Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 6618168, 11 pages
https://doi.org/10.1155/2020/6618168

Research Article

Formal Verification on the Safety of Internet of Vehicles Based on TPN and *Z*

Yang Liu,[1] **Liyuan Huang**,[1] **and Jingwei Chen**[2]

[1]*Information Science and Engineering, Chongqing Jiaotong University, Chongqing 400074, China*
[2]*Chongqing Key Laboratory of Automated Reasoning and Cognition, Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China*

Correspondence should be addressed to Jingwei Chen; chenjingwei@cigit.ac.cn

Received 2 November 2020; Revised 30 November 2020; Accepted 3 December 2020; Published 29 December 2020

Academic Editor: Yong Chen

Nowadays, the Internet of Vehicles has become the focus of global technological innovation and transformation in the automotive industry. Its flow modelling appears to play a very important role for designing and controlling the transportation systems, since it is not only necessary for improving safety and transportation efficiency but also can yield a series of society, economy, and ecosystem environment problems. Considering the characteristics of the frame structure includes states and actions and discrete and continuous aspects of traffic flow dynamics, both petri net and *Z* have proved to be useful tools for modelling the Internet of Vehicles. It can formally describe the vehicle behavior accurately with petri net and more details with *Z* frame structure. A new integration formal method of time petri net and *Z* is presented in this paper for modelling the vehicle behaviors and traffic rules through taking into account state dependencies on external rules. Moreover, a case study in the Internet of Vehicles is proposed to deal with the accurate localization of events. It shows that this formal verification methods significantly improves the safety and intelligence of the Internet of Vehicles.

1. Introduction

With the development of communication technology, wireless sensing technology, automatics, artificial intelligence, and so on, the Internet of Vehicles techniques come out. It is the achievements combined with the latest technological of computers and the modern automobile industry. Because of the complex and dynamic environment when it is working, the control system becomes more and more complex. Since it is about life, the key safety factor, such as automotive engine, air bag control, brake system, sensor monitoring system, and traffic regulations, have very strict reliability requirements. Internet of Vehicles has made our life convenient; nevertheless, at the same time, accidents still happen often. Many researchers ensure the safety from different aspects [1–3] by different methods, such as control strategy, security factor, and intelligent platform. More and more experiences show that the formal method is very effective to ensure the safety of the Internet of Vehicles [4–7] systems.

In fact, the formal method is a good way to inspect the problems in system design or requirement design [8, 9]. The running environment of the Internet of Vehicles is very complex and changes dynamically. It is hard to describe the Internet of Vehicle using only one single formal language. The traditional process analysis methods, such as Petri nets [10], CCS (Calculus of Communicating Systems) [11, 12], and CSP (Communication Sequential Processes) [13, 14], can model different aspects of the system from different angles and abstractions, but the powers of description for functional and nonfunctional attribute and constraint condition are deficient. The traditional model languages such as V [15, 16], B [17], and Z [18, 20] are good at modelling description, but poor at describing system concurrency. At present, the integrated specification languages are a hot topic, which produced CSPZ [21], TCOZ [22], PZN [23, 24], and so on. However, it seems that these languages do not aim at the Internet of Vehicles. PZN has a good advantage in describing traditional systems, since specification Z has a good frame structure both in state description

and operation description, and Petri nets [25–28] are very suitable to express the behavior of the parallel and concurrent system model. So, the hybrid methodology which combines the advantages of both specification Z and Petri nets is very suitable for modelling and analyzing the Internet of Vehicles system. PZN has been used to model and analyze validity and accessibility of networked software. Experimental results showed that PZN is very suitable to apply in it. In the Internet of Vehicles circumstance, except states and operation, time constraint is also very important. It not only has continuous part time but also has discrete time. Some researchers used time Petri nets to model the requirements and software of system [29–34], but it lacked specific rule descriptions and state depictions.

Motivated by the previous experience in formal verification of requirements modelling and analyzing of networked software, in this paper, TPZN (integration Time Petri Net and Z) is proposed to formal modelling and verifying the Internet of Vehicles systems. It is able to describe the concurrent process and fore-and-aft states in systems at different times. TPZN consists of two parts TPZN-TPN and TPZN-Z. TPZN-TPN defines the data flow of the whole structure, order, and behavior of process at one moment, while, TPZN-Z depicts the abstract data frame, specific rule restriction, and time constraint. So, based on enhancing the abstraction of the data and refinements by Z, the number of states of the Time Petri Nets can be decreased effectively. A case study shows the modelling method in detail. This formal method is proved greatly by improving the safety and validity of the intelligent vehicle systems.

2. Background

In this section, we recall some preliminary backgrounds that are necessary for the rest of the paper.

2.1. Hybrid Petri Net Extension. Hybrid petri net extension for traffic road modelling is proposed by Riouali et al. in [7]. It brought discrete parts and continuous parts which include discrete and continuous places and transitions. The moving and evolution of the Internet of Vehicles depend on the state of places and are governed by various function, namely, creation, destruction, merging, and splitting; meanwhile, it defined the speed, maximum density, length, and maximum flow of the traffic road modelling.

A hybrid petri net consists of three kinds of objects: places, transitions, and directed arcs. However, unlike the traditional petri net, here places are divided into two kinds: discrete places and continuous places. Transitions as well as places also fall into discrete transitions and continuous transitions. Arcs still show the state dynamic from places to transitions or from transitions to places. Hybrid petri net extension is defined 6-tuplet $N = (P, T, \text{Pre}, \text{Post}, \Upsilon, \text{Time})$.

(1) P is a set of places, $P = \text{Pc} \cup \text{Pd}$, where Pc represents continuous places and Pd represents discrete places.

(2) T is a set of transitions.

(3) Pre is the backward incidence matrix $P \times T \longrightarrow N$.

(4) Post is the forward incidence matrix $T \times P \longrightarrow N$.

(5) γ represents the batch place function. It associates with each batch place 4-tuplet (Vi: speed; di: a maximum density; Si: length; $\Phi^{\max}$: a maximum flow).

(6) Time represents the firing delay in case of continuous or batch transitions.

