16 pages
https://doi.org/10.1155/2018/1857202

Research Article

Industrial Internet of Things Based Efficient and Reliable Data Dissemination Solution for Vehicular Ad Hoc Networks

Shahid Latif,[1,2] Saeed Mahfooz,[1] Naveed Ahmad,[1] Bilal Jan,[3] Haleem Farman,[1] Murad Khan,[2] and Kijun Han[4]

[1]*Department of Computer Science, University of Peshawar, Khyber Pakhtunkhwa, Pakistan*
[2]*Department of Computer Science, Sarhad University of Science and Information Technology, Peshawar, Pakistan*
[3]*Department of Computer Science, Fata University, FR Kohat, Pakistan*
[4]*School of Computer Science and Engineering, Kyungpook National University, Daegu, Republic of Korea*

Correspondence should be addressed to Kijun Han; kjhan@knu.ac.kr

Received 21 December 2017; Revised 2 February 2018; Accepted 27 February 2018; Published 8 April 2018

Academic Editor: Jiafu Wan

Industrial Internet of Things (IIoT) is the other name of industrial Internet. It integrates a variety of existing industrial automation technologies with computing, machine learning, and communication technologies. Vehicular ad hoc network, an application of IIoT, is a self-organized network of vehicles which tends to provide improved road safety, diminished traffic congestion, and ultimate comfort to the travellers. In VANETs, vehicles exchange data with each other directly or through roadside units (RSUs). Data dissemination in VANETs experiences numerous challenging issues including broadcast storm, network partitions, intermittent connectivity between vehicles, and limited bandwidth. In literature, various data dissemination schemes are proposed. However, most of these schemes are designed for either urban or highway VANET scenarios and evaluated under sparse or dense traffic conditions. Moreover, these schemes do not effectively overcome the aforementioned issues simultaneously. In this paper, we present a new data dissemination protocol for VANETs, which disseminates the emergency messages in different scenarios under varying traffic conditions. During dense traffic conditions, DDP4V employs the segmentation of transmission region of a vehicle in order to select the most appropriate next forwarding vehicle (NFV). Accordingly, it divides the transmission region of a vehicle in three distinct segments and selects vehicle(s) inside the highest priority segment to forward the message to all neighbour vehicles, whereas it also uses implicit acknowledgements for guaranteed message delivery during sparse traffic Conditions. Simulation results show that DDP4V protocol outperforms the other existing related protocols in terms of coverage, network overhead, collision, and end-to-end delay.

1. Introduction

The Internet of Things (IoT) is a network of miscellaneous items such as physical devices, automobiles, and home appliances. embedded with recent electronics, sensing, networking, and communication technologies in order to connect and communicate. Industrial Internet of Things (IIoT), also known as industrial Internet, put forward the IoT technologies in manufacturing [1, 2]. IIoT integrates a large set of existing industrial automation technologies with recent electronics, computing, machine learning, and communication technologies. IIoT believes that smart machines are more competent than humans in collecting and communicating data. This data facilitates the industrial and business communities in business intelligence efforts. The vehicular ad hoc network, an application of IIoT, is a large network of vehicles communicating with each other and roadside units for sharing of information [3]. These are spontaneously created networks from interconnected vehicles for particular needs. VANETs aim to provide comfort to travellers and improve the road safety and traffic congestion. In VANETs, information is exchanged wirelessly between vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication in both highway and urban scenarios. Day by day, the increasing number of vehicles on roads raises several serious traffic problems including traffic jams and vehicle pile-ups. According to

World Health Organization (WHO), 1.25 million people die every year and 50 million injuries occur due to the road accidents [4]. Likewise, [5, 6] present records and statistics on the topic of accidents, fatalities, and injuries. Therefore, safety is one of the prime objectives of VANETs. These networks are also supposed to provide automated roadways, comfort, infotainment, environment protection [7, 8], and so forth. Presently, VANETs are emerging as an important research area and grabbing the attention of the researchers from both academia and industrial communities. VANETs are decentralized networks, where every vehicle node periodically broadcasts its information such as direction, position, and speed through beacon messages to inform its neighbour vehicles. Emergency situations like accidents or traffic jams make use of this information to alert other distant vehicles in the network. VANETs present a large set of applications categorized into two groups, that is, safety and nonsafety applications. Usually, safety applications need few small (data packet) messages with high latency and high-reliability ratios, while nonsafety applications such as traffic information, convenience, and entertainment need diverse messages with high data rate capabilities [9].

Data dissemination in VANETs is a data distribution process, where every vehicle transmits the message (data packet) to its neighbour vehicles inside the network. Usually, data dissemination in VANETs is broadcast-oriented. Thus the transmitting vehicle does not require the address and route information of the receiving vehicles. This leads the VANETs to eliminate the need for several complex issues such as address design and resolution, route finding, and topology management. Data dissemination in VANETs experiences many challenging issues such as broadcast storm, network partitions, intermittently connected network, and limited bandwidth. In the presence of all these issues, an efficient and robust data dissemination process turns out to be an exceptional and difficult task. Most of the automobile industries are concentrating on the efficient data dissemination in VANETs. Literature presents several data dissemination protocols designed for various VANETs scenarios to operate under diverse traffic conditions [10, 11].

This paper focuses on data dissemination technique that addresses the broadcast storm, network partition, and intermittently connected network problems simultaneously. The broad picture of the paper is summarized below as major contributions of our work:

(i) At first, an efficient and reliable data dissemination protocol is developed to manage data dissemination in highway and urban VANET scenarios.

(ii) Secondly, wagon wheel model is introduced, which divides the transmission area of source vehicle into three distinct segments. This segmentation assists the proposed protocol to select the best vehicle among many to carry on the dissemination process and hence mitigate the broadcast storm problem.

(iii) Additionally, the number of redundant transmissions is controlled and reduced through careful selection of the next forwarding vehicle using NFV selection algorithm to tackle the scalability issue under dense traffic conditions.

(iv) Furthermore, in-depth review of the important characteristics of several existing data dissemination protocols in VANETs is also presented.

The rest of the paper is organized as follows: Section 2 presents the background knowledge, related work, and qualitative comparison of various data dissemination protocols. Section 3 focuses on the proposed DDP4V protocol. Section 4 covers the performance evaluation and corresponding results, while Section 5 concludes this work.

2. Related Work

In literature, a large number of protocols have been designed and proposed for data dissemination in VANETs. These protocols can be categorized into two groups on the basis of road scenarios: protocols for highway and urban VANET scenarios.

2.1. Data Dissemination Protocols for Highway Scenarios. Distributed Vehicular broadCAST (DV-CAST) protocol [12] and Simple and Robust Dissemination (SRD) [13] protocols were proposed to work under both sparse and dense traffic conditions. Both DV-CAST and SRD depend on solely one-hop neighbour information gathered through periodic beacon messages. These protocols work in three steps: (1) determine neighbour vehicles and concerned information through periodic beacon messages to establish local topology, (2) employ the broadcast suppression mechanism to mitigate the broadcast storm problem, and (3) use the store-carry-forward mechanism to deal with partitioned and fragmented networks. SRD improves the robustness, high delivery ratio, and low delivery delay of the DV-CAST protocols. Both protocols operate well in well-connected and intermittently connected networks. However, both were designed for highway scenario and did not perform well in high mobility scenarios as the formation of ideal beacon frequency turns out to be difficult. Likewise, DRIFT protocol [14] addresses both the broadcast storm and network fragmentation problems simultaneously.

Unlike DV-CAST and SRD, DRIFT protocol does not need neighbour vehicles information to disseminate data. DRIFT protocol assigns highest forwarding priority to the vehicle(s) located inside the preference zone to disseminate data packet further and it consequently provides efficient data dissemination along with low overhead, small delay, and extended coverage. It mitigates the broadcast storm through a controlled selection of NFVs and deals with intermittently connected networks using store-carry-forward mechanism. Another such protocol that does not depend on neighbour information during data dissemination process is Speed Adaptive Broadcast (SAB): Probabilistic SAB, Slotted SAB, and Grid SAB [15]. It uses speed information of vehicles to determine the traffic conditions and select the most suitable NFV. Similarly, another beaconless approach is Simple and Efficient Adaptive Data (SEAD) dissemination protocol [16]. Herein each vehicle updates a redundancy ratio to determine

its neighbour vehicles density. Subsequently, each vehicle node computes its forwarding priority and corresponding waiting time on the basis of slotted-1 persistence approach and calculated redundancy ratio.

A protocol termed BROADCOMM has been proposed, which disseminates the emergency warning messages in VANETs using geographical routing [17]. It divides the highway into virtual cells, establishing a virtual infrastructure moving with vehicle's movement. This protocol uses two-level hierarchical structure. In the first level, vehicles inside a cell communicate with each other and with vehicles of neighbour cells located inside the communication range. In the second level, vehicles closer to the geographical centre of a cell are selected for communication with other vehicles in the same as well as the neighbour cell. These vehicles are termed as Cell Reflectors (CR) and work like cluster heads (CHs). CR of each cell is selected by the vehicles of that cell through a selection process. An additional, cluster-based safety message broadcast approach is VMaSC-LTE [18] that combines cluster-based multihop 802.11p approach (Vehicular Multihop algorithm for Stable Clustering (VMaSC)) with cellular system Long-Term Evolution (LTE). It employs IEEE 802.11p to deal with the broadcast storm and intermittently connected network problems, while cellular system LTE is used due to its wide-range communication and low latency. Herein vehicles are clustered using VMaSC approach followed by cluster head (CH) selection. Elected CHs perform as dual-interface nodes to connect the VANET to LTE network.