Here, we consider the time factor, while the γ is more suitable to be used in more intelligent vehicle concurrent environment.

2.2. Z Frame Structure. Z is a good formalism for modelling and designing. Compared with Petri Net, Z has better abilities in type definition and data abstraction and model refining. Its basic frame contains states and operations as Figure 1. Every operation has relative states and constrain rules. However, it does not describe the dynamic behavior of the systems.

Although Ding et al. and Wei et al. proposed a method that models systems by both Z and Petri Nets in [27, 28] and the authors also showed that using PZN (Z and Petri Nets) to model the requirements of software is an effective and feasible way [9], it is still not good enough to model the Internet of Vehicles. The reason is that PZN does not have the ability to describe the real-time performance which is very important in vehicle systems. In transportation systems, time is a very important factor. So, all previous works have to be improved and time constraints will be added in PZN [9]. TPZN stands for the integration of PZN and time factor. In Section 3, we will introduce the modelling and analysis methods by TPZN.

3. Modelling with TPZN

For satisfying the real-time capability and dynamic evolution and data abstraction and type definition capabilities of the Internet of Vehicles, the integrated specification TPZN is presented in this paper. Based on enhancing the abstraction of the data and refinements by Z, the state-of-the-time Petri Nets can be decreased effectively. Compared with time petri nets, color petri nets, PZN, and CSPZ, TPZN is more suitable to define the intelligent vehicle systems.

3.1. TPZN

Definition 1. A TPZN is a tuple $< P, T, F, Zp, Z_T, S, C, M_0, \text{SI}>$, where

(1) P is a set of the states.

(2) T is a set of the transitions.

(3) F is a set of the arcs which links state and transition.

(4) $N = (P, T, F)$ is a SISO net.

(5) $\text{TPN} = (P, T, F, M_0, \text{SI})$ is like a traditional time petri net.

(6) $\text{PZN} = (P, T, F, Zp, Z_T, S, C)$ is a PZN as in [9, 19].

(7) Zp is a set of the state frame based on Z.

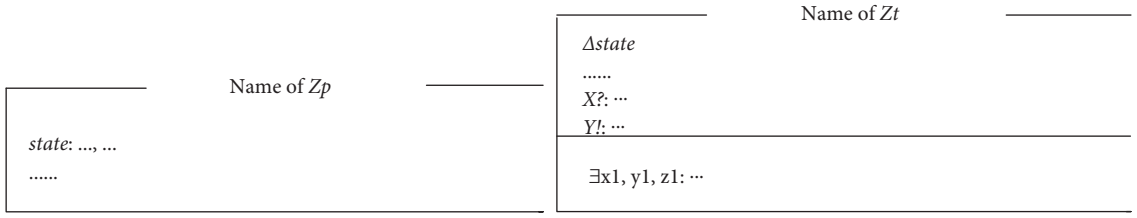

Figure 1: Frame structure of Z.

(8) Z_T is a set of the operation frame based on Z.

(9) S: $P \longrightarrow Zp$ is a set of the one-to-one map relationship between P and Zp.

(10) C: $T \longrightarrow Z_T$ is a set of the one-to-one map relationship between T and Z_T.

(11) M_0: is the initial mark, and $\exists t \in T, (p_0, t) \in F, M_0[t>$.

(12) $\exists \omega, \omega \in L(\text{TPN})$, φ_f $(\text{TPN}, \omega) = (M_f, D_f, \text{SI}_f)$, $M_0 = p_i + p_j + \cdots + p_k$, $D_0 = \{D_0(t_m), D_0(t_n), \ldots\}$, $\text{SI}_0 = [0, 0]$, $p_i, p_j, \ldots, p_k$ are all trigger states in the beginning and $t_m, t_{n,\ldots}$ are all trigger transitions. M_f represents the state of every node device in one time. D_f represents a set of the time interval of the next possible transition. SI_f represents the time interval of the system may need when it arrives M_f. φ_f represents the system's situation during time interval-SI_f. If M_f is the final state, $D_f = \varnothing$.

To ensure the compatibility and validity of the design, TPZN-Z frame is used to describe the sign, property, rules, and so on. The corresponding relation of TPN and Z is shown in Figure 2. The green dashed box is the precondition of transition. The rules and constraints are formally described by Z in Z_t. The purple dashed box represents the postcondition by Z.

3.2. Time Constrained in TPZN. This paper introduces global time and relative time for TPZN. The global time proves the standard system time, and the relative time supplies the time relative to previous status M_i. Here, it needs to define two variables. One is the earliest occurrence time, EAR(t), the other one is the latest occurrence time, $LAT(t)$. SI_i contains the earliest occurrence time EAR(t_i) and the latest occurrence time $\text{LAT}(t_i)$. $\text{SI}_i = [\text{EAR}(t_i), \text{LAT}(t_i)]$. $D_i(t)$ is the relative time to M_{i-1}, $M_{i-1}[t_i>$.

For example, in Figure 3, relative time is marked. For example, "$t7$ [15, 25]" means that $t7$ can be triggered in 15 seconds at least and 25 seconds at most. If it exceeds 25 seconds, the automatic delivery truck will stop working. Accordingly, the system will be warning. The global time is always synchronized with the time of the system.

3.3. Model Refining. The environment of the Internet of Vehicles running is always complex, dynamic, and unexpected so that model refining and topological evolution capability is to be very important. Suppose $\text{TPZN}_{11}$and TPZN_{12} are the subnet of TPZN_1:

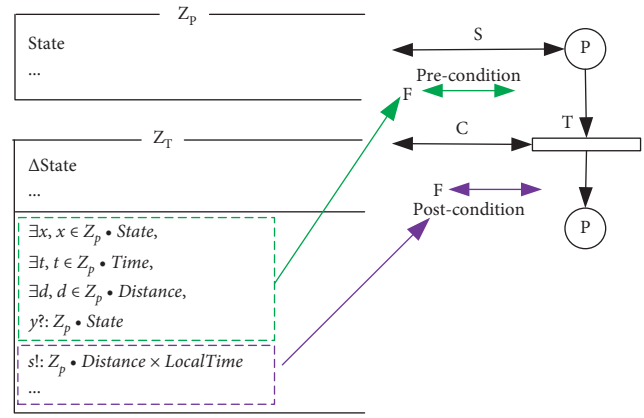

Figure 2: The relation between TPN and Z in TPZN.

$$\begin{aligned}\text{TPZN}_{11} &= \langle P_{11}, T_{11}, F_{11}, Zp_{11}, Z_{T11}, S_{11}, C_{11}, M_{011}, \text{SI}_{11}\rangle,\\ \text{TPZN}_{12} &= \langle P_{12}, T_{12}, F_{12}, Zp_{12}, Z_{T12}, S_{12}, C_{12}, M_{012}, \text{SI}_{12}\rangle.\end{aligned} \tag{1}$$

Then, $(\text{TPZN}_{11} \cap \text{TPZN}_{12}) \subset \text{TPZN}_1$. $\forall p_i, p_i \in P$, $P \in \text{TPZN}_{11}/(\text{TPZN}_{11} \cap \text{TPZN}_{12})$ are all the new additional virtual states which represent the possible states before or after the subnet TPZN_{11}. $\forall t_i, t_i \in T$, $T \in \text{TPZN}_{11}/$ $(\text{TPZN}_{11} \cap \text{TPZN}_{12})$ are all the new additional virtual transitions which represent the possible preconditions or postconditions. Of course, new Z frame structure Z_p' and Z_t' should be redefined by additional rules. In the similar way, a new TPZN′ can substitute a transition t_i, when the control structure change.

On the contrary, when one model is needed to be abstracted, it can be seen as a new transition t'; then adding its precondition and postcondition and reserving input and output are relative to the conterminal model.

Theorem 1. *If the global execution time of every transition sequence of the new refined TPZN model from the beginning to the end is equal to the execution time of the substituted t_i of the original TPZN, the new refined TPZN can maintain behavioral consistency with the original one.*

Because TPZN integrates TPN and Z, the refined TPN can maintain behavioral consistency with the original one and has been proved in [35–37].

4. Modelling Analysis

4.1. Accessibility. Traditionally speaking, there are two ways to analysis the accessibility of the model. One way is using

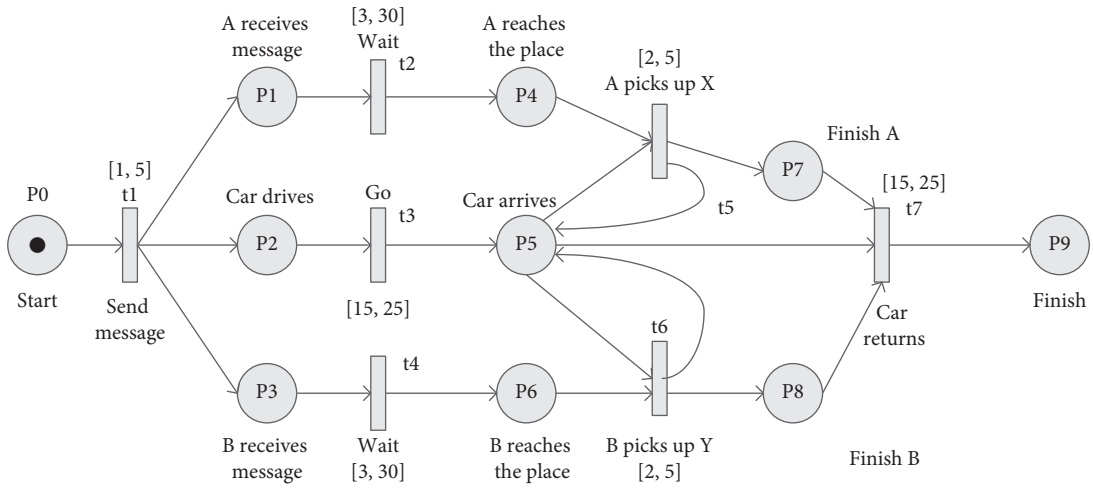

Figure 3: The TPZN of automatic delivery truck.

the reachability tree which is used to analysis the accessibility of model states. Because the accessibility of the TPZN involves limited time and there are lots of the state classes, some methods to reduce the state classes are necessary. For instance, Bourdil and Berthomieu have proposed some methods to reduce the state classes [28, 31]. Based on their work, we use Z frame to abstract the system so to reduce the state number. The layer can be subdivided into smaller layers. If the lowest layers can be verified to be correct, accessible, and safe, the whole upper layer will have the same character. The reachability tree can be built by φ_f based on TPZN. From φ_{fi} to φ_{fj}, the path from the node φ_{fi} of the tree to the node φ_{fj} shows the transition sequence (Figure 4).

The other way is using the incidence matrix marked $C(C = D^+ - D^-)$. Here, the output matrix-D^+ is defined as

$$D^+[i,j] = \begin{cases} 0, & \nexists f_k = (t_i, p_j), f_k \in T \times P, \\ n, & \exists f_k = (t_i, p_j), f_k \in T \times P \wedge \text{Token}_{Pj} = n, \end{cases} \tag{2}$$

where $D^+[i, j] = 0$ means there does not exist an arc from the t_i to p_j. While, $D^+[i, j] = n$ means that there is an arc from the t_i to p_j, and it will produce n same type elements with the transfer. The (i, j) entry of D^- is defined as

$$D^-[i,j] = \begin{cases} 0, & \nexists f_k = (p_i, t_j), f_k \in P \times T, \\ n, & \exists f_k = (p_i, t_j, f_k \in P \times T \wedge \text{Token}_{Pi} = n] > t_j, \end{cases} \tag{3}$$

where $D^-[i, j] = 0$ means there is not an arc from the p_i to t_j, while $D^-[i, j] = n$ means that there is an arc from the p_i to t_j and the transition can happen only if there is n same type elements in the p_j.