Probabilistic data dissemination protocols have been largely used for data distribution in VANETs. Irresponsible Forwarding (IF) [19] selects the NFV on the basis of (1) distance from the source vehicle and (2) density of neighbour vehicles. In IF protocol, each vehicle determines whether there is such vehicle among its neighbours, which explains comparatively high forwarding probability. If yes is the case, then the former vehicle does not retransmit the data packet "irresponsibly" and leave it for the later vehicle to retransmit. It mitigates the broadcast storm by controlling and regulating the transmission rate of data packets. Similarly, cluster-based Irresponsible Forwarding (CIF) [20], an event-driven broadcast protocol, is derived from IF. It combines the probabilistic approach of IF with the cluster-based structure to increase the reliability and performance of actual IF. Similarly, three distributed and probabilistic data dissemination protocols, weighted p-persistence, slotted 1-persistence, and slotted p-persistence [21], were designed for increased reach ability, reduced number of collisions at MAC layer, and decreased redundancy in dense traffic conditions. Accordingly, each vehicle computes its retransmission probability using information obtained from global positioning system before transmitting the received data packet.

MobiCast [22] protocol is designed for some particular VANET applications that involve spatiotemporal coordination and support time-based changes in the network topology. In MobiCast, vehicles transmit data packets to their neighbours connected for real-time communication inside the zone of relevance (ZoR). It also uses another dynamic time-variant area, that is, zone of forwarding (ZoF) inside ZoR. Vehicles located inside ZoF are responsible for further retransmission of received data packet inside a network. Main complexity of this protocol is the selection of ZoR, ZoF, and vehicles inside the ZoF through a complex technique. Likewise, MobiCast-Carry-Forward (MCF) protocol [9], an improvement over actual MobiCast, was proposed for comfort applications in VANETs based on a prescribed geographical area (ZoR). The major difference between the two is the use of the carry-forward technique in MCF, which results in higher end-to-end delay (comparatively) and might not be acceptable in emergency applications. It improves the actual MobiCast protocol in terms of dissemination success rate and network overhead.

Emergency-Degree-based broadCast (EDCast) [23] is designed on the basis of safety information quantity. Therefore, an emergency-degree parameter is used to evaluate the information quantity in broadcast safety messages. EDCast computes priority of each safety message to access the channel based on emergency degree. Hence it offers precise and up-to-date vehicular information to the vehicles of a network.

2.2. Data Dissemination Protocols for Urban Scenarios. Urban Multihop Broadcast (UMB) [24] protocol based on IEEE 802.11 standard was designed to deal with broadcast storm via multihop V2V and V2I communication. UMB protocol selects the farthest vehicle from the sender as the next data packet forwarder in the broadcast direction. It makes use of repeaters at an intersection in order to broadcast data packet in all directions within the area of interest. It is worth noticing that use of repeaters at every intersection may not be feasible in various scenarios. Conversely, Ad hoc Multihop Broadcast (AMB) protocol [25], an extension to UMB protocol, eliminates the need of infrastructure support (until obstacles block the sight line among different road segments at intersections). Comparatively, AMB protocol confirms high success rate and efficient bandwidth utilization even at high data loads. Similarly, Distance-Based Relay Selection (DBRS) protocol [26] uses distance-based forwarding strategy to control and regulate the number of forwarding vehicles. In DBRS, each vehicle retains the received data packet for a certain waiting time. The waiting time is inversely proportional to the distance between the receiving and the transmitting vehicles. Therefore, farthest vehicles have shortest waiting time and vice versa. DBRS performs well in mitigating the broadcast storm by controlling retransmissions of same data packet but may face low coverage. Also, lack of vehicles at the edges of transmission range may incur high delays.

A decentralized data dissemination protocol, Adaptive approach for Information Dissemination (AID) [27], makes use of counter-based forwarding strategy to select the NFV. In AID, selection of NFV depends upon the number of the same data packets received in a specified period. It mitigates the broadcast storm as several vehicles discard the scheduled data packet received from multiple other vehicles. AID does not address the network partition and network fragmentation problems. U-Hydi [28] protocol addresses both the broadcast storm and network fragmentation problems simultaneously. It uses both sender-based and receiver-based approaches to mitigate the broadcast storm in dense urban scenarios. It

makes use of store-carry-forward technique to ensure data packet delivery through fragmented networks.

A geocast Data dissemination Protocol based on Map Splitting (DPMS) [29] does not require any infrastructure support. It splits the map of an urban scenario into multiple regions. Afterwards, DPMS creates a zone of relevance (ZoR) by combining multiple significant regions where the event arises. It requires a smart technique for digital map autosplitting. Similarly, Hybrid Based Election Backbone (HBEB) [30] and Network-Coding-Assisted Scheduling (NCAS) [31] protocols use backbone-based and network-coding-based forwarding strategies, respectively, to disseminate data. In HBEB, small backbones with a single originator (backbone node) are formed for the fast and efficient propagation of data inside a network. Selection of backbone nodes is accomplished through contention-based forwarding mechanism. The NCAS protocol performs both V2V and V2I communications and employs them for efficient data services. It also makes use of cache strategy that allows vehicle nodes to recall their unrequested data and hence improve data sharing and data services.

3. Data Dissemination Protocol for Vehicular Ad Hoc Networks (DDP4V)

To address the problems of broadcast storm, network partition, intermittently connected network, and most suitable NFV selection mentioned in the previous section, we propose an efficient data dissemination protocol for VANETs (DDP4V). The proposed protocol evenly distributes emergency messages across the network and maximizes the data dissemination inside concerned area of a network with low overhead, low collisions, and smaller delay. It operates in both highway and urban VANET scenarios under diverse traffic conditions. The proposed protocol mitigates the broadcast storm though controlled number of redundant transmissions. It provides the guaranteed distribution of data across network partitions and intermittently connected networks using vehicles outside the concerned area. DDP4V employs the segmentation of transmission region (termed as wagon wheel) of a vehicle to select the most appropriate NFV and enhance the coverage.

Definition (Wagon Wheel). It presents a circular area around a vehicle where it can directly communicate with its neighbours. This area is divided into two distinct segments, that is, behind and ahead segments. Behind segment is further divided into two subsegments: ideal and normal. The segmentation of transmission region is based on the significance of vehicles to be selected as the next forwarder. Each segment has a unique packet forwarding priority such that ideal segment comprises the highest priority followed by normal and then ahead segment.

Figure 1 depicts the concept of a wagon wheel. Here neighbour vehicles can be located indiscriminately anywhere inside the three segments of wagon wheel. In our case, the transmission region of a vehicle is considered to be a circular area around itself ranging from 200 to 300 meters. However, shapes other than circle also work with the proposed protocol. The prime advantage of using the wagon wheel concept is to avoid the transmissions of the same data packet by the neighbour vehicles (closer to each other) located in different segments and covering similar or even less significant area, hence mitigating the broadcast storm problem. Vehicles inside the ideal segment are most appropriate and sufficient to carry on data dissemination process with reduced number of transmissions (network load).

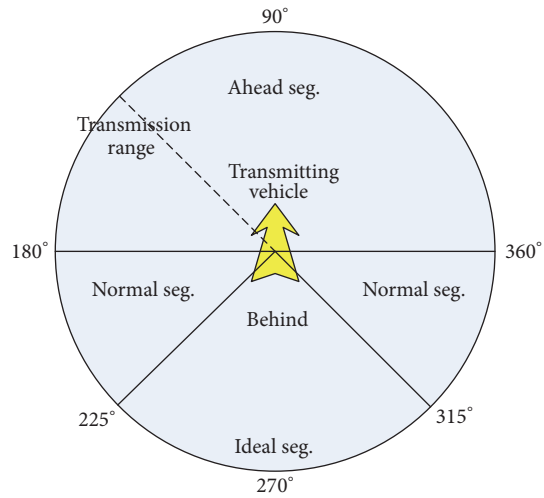


Figure 1: Wagon wheel (segmentation of transmission range).

3.1. Proposed Protocol. The proposed protocol for data dissemination is divided into various phases as shown in Figure 2. The whole process is divided into the following six phases.

3.1.1. Emergency Detection Phase. In this phase, an emergency is detected or experienced by the source vehicle (also called the originating vehicle). The emergency is produced by an arbitrary vehicle that stops abruptly at road describing an accident. We assume some basic information of 1-hop neighbour vehicles such as vehicle ID, position, moving direction, and distance concerning the source. Information about vehicles is collected through the position-aware system such as global positioning systems. These vehicles share this information with source as well as other vehicles of the network to establish the network topology accordingly. This information is later used by the source vehicle to determine the forwarding priorities of vehicles and the best NFV.

3.1.2. Emergency Message Broadcast Phase. The source vehicle initiates the data dissemination process by broadcasting data packet containing an emergency message. The data packet is disseminated in the opposite direction to inform the drivers approaching the place of emergency. One of the aims of proposed method is to distribute a data packet through multihop communication in order to be aware of maximum number of vehicles from this communication. It is worth noticing that DDP4V protocol does not require maintaining and updating neighbour information table, which turns out to be difficult during dense traffic conditions.