Supposing M_i is a marked state. From M_i to M_j, if there is an transition sequences $\sigma = t_i t_{i+1}, \ldots, t_j$ marked by X-vector quantity and it satisfies $M_j = M_i + X \bullet (D^+ - D^-)$, it proves that the Mi state is accessibility. However, in TPZN, it must consider the limited time. The time constrained rules are described by Z frame. In the automotive vehicles system,

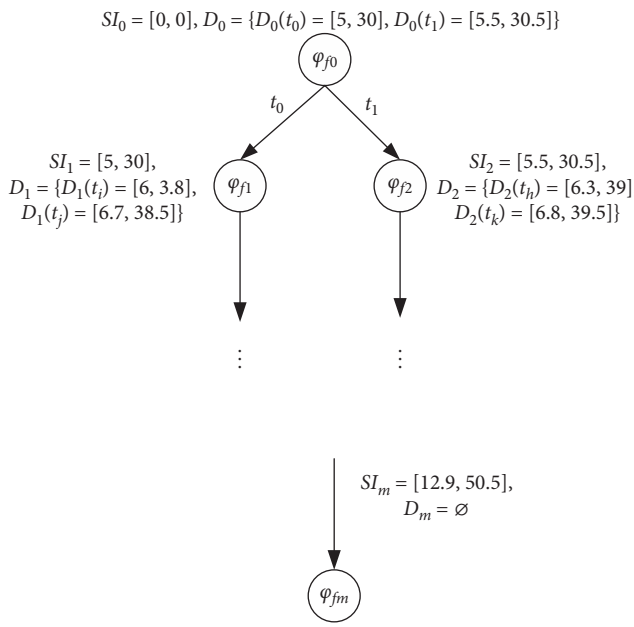

Figure 4: The TPZN of automatic delivery truck.

time constrained rules must be built strictly because subtle time change may cause serious traffic accident. So, modelling the vehicles' system, it needs to abstract the whole system, then subdivide the whole system into specific layers, and go on subdividing until it is subdivided into atom modules. By φ_f which represents the state class containing timestamp, we can get the possible behavior information of the system in certain time interval and then predict the next step. The algorithm of accessibility is designed as Algorithm 1 which shows the accessibility decision from Mi to Mj, and the case study explains how to use it in Figure 5.

4.2. Validity. The validity of the control structure can be analyzed by the transfer matrix L_{DP} of TPZN. From the L_{DP}, concurrent transition can be obtained by the same column and row. As the following in L_{DP1}, t_1 and t_2 can be trigger

simultaneously from p_0 to p_1 and p_e, while, if p_1 is arrived, t_1 and t_3 must be triggered:

$$L_{DP1} = \begin{array}{c} \\ p_0 \\ p_1 \\ \vdots \\ p_e \end{array} \begin{array}{c} \begin{array}{cccc} p_0 & p_1 & \cdots & p_e \end{array} \\ \begin{bmatrix} 0 & t_1 & \cdots & t_2 \\ 0 & t_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}. \quad (4)$$

So, the data flow structure can be mapped into the transfer matrix L_{DP}. If there exist several transitions in the same row p_i, it means when the system arrives into the state p_i, these transitions will be simultaneously triggered. While if there exist several transitions in the same column p_i, it means only under the condition that all the transitions are triggered, and p_i can be reached.

After getting the initial model and parameters, the sampled data or historical data can be used to correct the model and parameters. Of course, real time data also can be used to modify the model and parameters, but more often, it is used to predict possible state of the future.

The process of modelling the Internet of Vehicles with TPZN is as Figure 6. First, the node device information and traffic rules and evaluation indicators are obtained from the initial system model. Meanwhile, the data flow structure of the system should be obtained, and divide the initial system into subsystem. Second, the foregoing information is described by Z frame structures, and the latter is described by TPN. Third, the subsystem should be refined. Then, the whole system can be formally modelled by TPZN. Next, the related parameters such as L_{DP}, φ_f, D^+, and D^- can be obtained from the TPZN model. Combined with the current information of the system, the initial parameters are used to analyze the character. At last, the future behavior of the vehicle system can be predicted. If the prediction shows, it will be in danger, and some strategies can be adopted. If the danger is caused by some traffic rules, these rules will be modified.

4.3. Advantage. Compared with TPN, PZN, and Z, TPZN has better dynamic structure and more convenient time constraint which are very important to the Internet of Vehicles. Except these, TPZN has better frame structure which can abstract the system to reduce the number of the states to avoid the explosive growth usually happened in traditional Petri Net. So, the advantage of modelling with TPZN is shown very clearly in Table 1.

5. A Case Study

To verify effectiveness of our modelling methods to analyze our verification algorithms, in this section, a simple case study is offered. Suppose that an intelligent car has 4 lidars, 4 radars, 4 side vision, 1 full vision, image processing system, radar system, lidar system, brake system, and so on. It is running on the straight road, as shown in Figure 7.

For modelling the system, the first step is to obtain the Z frame structure of every node device. Here, parts of the system model's, such as Z_p and Z_t, are put forward as space is limited.

CAR

Number: *number*
Brand: *Volk*, *Ford*, *Benz*, *BMW*, ···
Fuel: *Gasoline*, *Electric*, ···
FuelState: *full*, *over*, *normal*
Lidar: *FrontLeftLi*, *FrontRightLi*, *FrontMiddleLi*, *BackLeftLi*, *BackRightLi*
Radar: *FrontLeftRa*, *FrontRightRa*, *BackLeftRa*, *BackRightRa*
Vision: *FrontLeftVi*, *FrontRightVi*, *BackLeftVi*, *BackRightVi*, *FullVi*
ProcessSystem: *RadarSystem*, *LidarSystem*, *VisionSystem*, *BrakeSystem*, ···
.....
State: *Start*, *Stop*, *Brake*, *Acceleration*, *Deceleration*, *Back*, *TurnLeft*, *TurnRight*, ···

The above frame is the same parts of one element of the Z_p, which is defined as one kind of state of the system. As the blue dashed box shows, it formally defines relative devices. The following one defines one node device of the system.

FrontLeftLi

Name: *Lidar*
Time: *GlobalTime*, *LocalTime*
Distance: *LongDistance*, *LimitDistance*, *SafeDistance*
Speed: *Distance* X *LocalTime*, *ConstrainSpeed*
*StateLi*1: *Work*, *Rest*
.....

The next frame is one element of the Z_t which defines one kind of possible transition of the system.

BEGIN

Δ CAR
Δ FrontLeftLi
Δ FrontRightLi
Δ FrontMiddleLi
Δ BackLeftLi
Δ BackRightLi
Δ FrontLeftRa
Δ Door
.....
x?: *CAR.State*
x1!: *FrontLeftLi.StateLi*1
x2!: *FrontRightLi.StateLi*2
x3!: *FrontMiddleLi.StateLi*3
x4!: *BackLeftLi.StateLi*4
x5!: *BackRightLi.StateLi*5
z!: *Door.StateDoor*
.....

∃*n*: *CAR.Number*, ∃*y*: *CAR.FuelState*···
((*n* ∈ *number*) ∧ (*x*? ∈ *Start*) ∧ *y* ∉ *over* ∧ ···)
→(*x*1! = 1) ∧ (*x*2! = 1) ∧ (*x*3! = 1) ∧ (*x*4! = 1) ∧ (*x*5! = 1) ∧ (*z*! = {1, 1, 1, 1})···
...

So, at the first step, every node device's Z frame structure and every transition can be defined. In second step, the TPN model of the Internet of Vehicles system will be built. Parts of the TPN model are shown in Figure 8.

Then, the TPZN of this case is <P, T, F, Zp, Z_T, S, C, M_0, SI>, where

(1) $P = \{p_0, p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}\}$.

(2) $T = \{t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9\}$.

Input: $\varphi_{\mathbf{f}} = \{\varphi_{\mathbf{f0}}, \varphi_{\mathbf{f1}}, \varphi_{\mathbf{f2}}, \ldots, \varphi_{\mathbf{fe}}\}$, M_i, M_j, D^+, D^-
Output: true (print the path); false
Find the X, $X = (\mathbf{M_j} - \mathbf{M_i}) \bullet (\mathbf{D}^+ - \mathbf{D}^-)^{-1}$
If X not exist, return false;
  Else
    For ($k = 0$; $k < n$; k++)
      $\sigma_{\mathbf{k}} = \mathbf{t_h}, \mathbf{t_{h+1}}, \ldots, \mathbf{t_{h+c}}$;//$\sigma_{\mathbf{k}}$ store the different value of X, n is the number of X.
$\varphi_{\mathbf{f0}}$ is the root node;//built the reachability tree
For ($k = 1$; $k \le e$; k++)
    {if ($\exists \mathbf{t_m}, \mathbf{t_m} \in \mathbf{T}$, $\mathbf{M_k}] > \mathbf{t_m}$) $\wedge$ ($[\mathbf{SI_k} \cdot \mathbf{EAR}(\mathbf{t_k}), \mathbf{SI_k} \cdot \mathbf{LAR}(\mathbf{t_k})] \subseteq \{\mathbf{system}(\mathbf{t}) + \mathbf{interval\ time}(\mathbf{t_k})\}$
    $\varphi_{\mathbf{fk}}$ is the child node of $\varphi_{\mathbf{f(k-1)}}$;
    }//test the time constrain
For($k = 0$; $k < n$; k++)
  {If ($\sigma_{\mathbf{k}} = \mathbf{t_h}, \mathbf{t_{h+1}}, \ldots, \mathbf{t_{h+c}}$) exist in one path of $\varphi_{\mathbf{fi}}$ to $\varphi_{\mathbf{fj}}$,
    Lookup(S, C); //find the relative $\mathbf{Z}'_{\mathbf{p}}$ and $\mathbf{Z}'_{\mathbf{T}}$, test the logical relationship
    If the logical relationship from $\mathbf{Z_{pa}}, \mathbf{Z_{pb}}, \ldots, \mathbf{Z_{p\,d}}$ $((\mathbf{M_i} = \mathbf{P_a} + \mathbf{P_b} + \cdots + \mathbf{P_d}), \mathbf{Z_{pa}}, \mathbf{Z_{pb}}, \ldots, \mathbf{Z_{p\,d}} \in \mathbf{Z}'_{\mathbf{p}})$ to $\mathbf{Z_{pe}}, \mathbf{Z_{pf}}, \ldots, \mathbf{Z_{pr}}$ $((\mathbf{M_j} = \mathbf{Z_{pe}} + \mathbf{Z_{pf}} + \cdots + \mathbf{Z_{pr}}), \mathbf{Z_{pe}}, \mathbf{Z_{pf}}, \ldots, \mathbf{Z_{pr}} \in \mathbf{Z}'_{\mathbf{p}})$ is reasoned to be correct.
      Print $\varphi_{\mathbf{fi}}, \mathbf{t_n}, \varphi_{\mathbf{fi+1}}, \mathbf{t_{n+1}}, \ldots, \mathbf{t_{n+c}}, \varphi_{\mathbf{fj}}$;
  }

Algorithm 1: The algorithm of accessibility. Accessibility decision from M_i to M_j.