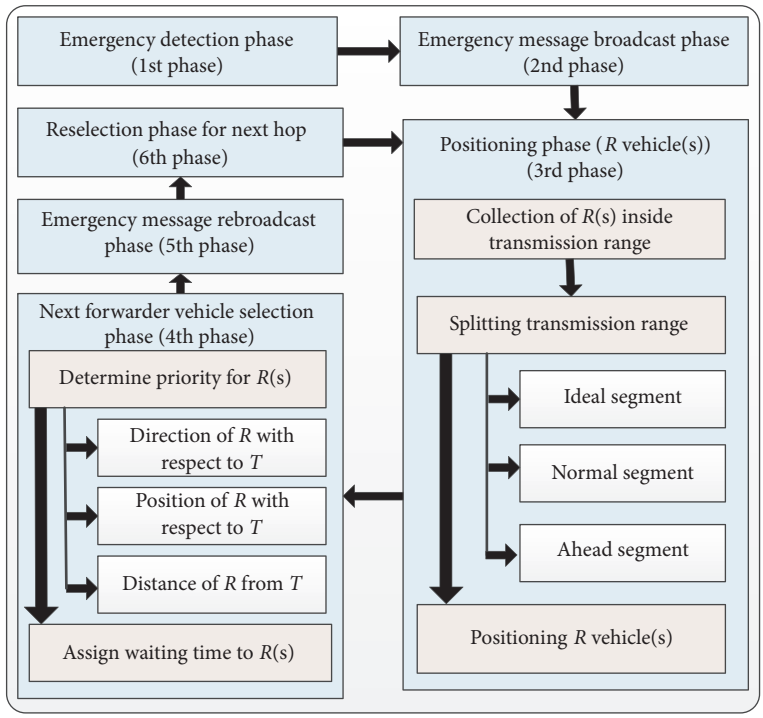


Figure 2: The proposed protocol for data dissemination in VANETs.

3.1.3. Positioning Phase. The proposed protocol requires an appropriate classification of neighbour vehicles to select the best option for further retransmissions and continue data dissemination in VANETs. In this phase, at first, the information collected from neighbouring vehicles is used to determine the number of receiving vehicles inside the transmission region. Afterwards, the source vehicle divides its transmission region into different segments, where each segment represents an area identified by a unique label, as shown in Figure 1. Once the segmentation is completed, source determines the position of each receiving vehicle (denoted by R) inside these segments and categorizes them into three groups; ideal, normal, and ahead vehicles. Hence, farthest vehicles inside the ideal segment moving towards source vehicle are the best option to be the next forwarder of the data packet. Using this approach, a single farthest vehicle inside the ideal segment is enough to retransmit the data packet with the lowest delay and cancel the redundant retransmissions as well. In case no vehicle exists inside the ideal segment, the farthest vehicle(s) from normal segment will forward the data packet to carry on the dissemination process. Similarly, if no vehicle exists inside both the ideal and normal segments, then the farthest vehicle(s) from ahead segment will forward the data packet further. Herein, vehicles inside the high priority segments are supposed to deliver the data packet to a maximum number of neighbour vehicles directly not accessible to the source vehicle. Algorithm 1 presents the process whether the receiving vehicle (R) is positioned in ideal, normal, or ahead segment of the wagon wheel.

3.1.4. Next Forwarder Vehicle Selection Phase. Next forwarder vehicle is responsible for further retransmission of received data packets to neighbours inside the concerned area of a network. The overall performance of a protocol strictly depends on the selection of NFV. Thus, the NFV selection process is very critical for any efficient data dissemination protocol and requires some criteria before selection.

NFV Selection Criteria. It is important to select the best vehicle as NFV among all participating vehicles of a network. In the proposed protocol, NFV selection depends on three different parameters; (1) position, (2) distance, and (3) direction concerning source vehicle. Selection decision based on these three parameters ensures routing the data packet effectively towards the target. These parameters are represented by POS, DIST, and DIR, respectively, and aggregated to come up with *cumulative value* (CV) for each receiving vehicle R_i as follows:

$$\mathrm{CV}_{R_i} = \text{aggregate}\left\{\mathrm{POS}_i + \mathrm{DIST}_i + \mathrm{DIR}_i\right\}. \quad (1)$$

Position parameter defines the location of a vehicle inside the transmission region. The position of R_i vehicle inside the ideal segment increases its chances to become the NFV. Similarly, distance parameter represents the maximum distance of a vehicle from source vehicle. Farthest R_i vehicle from source has more chances to become the NFV. Likewise, the direction parameter describes the direction of a vehicle concerning source. In proposed protocol, R_i vehicles moving towards the source vehicle will directly participate in data dissemination process during dense traffic conditions. Vehicles moving

(1) **Procedure** Next Forwarder Vehicle (NFV) Selection
(2) **Input**: **N** // all vehicles within Concern Area
(3) **Source**// vehicle (**S**) that initiate the data dissemination process
(4) (**Tx, Ty**)// x and y coordinates of transmitter vehicle (**Tj**) during 2nd and onward hops
(5) (**Rx, Ry**)// x and y coordinates of receiver vehicle ($\mathbf{R_i}$)
(6) **Output**: number of vehicle(s) to carry on data dissemination process// NFV(s)
(7) **if** the message is received for first time **then**
(8) Direction $\mathbf{R_i}$ to **Tj** = $\angle|(\overrightarrow{\mathbf{T'T}}, \overrightarrow{\mathbf{R'R}})| \leq \boldsymbol{\theta}$ ***for some threshold value of*** $\boldsymbol{\theta}$// R_i & Tj same/opposite direction
(9) Position $\mathbf{R_i}$ to **Tj** = angle = atan2$(Ty - Ry, Tx - Rx)$ // two argument function atan2 (arctangent function)
(10) Distance $\mathbf{R_i}$ to **Tj** = $\sqrt{(Tx - Rx)2 + (Ty - Ry)2}$// distance between R_i and Tj
(11) Defaultdelay = 0.01 × (distance R to S/communicationRadius)// default delay of R_i
(12) **if** R_i inside the behind segment i.e. (angle ≥ 181° and angle ≤ 360°) **then** //waiting time for priority 1 & 2
(13) **if** R_i inside the ideal segment i.e. (angle ≥ 226° and angle ≤ 324°) **then**
(14) **Delay** = defaultdelay + random(0, 0.01); //set waiting time for priority 1
(15) **end**
(16) **else** //set the waiting time for and priority 2
(17) **Delay** = defaultdelay + random(0.02, 0.04);
(18) **end**
(19) **end**
(20) **else** // if inside the ahead segment, set waiting time for priority 3
(21) **Delay** = defaultdelay + random(0.05, 0.07);
(22) **end**
(23) $\mathbf{R_i}$ .ScheduleMessage(Delay); // R_i has to schedule the transmission
(24) **end**
(25) **else**
(26) **if** Scheduled message **then** // if already scheduled, cancel scheduling and discard
(27) **if** Dist($\mathbf{R_i}$, **S**) > Dist(**Tj**, **S**) **then**// calculate the distance
(28) Discards the received message;
(29) **end**
(30) **end**
(31) Cancel message scheduled;
(32) **end**
(33) **end Procedure**

Algorithm 1: Verifies whether the receiving vehicle (R) is positioned in ideal, normal, or ahead segment of the wagon wheel. It also computes the corresponding waiting time (P1/P2/P3) for R_i vehicles to schedule the transmission of received data packet.

away from source vehicle will participate in data dissemination process during sparse traffic conditions to deal with network partition and/or intermittently connected network problems.

NFV Selection. Once the NFV selection criteria are set, the next forwarder vehicle is selected for data packet transmission in its first hop. As discussed earlier, a vehicle with maximum cumulative value will be selected as the NFV for the first hop. In DDP4V protocol, vehicles do not need to maintain the neighbor information table containing the cumulative value of all vehicles in a network. The reselection process of NFV for forthcoming hops (2nd, 3rd, and so on) is decentralized, which does not require the involvement of source vehicle.

3.1.5. Emergency Message Rebroadcast Phase. After aggregating the cumulative value, calculated by source vehicle, each R_i vehicle is assigned a specified waiting time before the retransmission of received data packet. A vehicle with highest cumulative value is designated with shortest waiting time. The computation and assignment of waiting time, that is, P1, P2, or P3, to each R_i vehicle are shown in Algorithm 1. Once the waiting times are set, each R_i vehicle schedules the data packet retransmission. A vehicle with shortest waiting time immediately retransmits the data packet after completing its waiting time in order to carry on the dissemination process.

3.1.6. Reselection Phase. In this phase, our focus is to avoid the reselection of such vehicles, as the NFV, which were previously involved in the dissemination process. By reconsidering such vehicles, the proposed protocol incurs the increased number of retransmissions and network overhead. Therefore, for such purpose, the proposed protocol DDP4V adopts the cancellation strategy of already scheduled data packets. Accordingly, on receiving the data packet again during coming hops, vehicles cancel the retransmission of already scheduled data packet and discards. The same procedure (3rd to 5th phase) will be repeated for the reselection of NFV in the next hops (2nd, 3rd, . . . and so on) during data dissemination process.

3.2. Data Dissemination in Urban VANET Scenarios. The data dissemination in urban VANET scenario is performed

using two different scenarios: (1) data dissemination with no network partitions and (2) data dissemination across network partitions.

3.2.1. Data Dissemination with No Network Partitions. In DDP4V, once an emergency is detected by a vehicle, it immediately starts the data dissemination by flooding an emergency data packet (denoted by M) within the concerned area. As discussed earlier, vehicles located inside the ideal segment are the best options to be the NFV(s) to continue the dissemination process and mitigate the broadcast storm problem, explained in Figure 4. By receiving the data packet, each vehicle (Ri) verifies whether it is inside the concerned area or not. If yes, then allow Ri to participate in the data dissemination process (Figure 4, point A); otherwise, it can be used in network partitions. If R_i lies inside the concerned area, then it checks whether it is within the transmission range of the transmitting vehicle (Figure 4, point B). If not, R_i simply discards the received data packet. Otherwise, it verifies whether the data packet is received for the first time (Figure 4, point C). If not, R_i verifies whether the received data packet is already scheduled (Figure 4, point D). If the data packet is already scheduled, it cancels the scheduled data packet and then discards it. Otherwise, it directly discards the received data packet. In case R_i receives the data packet for the first time, it verifies whether it is inside the ideal segment of wagon wheel (Figure 4, point E). If so, R_i computes its waiting time (P1). Otherwise, it checks whether it is inside the normal segment (Figure 4, point F). If so, R_i computes its waiting time (P2). Otherwise, it sets its waiting time (P3), respectively, as shown in Algorithm 1. Hence, R_i continues the data dissemination process by flooding the received data packet inside the concerned area.

3.2.2. Data Dissemination across Network Partitions. An urban scenario with network partitions discontinues the data dissemination process as shown in Figure 3. As data packet carrying information about the event does not propagate within concerned area of a network, whenever a source vehicle detects network partition(s), it makes use of the vehicle(s) outside the concerned area to disseminate data packet in a network. The main advantage of using vehicles outside the concerned area is to perform data dissemination among vehicles within concerned area separated by network partitions. Upon receiving the data packet M, each R_i vehicle outside the concerned area verifies whether network partition(s) exists or not (Figure 4, point G). If no, then R_i simply discards the received data packet. Otherwise, it checks if the data packet is received for the first time (Figure 4, point H). If no, then R_i verifies whether the received data packet is already scheduled (Figure 4, the point I). If the data packet is already scheduled, it first cancels the already scheduled data packet and then discards it; otherwise, it directly discards the received data packet. In case R_i receives the data packet for the first time, it calculates the waiting time and schedules the transmission of a data packet for calculated time. The computation process of waiting time for vehicles outside the concerned area is presented in Algorithm 2.

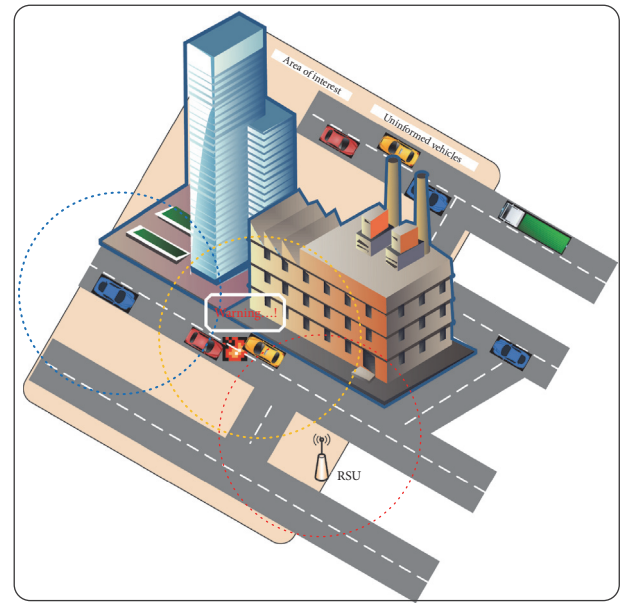


Figure 3: An urban scenario with network partitions.

3.3. Data Dissemination in Highway VANET Scenarios. In the proposed protocol, the dissemination of received data packet in the highway scenarios is the same as in urban scenarios. The main difference is that, in highway scenarios, before cancelling a scheduled data packet (M), R_i determines its distance from S vehicle as shown in Algorithm 1. If R_i is closer to S than T_j, then scheduling is cancelled; otherwise, the received data packet is discarded as its transmission is not required anymore.

4. Results and Discussions

The performance of the proposed method is evaluated using the network simulator OMNET++ [32], the Vehicular Network Simulation framework Veins [33], and the Simulator for Urban MObility (SUMO) [34]. Moreover, the proposed method is compared with existing well-known data dissemination protocols including Simple Flooding, AID [27], DBRS [26], and DV-CAST [12]. The proposed DDP4V approach is evaluated in terms of coverage, a number of data packets transmitted, delay, and number of collisions.

4.1. Evaluation Metrics. To evaluate the performance of the proposed DDP4V protocol against existing data dissemination protocols, the following evaluation metrics were considered:

(i) Coverage: it presents the number of vehicles (in percentage) inside the concerned area which is truly receiving the data packets. In other words, it describes the relation between the total vehicles inside the concerned area and those vehicles that received the data packets. More coverage (approaching 100%) specifies the more reliable solution for data dissemination in VANETs

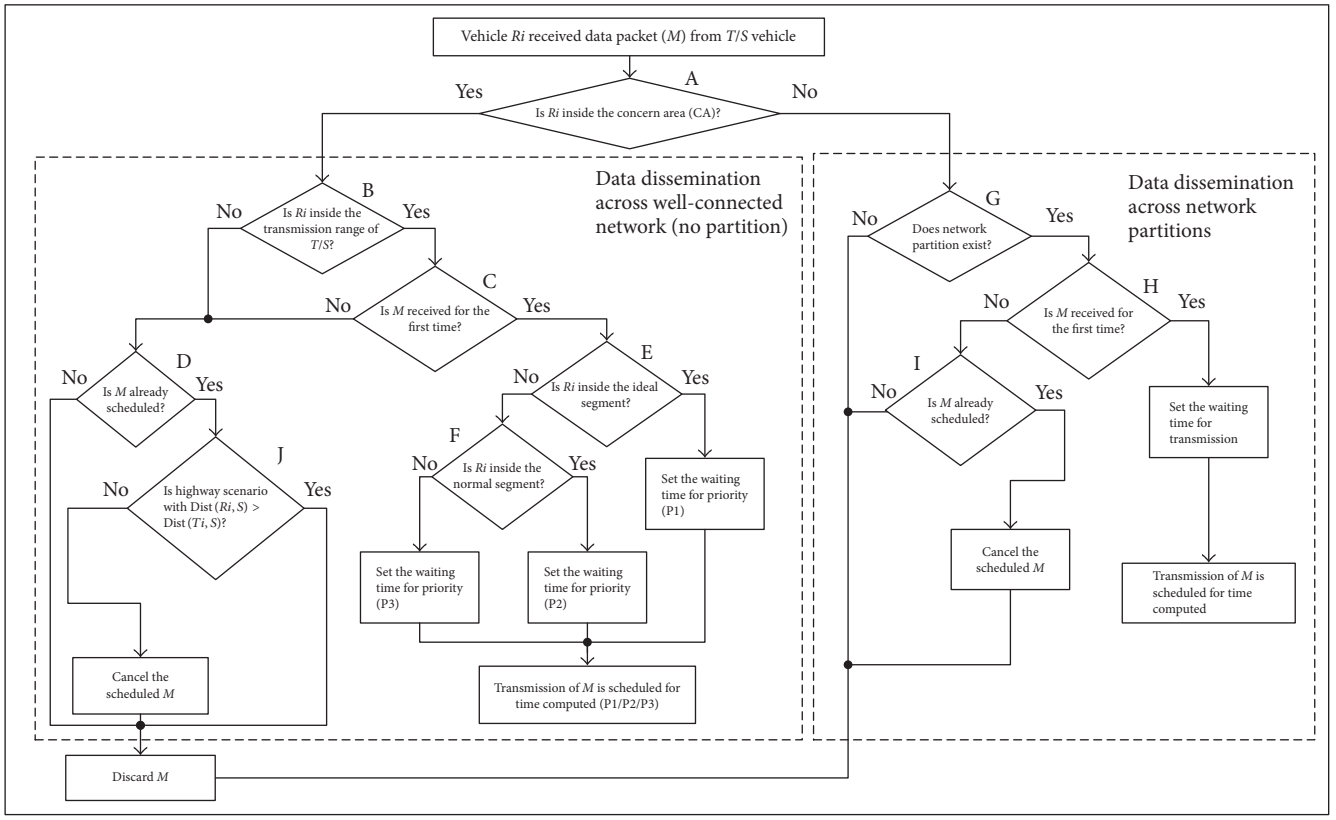


Figure 4: Data dissemination process of DDP4V.

(1) **Procedure** Waiting Time for R during Network Partitions //target is the center of partitioned area inside CA
(2) **Input**: M //all vehicles outside the Concern Area (CA)
(3) ($SPos_x$, $SPos_y$) //position of source vehicle (S) that initiate the data dissemination process
(4) ($RPos_x$, $RPos_y$) //position of receiving vehicle (R)
(5) ($holeGeometry_x$, $holeGeometry_y$) // partitioning the concern area
(6) **Output**: Waiting Time //calculated waiting time for R to schedule the transmission of data packet (M)
(7) **if** $RPos_x > holeGeopmetry_{x1}$ then
(8) Partitioned $V_x = SPos_x - CA/2$;
(9) **end if**
(10) **else**
(11) Partitioned $V_x = SPos_x + CA/2$;
(12) **end else**
(13) Partitioned $V_y = (holeGeometry_{y2} - holeGeometry_{y1})/2$;
(14) DistParitioned $V = \sqrt{(\mathbf{RPos_x} - \mathbf{partitioned\ V_x})^2 + (\mathbf{RPos_y} - \mathbf{partitioned\ V_y})^2}$
(15) Delay = 0.07 × (distParitioned V/CA/2);
(16) **end Procedure**

Algorithm 2: Computes the waiting time for R vehicle positioned outside the concerned area to schedule transmission of received data packet (M) across network partitions.