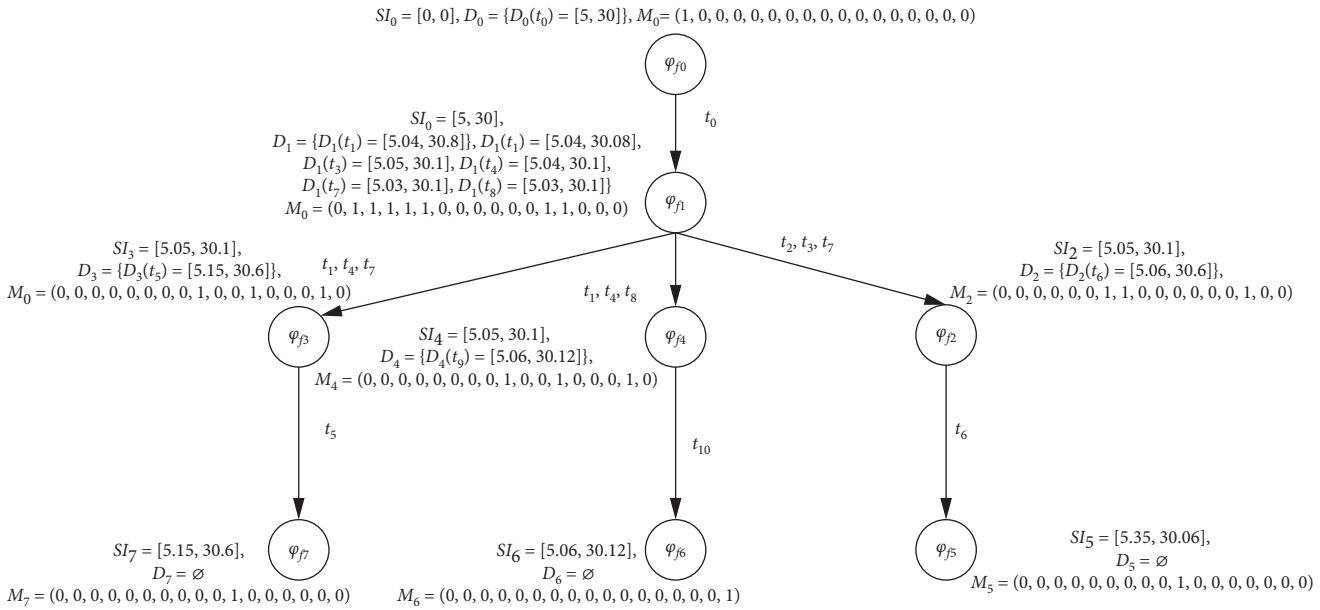

Figure 5: Reachability tree of the case study.

(3) F is the set of arcs in Figure 8. The elements are like the following form (p_0, t_0), (t_0, p_1), (t_0, p_2), (t_0, p_3), (t_0, p_4), (t_0, p_5), (p_1, t_1), ….

(4) Z_{pi} is the element of the set of Z_p, and it represents the state of Z frame of the node devices as CAR and FrontLeftLi.

(5) Z_{ti} is the element of the set of Z_T, and it represents the transition of Z frame of the system as BEGIN.

(6) S maps the relationship from state pi to Z frame of the state, as p_0- > CAR.

(7) C maps the relationship from transition ti to Z frame of the transition, as t_0- > BEGIN.

(8) M_0=(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0) represents the initial condition of the system.

SI_i is shown in Table 2, which represents the temporal interval under M_i. Some of the details of each p_i and t_i are shown as Table 3. Figure 7 shows parts of the case study, so, the p_9 and p_{10} are not the real final states. In fact, p_9 and p_{10} can turn into normal state by some steps.

From Figure 8, the final state classes are φ_{f5}, φ_{f6}, and φ_{f7}, where φ_{f5} is the emergency brake, $\varphi_{f7}$is slow

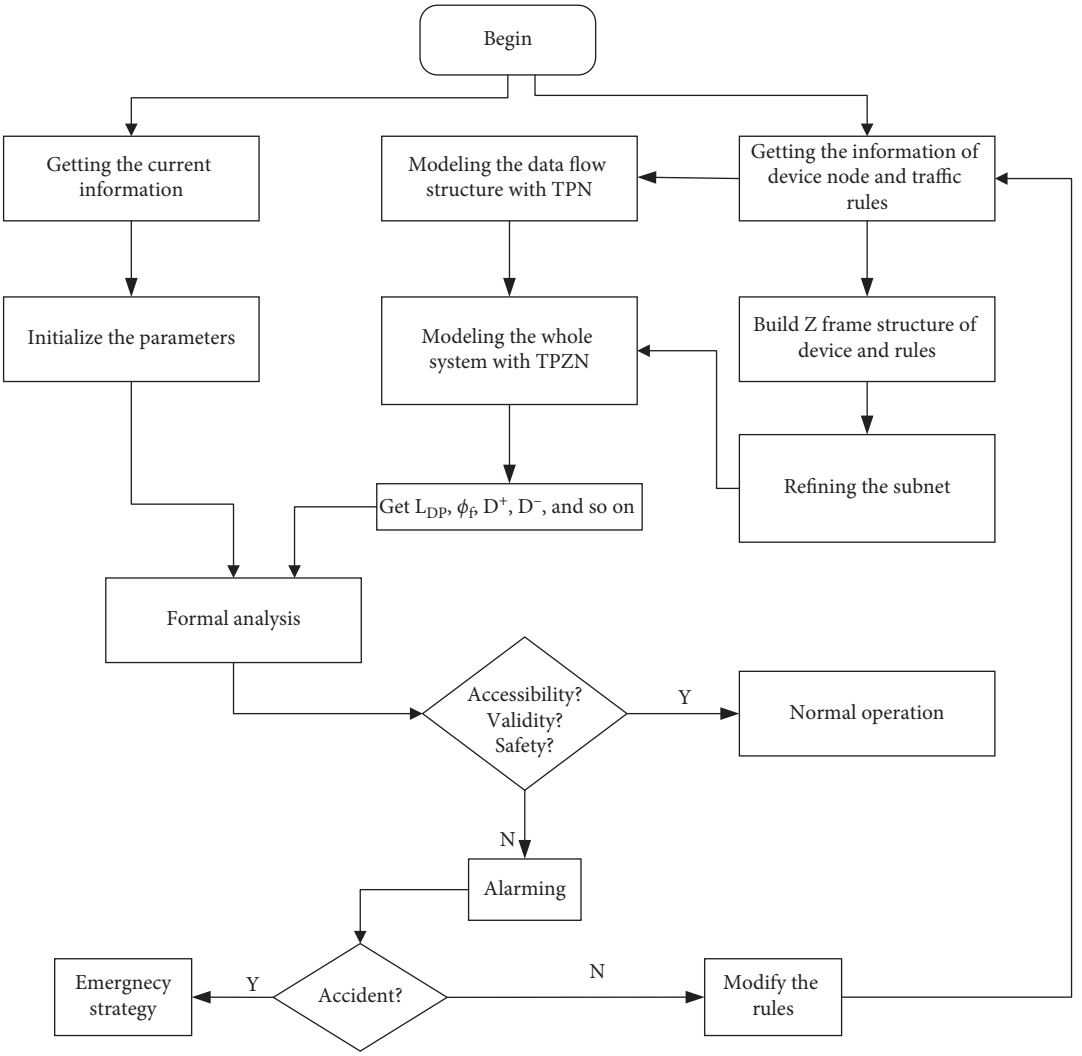

Figure 6: The flow diagram of modelling with TPZN.