(ii) A number of data packets transmitted: it corresponds to the total number of data packets transmitted by all vehicles in a network during dissemination process. The high transmission rate of data packets can be due to redundant packets which might lead to broadcast storm

(iii) Delay: it is the amount of time elapsed by disseminated data packets to pass through the entire network, that is, from source to the target vehicles within the concerned area. Low delay is an essential parameter of dissemination protocol especially in applications having time constraints such as emergency messages distribution

(iv) Number of collisions: it describes the average number of data packets' collisions at MAC layer for all vehicles inside the concerned area. High collision rate during dense traffic conditions may lead the network to face the broadcast storm problem. Thus a minimum number of collisions (packet loss ratio) are required for efficient and reliable transmission

4.2. Highway VANET Scenario. The proposed DDP4V protocol operates in diverse VANET scenarios with diverse traffic conditions. Initially, we considered a 10-kilometer-long and straight three-lane highway scenario, where vehicles can move in two opposite directions, that is, from east to west and from west to east. At each opposite edge of the highway, vehicle flow production is established generating and inserting vehicles into the network with a constant rate of 800, 1000, 1200, 1400, 1600, 1800, and 2000 vehicles/hour. In this scenario, the overtaking of vehicles is also ensured and achieved by inserting three types of vehicles: high, moderate, and slow speed vehicles in the network using vehicle production flows. Among these three types, vehicles are capable of reaching a maximum speed of 33 meters/sec, 26 meters/sec, and 20 meters/sec, respectively. Therefore, a scenario with three types of vehicles having different attributes may be considered as a dynamic vehicular network. Number of these vehicles is set different during the simulation process. Each simulation has vehicles with a ratio of 2 : 1 : 1 (i.e., 50% of high, 25% of moderate, and 25% of low speed) of the total vehicles.

Once the simulation gets stable, after certain period, an arbitrary vehicle stops abruptly at highway describing an accident. It generates a data packet of 2048 bytes and initiates data dissemination process within the concerned area of 5 km length. At this point, the data packet is an emergency awareness message having information regarding the accident. It has to be disseminated in the opposite direction of the highway in order to inform the drivers approaching the place of the accident. Thus, our intention is to distribute the emergency awareness message through multihop communication within the specified time in the west direction of the highway to alert a maximum number of vehicles moving towards the east side of highway (place of accident). Furthermore, the bit rate is set to 18 Mbits/sec (at MAC layer) along with the transmission power of 1.6 mW, resulting in a transmission range of approx. 250 meters for each vehicle under a two-ray ground propagation model [35].

Table 1: Simulation parameters for highway scenario.

Parameter	Value
Network simulator	
Transmission power	1.6 mW
Transmission range	250 m
Bit rate	18 Mbit/s
Concern area	5 Km
Data message size	2048 bytes
Number of runs/iterations	50+
Mobility simulator	
Speed (max.), 1st type	33 m/sec
Speed (max.), 2nd type	26 m/sec
Speed (max.), 3rd type	20 m/sec
Acceleration	0.8 m/sec
Deceleration	4.5 m/sec
Road direction/lanes	Two-way/3

Every point of the results (shown in Figure 5) represents an average of more than 50 iterations. Table 1 covers the summary of key simulation parameters used in the highway scenario.

In Figure 5, the results for DDP4V and other protocols at highway scenario with diverse traffic conditions are illustrated. Accordingly, Figure 5(a) presents the coverage results for DDP4V and all other protocols in terms of successful data dissemination at highway scenario under various traffic flows. It can be noticed that only two protocols, that is, DDP4V and Flooding protocol, acquire maximum possible coverage inside the concerned area. Among all the compared protocols, DDP4V presents the best performance in terms of coverage at all traffic circumstances and simulation intervals, that is, at an average of 96%. DDP4V produces closer results as the Flooding protocol does; however, Flooding protocol suffers from a high number of transmissions, which leads the network towards the broadcast storm.

Figure 5(b) presents a total number of transmitted data packets within the concerned area for all protocols during dissemination process at highway scenario under various traffic flows. It evaluates the efficiency of data dissemination protocol in terms of minimizing the broadcast storm under dense traffic situations. Thus, less number of transmissions confirms better performance in terms of mitigating the broadcast storm. Flooding protocol generates the highest transmission rate as it does not use any broadcast suppression mechanism. On one hand, AID, DBRS, and DV-CAST generate less transmission rate as compared to the DDP4V protocol, but, on the other hand, they present less coverage area inside the concerned area (Figure 5(a)). DDP4V protocol presents high coverage with acceptable low overhead. DBRS is the protocol with the lowest overhead due to its low data packet delivery to all other vehicles within concerned area. Among the group of protocols that provide 100% coverage, DDP4V shows lower overhead than Flooding protocol by approximately 55%.

Delivery delay evaluates the performance of data dissemination protocol in terms of average amount of time it requires

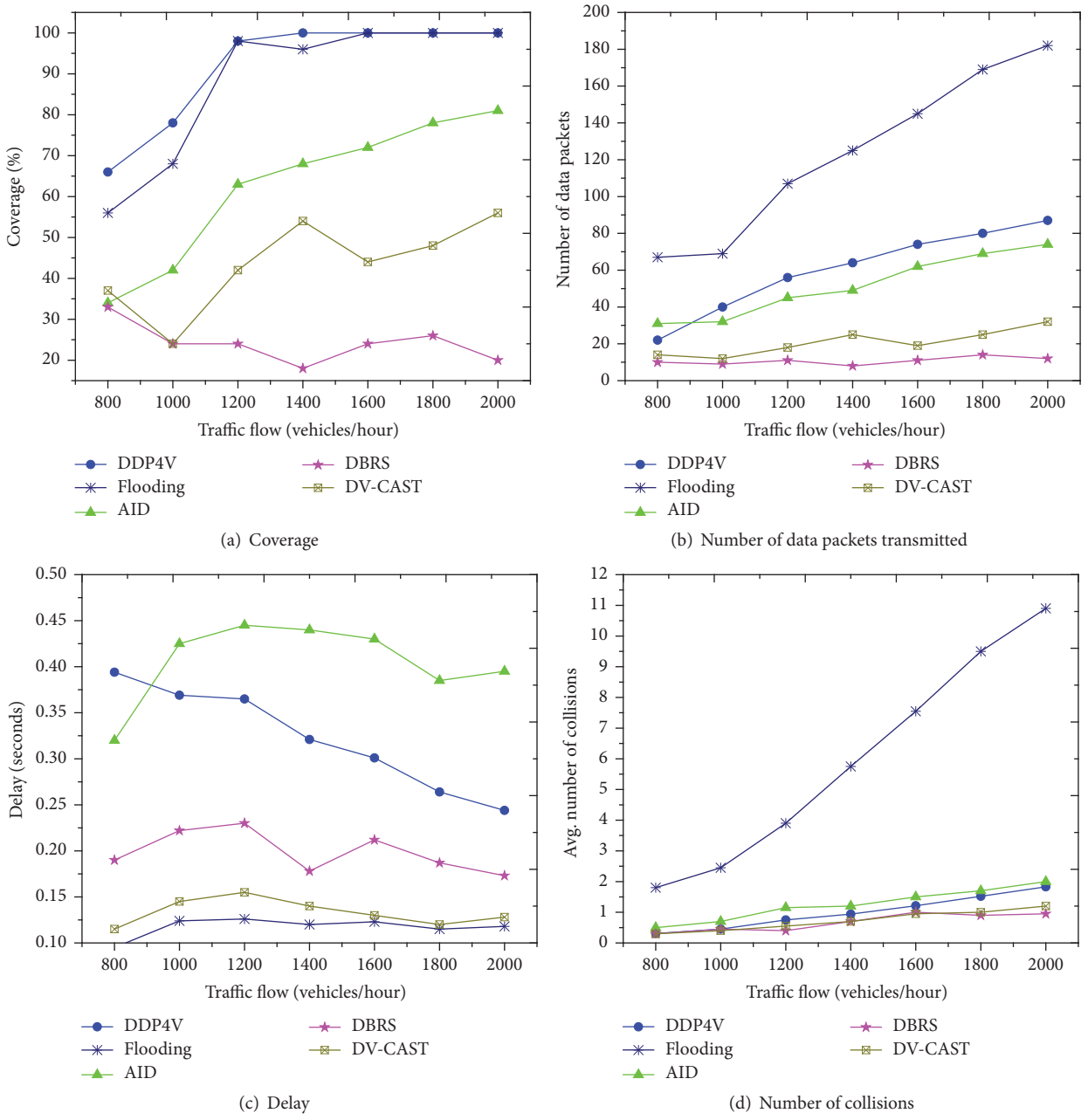


Figure 5: Simulation results for highway scenario.

to deliver a data packet from source to the target vehicle inside the concerned area. Figure 5(c) shows the average delay for all protocols and illustrates that AID protocol suffers from the highest delay. In fact, AID waits for data packet receptions from other vehicles (discussed in Section 3.2), and on the basis of these receptions, each vehicle decides whether to forward the data packet or leave it for any other vehicle. In DDP4V, delivery delay is inversely proportional to vehicles density inside the transmission range of a vehicle. Thus a large number of vehicles present minimum delivery delay as vehicles inside the ideal segment retransmit the data packet with shortest waiting time. When compared to DDP4V protocol, the DV-CAST and DBRS protocols show smaller delays but also present low coverage and small propagation distance. Similarly, Flooding protocol confirms the lowest delay with a maximum risk of broadcast storm problem when traffic density increases the concerned area. Although DDP4V protocol presents little extra delay when compared to some protocols during data packet dissemination, still its overall delay decreases significantly with the presence of vehicles inside the ideal segment.

The number of collisions represents an average number of data packets' collisions during the dissemination process. This parameter plays a fundamental role in the performance

Table 2: Simulation parameters for urban scenario.

Parameter	Value
Transmission power	0.98 mW
Transmission range	200 m
Bit rate	18 Mbit/s
Concern area	1 Km^2
Grid numbers	20/10 (with/without partitions)
Grid length	2000/1000 m (with/without partitions)
Number of lanes	2
Data message size	2048 bytes
Number of iterations	50

evaluation of data dissemination protocol, especially in dense traffic conditions. Figure 5(d) shows all results for data packet collisions. Accordingly, Flooding protocol, among all the analyzed protocols, presents highest number of collisions that gradually increases with increase in traffic density. This happens because of lack of coordination among vehicles in data packet forwarding process; that is, several vehicles blindly retransmit and try to access the limited available bandwidth simultaneously. The residual protocols along with DDP4V present approximately similar behaviour in terms of collisions, that is, on average 1 collision/vehicle. The DDP4V protocol uses effective broadcast suppression technique to mitigate the broadcast storm in highway scenario and deliver data packet to all vehicles inside the concerned area along with acceptable delay and network overhead.

4.3. Urban VANET Scenario without Network Partition. The proposed protocol is evaluated using urban scenario without network partition. We consider a Manhattan-grid scenario composed of ten-by-ten equally spaced two-lane roads covering an area of 1 Km^2. For more realistic simulation environment, we set 80 m × 80 m obstacles in each grid using Veins framework representing high-rise buildings to attenuate the signal [36]. Unlike the highway scenario, herein, the bit rate is set to 18 Mbits/sec (at MAC layer) along with the transmission power of 0.98 mW, resulting in a transmission range of approx. 200 meters for each vehicle [35].

Once the simulation gets stable, a vehicle (source) located at the centre of the grid initiates the dissemination process by generating and transmitting a data packet of 2048 bytes to all neighbour vehicles of a network. Again, every point of the results is the average of approximately 50 iterations. Since DV-CAST protocol is exclusively designed to operate in highway scenarios, we do not evaluate this protocol over here. Moreover, the evaluation metrics used to assess the DDP4V and other protocols in highway scenario are considered the same to evaluate these protocols in urban scenario. Table 2 covers the summary of key simulation parameters used in the urban scenario. At this end, no signal attenuation due to the high-rise building in urban environments is considered.

Results for all protocols at urban scenario without network partition are summarized in Figure 6. Accordingly, Figure 6(a) presents the coverage (%) against diverse traffic densities. Flooding protocol declares maximum delivery ratio particularly at high traffic densities and delivers the data packet to almost all vehicles in the concerned area. In fact, Flooding protocol retransmits the data packet for all intended vehicles and thus raises the chance of achieving 100% coverage. At low traffic densities, that is, ≤250 vehicles/Km^2, Flooding protocol does not deliver the data packet to all intended vehicles. The rest of the protocols explain similar performance in terms of coverage. Preliminarily, at traffic densities of 200–250 vehicles/Km^2, all protocols deliver the data packet to approximately 80% of the intended vehicles. Finally, at high traffic density, that is, 400 vehicles/Km^2 and above, these protocols achieve 100% coverage. In DDP4V, we believe that vehicles located inside the ideal segment suppress the further transmission of the data packet by other vehicles located inside other segments. Thus, a small part of the coverage area may become uncovered at low traffic densities.

Figure 6(b) shows the total number of transmitted data packets during the dissemination process. Likewise, in highway scenario, Flooding protocol offers the highest overhead due to the simultaneous transmission of the same data packet by all vehicles, resulting in redundant retransmissions. Broadcast suppression mechanism of DDP4V protocol enables it to reduce the number of transmitted data packets, that is, approximately 60%, when compared to Flooding protocol. Also, DDP4V shows less number of disseminated data packets than AID and almost the same as DBRS. Thus DDP4V improves the network performance by avoiding the unnecessary retransmissions and efficient utilization of available bandwidth.

Figure 6(c) presents the average delay to deliver a data packet to all intended vehicles. Notice that Flooding protocol presents an almost constant delay at diverse traffic densities when compared to other protocols. At low traffic densities, 200–300 vehicles/Km^2, DDP4V results in higher delay than Flooding protocol. Conversely, with growing traffic density, DDP4V requires less time for data packets delivery to all intended vehicles. Relatively, DDP4V takes approximately 25% less time to deliver a data packet than AID and DBRS protocols. This confirms that DDP4V is the best solution for such VANET applications, where data packet delivery with strict time requirements is important without incurring in high network load, for example, emergency warning messages. It can be noticed that, in DDP4V protocol, delivery delay decreases with increase in traffic density. This can be explained as follows: whenever traffic density increases in a network, the chances of more vehicles inside the ideal segment also increase. Consequently, vehicles inside the ideal segment transmit the data packet with lowest waiting time, thus decreasing the overall delay during data packet dissemination.

Figure 6(d) shows the average collisions at the MAC layer during the data packet dissemination process. As can be noticed, Flooding protocol experiences the highest number of collisions, continuously increasing with the growing traffic density. Thus, Flooding protocol cannot handle the broadcast storm problem, particularly in dense traffic conditions. Remaining protocols show almost the same performance in terms of a number of collisions. In DDP4V protocol,

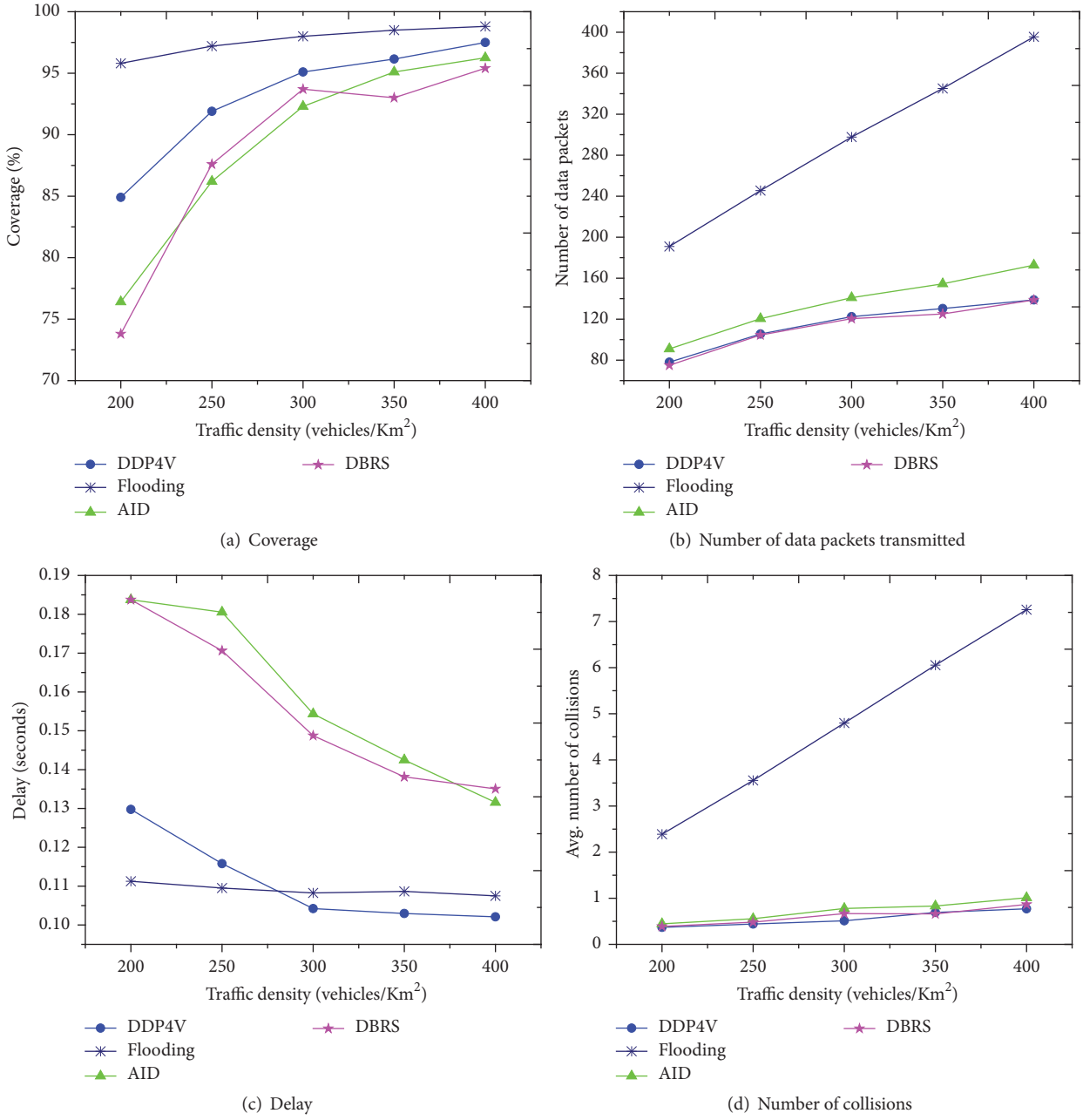


Figure 6: Simulation results for the urban scenario without network partition.

the controlled number of redundant retransmissions reduces the packet collisions as vehicles do not require competing for accessing the channel. These results reveal that DDP4V mitigates the broadcast storm along with data packet delivery to the intended vehicles with acceptable delay.