Table 1: Compared TPZN, TPN, PZN, and *Z*.

	Dynamic structure	Frame structure	Number of states	Time constraint
TPZN	Good	Good	Abstract	Good
TPN	Good	Not good	Explosive growth	Good
PZN	Good	Good	Abstract	Not good
Z	Not good	Good	Abstract	Not good

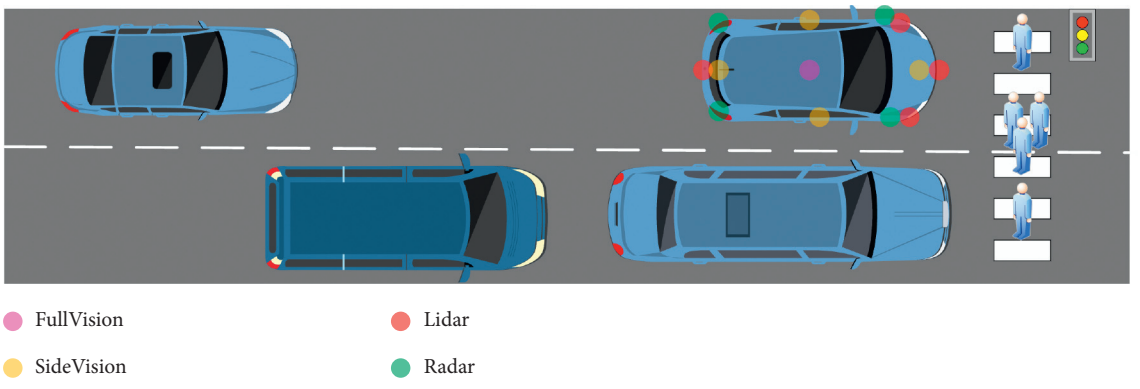

Figure 7: The environment of a case study.

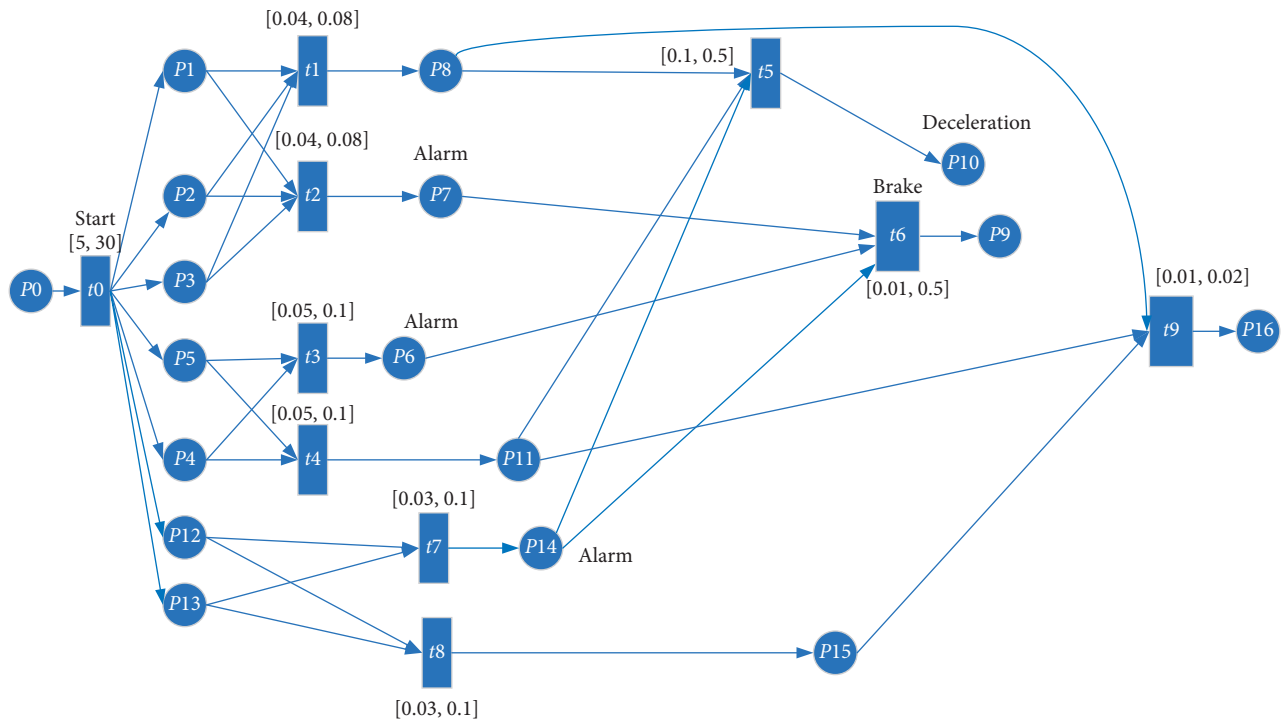

Figure 8: TPN model of the case study.

Table 2: The detail of SI.