4.4. Urban VANET Scenario with Network Partitions. The proposed protocol is further evaluated by considering an urban scenario with a network partition inside the concerned area. Hence, we prepare a Manhattan-grid scenario composed of twenty-by-twenty equally spaced two-lane roads covering an area of 4 Km^2. For a network partition inside the concerned area, we insert a large sized obstacle (200 m × 1000 m) aside the source vehicle splitting the concerned area into two horizontal regions. So the vehicles in one region cannot directly communicate with the vehicles in another region. Like the previous scenario, for signal attenuation, obstacles representing high-rise buildings are considered in each grid. Also, the bit rate is set to 18 Mbits/sec with the transmission power of 0.98 mW, resulting in a transmission range of approx. 200 meters for each vehicle.

As stated previously, once the simulation becomes stable, a vehicle (source) located at the centre of the grid initiates the dissemination process by generating and transmitting a data

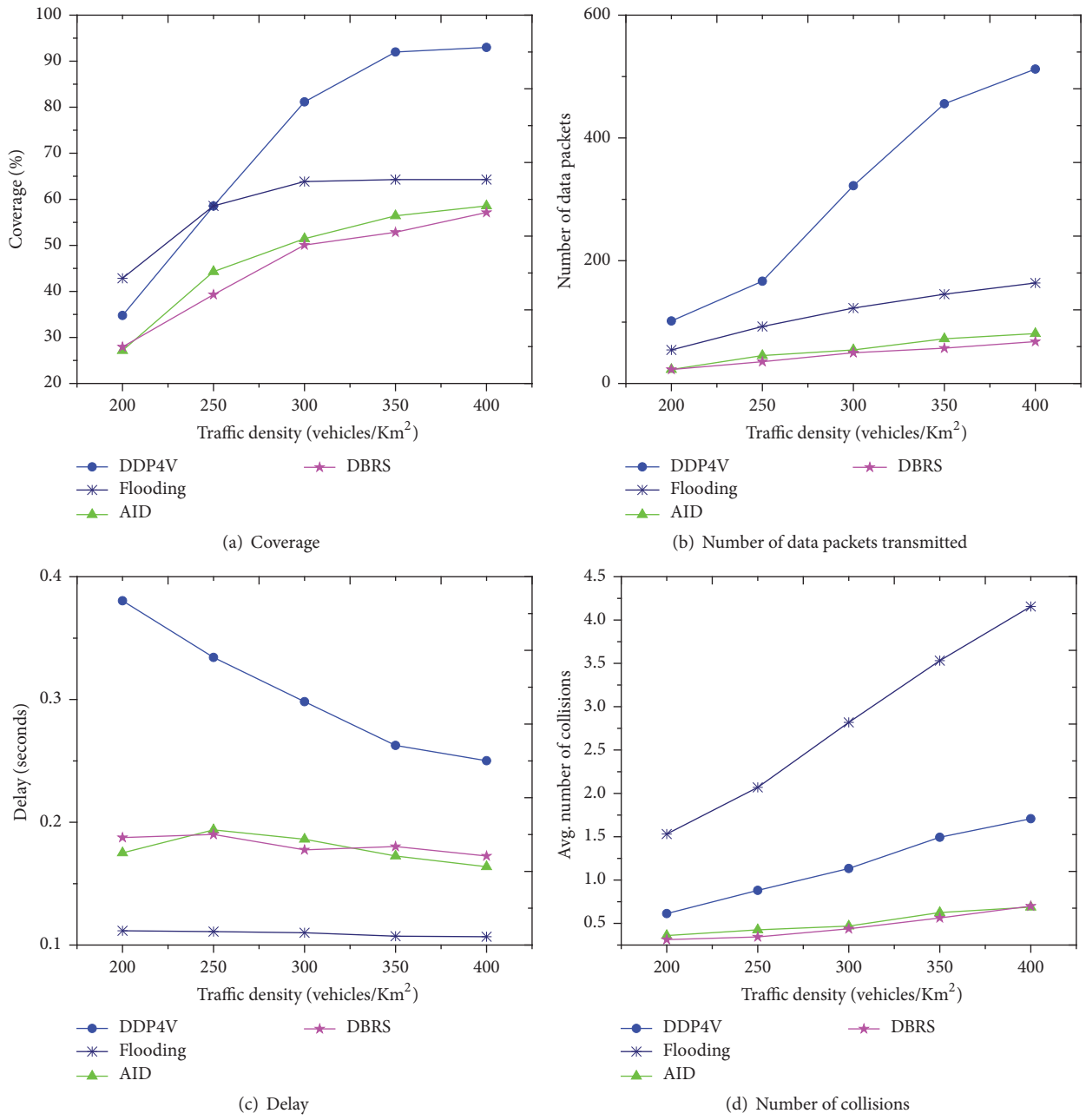


Figure 7: Simulation results for urban scenario with network partition.

packet of 2048 bytes to its neighbours in a network. However, the data packet is disseminated only in one horizontal region of concerned area around the source vehicle due to the network partition. Thus a data packet disseminated in one region of concerned area must be routed through vehicles located outside the concerned area to reach the target vehicles in the other region.

Results for all protocols at the urban scenario with network partitions are summarized in Figure 7. Accordingly, Figure 7(a) presents the coverage (%) for all protocols against diverse traffic densities. At low traffic densities (≤200 vehicles/Km^2), all protocols deliver the data packet to approximately 35–40% of the intended vehicles inside the concerned area. Therefore, it is obvious that, for all protocols, only vehicles located on the same side of the source vehicle (source region) are receiving the transmitted data packets, hence reducing their coverage performance. However, with increasing traffic density, the coverage performance of DDP4V increases as it begins the use of vehicles located outside the concerned area in order to route the data packets from the source region to the partitioned region. At high traffic density, that is, 400 vehicles/Km^2, DDP4V achieves more than 90% coverage, about 30% higher than Flooding protocol and other protocols.

Figure 7(b) shows the total number of transmitted data packets during the dissemination process. It is worth noticing that DDP4V protocol presents the highest overhead which gradually increases with increasing traffic density. In fact, additional data packets are required for message delivery through the network partitions, which directs the DDP4V to face such high overhead. Despite the high overhead of DDP4V, it still does not let the network meet the broadcast storm problem.

Figure 7(c) illustrates average delay to deliver data packets to all intended vehicles. DDP4V presents the highest delivery delay during data dissemination across network partitions. Higher delay for DDP4V is explained by the use of vehicle outside the concerned area, which requires additional time to route data packet across the partitioned regions of a network. Delivery delay of DDP4V varies from 300 to 400 ms under diverse traffic densities which are acceptable in the majority of the VANETs applications. As in the previous scenario, data delivery delay decreases as traffic density increases. This can be explained as follows: whenever traffic density increases in a network, the chances of more vehicles inside the ideal segment also increase. Thus vehicles inside the ideal segment transmit the data packet with lowest waiting time, decreasing the overall delay.

Finally, an average number of collisions at the MAC layer during the data dissemination process are shown in Figure 7(d). Despite the fact that DDP4V shows almost the same performance in terms of packet collisions (little higher) as AID and DBRS, it is still much lower (about half in number) than Flooding protocol. Controlled number of redundant retransmissions in DDP4V reduces packet collisions and enables mitigating the broadcast storm problem along with acceptable delivery delay. Thus, DDP4V is the most appropriate option to disseminate emergency messages in an urban scenario with or without network partitions inside the concerned area.

5. Conclusion and Future Work

In this paper, we proposed a new data dissemination protocol, DDP4V, to overcome the challenging broadcast storm, network partition, intermittently connected network, and optimum next forwarding vehicles (NFVs) selection problems. The proposed protocol shows the potential to provide an efficient data dissemination in diverse VANET scenarios with varying traffic conditions. It presents reasonable performance in three distinct evaluation scenarios: highway scenario and two urban scenarios with and without network partition.

Under dense traffic scenario, DDP4V prefers the vehicle(s) inside the ideal segment of transmission region to retransmit the data packet. It decreases the data packet delivery delay with growing traffic densities in all evaluated VANET scenarios as vehicles inside the ideal segment transmit the data packet with shortest waiting time. Wagon wheel concept assists the DDP4V protocol in selecting the best vehicle as next forwarding vehicle to carry on the dissemination process and mitigate the broadcast storm. DDP4V protocol minimizes the number of redundant transmissions through careful selection of the NFV.

DDP4V presents the best performance in terms of coverage at each simulation scenario under all traffic conditions. It outperforms the other competing protocols and achieves approximately 100% delivery ratio in highway scenario under dense traffic conditions. Even in the partitioned urban scenario, it presents more coverage by 30% when compared to other evaluated protocols. It makes use of vehicles outside the concerned area to guarantee the successful data dissemination through intermittently connected and partitioned network with extended coverage when compared to other evaluated protocols. It also makes use of a timer-based mechanism to decide when the selected vehicle(s) should actually retransmit the data packet. Accordingly, on receiving a new data packet, each vehicle stores and carries this data packet until its time-to-live expires or the vehicle leaves the concerned area, thus increasing the robustness of the protocol.

We build a simulation model and perform a comprehensive performance evaluation of DDP4V protocol in highway and urban VANET scenarios. Simulation results show that it outperforms other related protocols including AID, DBRS, DV-CAST, and Flooding in terms of data packet delivery to all intended vehicles with minimum delay and low overhead. As future work, we are aiming to improve the performance of DDP4V protocol in terms of other important parameters such as the number of duplicate packets, packet propagation, and the number of hops. Moreover, we aim to evaluate our proposed protocol in real city scenarios. Thus, DDP4V is an efficient and reliable option to disseminate emergency messages in highway and urban scenarios with or without network partitions inside the concerned area.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was partially supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korean government (MSIP) (no. 2017-0-00770). It was also supported by the BK21 Plus Project (SW Human Resource Development Program for Supporting Smart Life) funded by the School of Computer Science and Engineering, Kyungpook National University, Ministry of Education, Republic of Korea (21A20131600005).

References

[1] D. Serpanos and M. Wolf, "Industrial Internet of Things," in *in Internet-of-Things (IoT) Systems*, pp. 37–54, Springer, 2018.

[2] R. R. Yager and J. n. P. Espada, *Advances in the Internet of Things*, Springer, New, 2018.

[3] F. K. Shaikh, S. Zeadally, and E. Exposito, "Enabling technologies for green internet of things," *IEEE Systems Journal*, 2015.

[4] T. Toroyan, "Global status report on road safety 2013: supporting a decade of action," in *World Health Organization, Department of Violence and Injury Prevention and Disability*, World Health Organization, Chicago, 2013.

[5] R. Elvik, "Road safety effects of roundabouts: A meta-analysis," *Accident Analysis & Prevention*, vol. 99, pp. 364–371, 2017.

[6] D. Delen, L. Tomak, K. Topuz, and E. Eryarsoy, "Investigating injury severity risk factors in automobile crashes with predictive analytics and sensitivity analysis methods," *Journal of Transport & Health*, vol. 4, pp. 118–131, 2017.

[7] A. Rasheed, S. Gillani, S. Ajmal, and A. Qayyum, "Vehicular ad hoc network (VANET): A survey, challenges, and applications," *Advances in Intelligent Systems and Computing*, vol. 548, pp. 39–51, 2017.

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

[9] L.-W. Chen, Y.-C. Tseng, and K.-Z. Syue, "Surveillance on-the-road: Vehicular tracking and reporting by V2V communications," *Computer Networks*, vol. 67, pp. 154–163, 2014.

[10] B. Cui and X. Yan, "A review of data management and protocols for vehicular networks," *International Journal of Web and Grid Services*, vol. 13, no. 2, pp. 186–206, 2017.

[11] M. Ouyous, O. Zytoune, and D. Aboutajdine, "Multi-channel coordination based MAC protocols in vehicular ad hoc networks (VANETs): A survey," *Lecture Notes in Electrical Engineering*, vol. 397, pp. 81–94, 2017.

[12] O. K. Tonguz, N. Wisitpongphan, and F. Bai, "DV-CAST: a distributed vehicular broadcast protocol for vehicular ad hoc networks," *IEEE Wireless Communications Magazine*, vol. 17, no. 2, pp. 47–57, 2010.

[13] R. S. Schwartz, R. R. R. Barbosa, N. Meratnia, G. Heijenk, and H. Scholten, "A Simple and Robust Dissemination protocol for VANETs," in *Proceedings of the 2010 European Wireless Conference, EW 2010*, pp. 214–222, Italy, April 2010.

[14] L. A. Villas, T. P. C. De Andrade, and N. L. S. Da Fonseca, "An efficient and robust protocol to disseminate data in highway environments with different traffic conditions," in *Proceedings of the 19th IEEE Symposium on Computers and Communications, ISCC 2014*, Portugal, June 2014.

[15] M. Chaqfeh and A. Lakas, "A novel approach for scalable multi-hop data dissemination in vehicular ad hoc networks," *Ad Hoc Networks*, vol. 37, pp. 228–239, 2016.

[16] I. Achour, T. Bejaoui, A. Busson, and S. Tabbane, "SEAD: A simple and efficient adaptive data dissemination protocol in vehicular ad-hoc networks," *Wireless Networks*, vol. 22, no. 5, pp. 1673–1683, 2016.

[17] M. Durresi, A. Durresi, and L. Barolli, "Emergency broadcast protocol for inter-vehicle communications," in *Proceedings of the 11th International Conference on Parallel and Distributed Systems Workshops, ICPADS 2005*, pp. 402–406, Japan, July 2005.

[18] S. Ucar, S. Coleri Ergen, and O. Ozkasap, "Multi-hop cluster based IEEE 802.11p and LTE hybrid architecture for VANET safety message dissemination," *IEEE Transactions on Vehicular Technology*, 2015.

[19] S. Panichpapiboon and G. Ferrari, "Irresponsible forwarding," in *Proceedings of the 2008 8th International Conference on ITS Telecommunications (ITST)*, pp. 311–316, Phuket, Thailand, October 2008.

[20] S. Busanelli, G. Ferrari, and S. Panichpapiboon, *Cluster-based irresponsible forwarding, in The Internet of Things*, Springer, in The Internet of Things, 2010.

[21] N. Wisitpongphan, O. K. Tonguz, J. S. Parikh, P. Mudalige, F. Bai, and V. Sadekar, "Broadcast storm mitigation techniques in vehicular ad hoc networks," *IEEE Wireless Communications Magazine*, vol. 14, no. 6, pp. 84–94, 2007.

[22] Y.-S. Chen, Y.-W. Lin, and S.-L. Lee, "A mobicast routing protocol in vehicular ad-hoc networks," *Mobile Networks and Applications*, vol. 15, no. 1, pp. 20–35, 2010.

[23] W. Wang, T. Luo, and Y. Hu, "An Adaptive Information Quantity-Based Broadcast Protocol for Safety Services in VANET," *Mobile Information Systems*, vol. 2016, Article ID 2195496, 2016.

[24] G. Korkmaz, F. Özgüner, E. Ekici, and Ü. Özgüner, "Urban multi-hop broadcast protocol for inter-vehicle communication systems," in *Proceedings of the VANET - Proceedings of the First ACM International Workshop on Vehicular Ad Hoc Networks, Held in Conjunction with MOBICOM 2004*, pp. 76–85, usa, October 2004.

[25] G. Korkmaz, E. Ekici, and F. Özgüner, "An efficient fully ad-hoc multi-hop broadcast protocol for inter-vehicular communication systems," in *Proceedings of the 2006 IEEE International Conference on Communications, ICC 2006*, pp. 423–428, Turkey, July 2006.

[26] T. Kim, W. Hong, H. Kim, and Y. Lee, "An Effective Data Dissemination in Vehicular Ad-Hoc Network," in *Information Networking. Towards Ubiquitous Networking and Services*, vol. 5200 of *Lecture Notes in Computer Science*, pp. 295–304, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[27] M. Bakhouya, J. Gaber, and P. Lorenz, "An adaptive approach for information dissemination in Vehicular Ad hoc Networks," *Journal of Network and Computer Applications*, vol. 34, no. 6, pp. 1971–1978, 2011.

[28] G. Maia, A. Boukerche, A. L. L. Aquino, A. C. Viana, and A. A. F. Loureiro, "A data dissemination protocol for urban Vehicular Ad hoc Networks with extreme traffic conditions," in *Proceedings of the 2013 IEEE International Conference on Communications, ICC 2013*, pp. 5997–6001, Hungary, June 2013.

[29] S. Allani, T. Yeferny, R. Chbeir, and S. B. Yahia, "DPMS: A Swift Data Dissemination Protocol Based on Map Splitting," in *Proceedings of the 2016 IEEE 40th Annual Computer Software and Applications Conference, COMPSAC 2016*, pp. 817–822, USA, June 2016.

[30] S. Iadicicco, S. Infante, P. Salvo, A. Baiocchi, and F. Cuomo, "Multi-originator data dissemination in VANETs," in *Proceedings of the 12th Annual Conference on Wireless On-Demand Network Systems and Services, WONS 2016*, pp. 49–56, ita, January 2016.

[31] K. Liu et al., "Network-coding-assisted data dissemination via cooperative vehicle-to-vehicle/-infrastructure communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1509–1520, 2016.

[32] A. s. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proceedings of the in Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems workshops*, 2008.

[33] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, 2011.

[34] M. Behrisch et al., "SUMO-simulation of urban mobility: an overview," in *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*, 2011.

[35] C. Sommer, S. Joerer, and F. Dressler, "On the applicability of Two-Ray path loss models for vehicular network simulation," in

Proceedings of the IEEE Vehicular Networking Conference (VNC '12), pp. 64–69, IEEE, Seoul, Republic of Korea, November 2012.

[36] S. Grafling, P. Mähönen, and J. Riihijärvi, "Performance evaluation of IEEE 1609 WAVE and IEEE 802.11p for vehicular communications," in *Proceedings of the 2nd International Conference on Ubiquitous and Future Networks (ICUFN '10)*, pp. 344–348, June 2010.