I	φ_{fi}	$\mathbf{M_i}$	$\mathbf{D_i}$	$\mathbf{SI_i}$
$i=0$	$(M_0, D_0, \mathrm{SI}_0)$	P_0	$\{D_0(t_0) = [5, 30]\}$	$[0, 0]$
$i=1$	$(M_1, D_1, \mathrm{SI}_1)$	$P_1 + P_2 + P_3 + P_4 + P_5 + P_{12} + P_{13}$	$\{D_1(t_1) = [5.04, 30.08],$ $D_1(t_2) = [5.04, 30.08],$ $D_1(t_3) = [5.05, 30.1],$ $D_1(t_4) = [5.05, 30.1],$ $D_1(t_7) = [5.03, 30.1],$ $D_1(t_8) = [5.03, 30.1]\}$	$[5, 30]$
$i=2$	$(M_2, D_2, \mathrm{SI}_2)$	$P_7 + P_6 + P_{14}$	$\{D_2(t_6) = [5.06, 30.6]\}$	$[5.05, 30.01]$
$i=3$	$(M_3, D_3, \mathrm{SI}_3)$	$P_8 + P_{11} + P_{14}$	$\{D_3(t_5) = [5.15, 30.6]\}$	$[5.05, 30.1]$
$i=4$	$(M_4, D_4, \mathrm{SI}_4)$	$P_8 + P_{11} + P_{15}$	$\{D_4(t_9) = [5.06, 30.12]\}$	$[5.05, 30.1]$
$i=5$	$(M_5, D_5, \mathrm{SI}_5)$	P_9	$\varnothing$	$[5.35, 30.6]$
$i=6$	$(M_6, D_6, \mathrm{SI}_6)$	P_{16}	$\varnothing$	$[5.06, 30.12]$
$i=7$	$(M_7, D_7, \mathrm{SI}_7)$	P_{10}	$\varnothing$	$[5.15, 30.6]$

Table 3: The details of states and operations.

P_0	The start of intelligent car	P_{14}	Obstacles, traffic light, and so on
P_1	Lidar 1	P_{15}	Normal environment
P_2	Lidar 2	P_{16}	Keep running
P_3	Lidar 3	t_0	Start
P_4	Radar 1	t_1	Processed normal data by lidar system
P_5	Radar 2	t_2	Processed abnormal data by lidar system
P_6	Detected obstacles ahead by radar	t_3	Processed normal data by radar system
P_7	Detected obstacles ahead by lidar	t_4	Processed abnormal data by radar system
P_8	Detected normal environment by radar	t_5	Decelerating
P_9	Brake	t_6	Braking
P_{10}	Deceleration	t_7	Process by vision-front
P_{11}	Detected normal environment by lidar	t_8	Process by wide-angle
P_{12}	Vision-front	t_9	Check information
P_{13}	Wide-angle		

down, and φ_{f6} is running straight normally. The transfer matrix L_{DP}, D^+, and D^- of this case study is as follows:

$$
L_{DP} = \begin{array}{c} \\ p_0 \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ p_8 \\ p_9 \\ p_{10} \\ p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{15} \\ p_{16} \end{array}
\begin{array}{c}
\begin{array}{cccccccccccccccc} p_0 & p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & p_7 & p_8 & p_9 & p_{10} & p_{11} & p_{12} & p_{13} & p_{14} & p_{15} & p_{16} \end{array} \\
\begin{bmatrix}
0 & t_0 & t_0 & t_0 & t_0 & t_0 & 0 & 0 & 0 & 0 & 0 & 0 & t_0 & t_0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & t_2 & t_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & t_2 & t_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & t_2 & t_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & t_3 & 0 & 0 & 0 & 0 & t_4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & t_3 & 0 & 0 & 0 & 0 & t_4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t_5 & 0 & 0 & 0 & 0 & 0 & t_9 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t_5 & 0 & 0 & 0 & 0 & 0 & t_9 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t_7 & t_8 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t_7 & t_8 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t_6 & t_5 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t_9 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\end{array},
$$

$$
D^- = \begin{array}{c} \\ t_0 \\ t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \\ t_8 \\ t_9 \end{array}
\begin{array}{c}
\begin{array}{cccccccccccccccc} p_0 & p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & p_7 & p_8 & p_9 & p_{10} & p_{11} & p_{12} & p_{13} & p_{14} & p_{15} & p_{16} \end{array} \\
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
\end{array},
$$

$$
D^+ = \begin{array}{c} \\ t_0 \\ t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \\ t_8 \\ t_9 \end{array}
\begin{array}{c}
\begin{array}{cccccccccccccccc} p_0 & p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & p_7 & p_8 & p_9 & p_{10} & p_{11} & p_{12} & p_{13} & p_{14} & p_{15} & p_{16} \end{array} \\
\begin{bmatrix}
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\end{array}.
$$

(5)

From the matrix L_{DP}, the concurrent behavior can be easily found. By the D^+, D^-, M_i, M_j, φ_{fi}, and φ_{fj}, the next behavior can be deduced exactly. The exact arrival time can also be obtained from SI_i and SI_j from the reachability tree as shown in Figure 5. The rules can be amended through the Z_p and Z_t with the new data coming as well. Every Z frame

structure can be coded by high-level programming language so to reason the logic relationship easily.

6. Conclusions

In this paper, we propose a new way that uses TPN and *Z* frame structure to formally model and analyze the safety and accessibility of the Internet of Vehicles. The method has been explained in detail by a case study. Although it promotes the efficiency of finding problem when the system goes wrong and can predict the future behavior, the multiple intelligent vehicles working cooperatively are not taken into account, which is an important and intriguing topic that we are working on.

Data Availability

The case study data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by National Natural Science Foundation under Grants 61903053 and 61703063, Science and Technology Research Project of Chongqing Municipal Education Commission of China under Grants Nos. KJZD-K201800701 and KJQN201900702, Chongqing Engineering and Technology Research Center for Big Data of Public Transportation Operation under Grant 2019JTDSJ-YB02, and Guizhou Science and Technology Program [2020] 4Y056.

References

[1] C. M. Martinez, M. Heucke, F. Y. Wang et al., "Driving style recognition for intelligent vehicle control and advanced driver assistance: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 99, pp. 1–11, 2018.

[2] Y. Quan, H. Yang, and L. Yang, "Information security impacts future traffic safety of intelligent vehicle," in *Proceedings of the International Conference on Man-Machine-Environment System Engineering*, pp. 731–738, Beijing, China, 2018.

[3] L. B. Chen, H. Y. Li, W. J. Chang et al., "An intelligent vehicular telematics platform for vehicle driving safety supporting system," in *Proceedings of the International Conference on Connected Vehicles & Expo*, pp. 210-211, Shenzhen, China, October 2015.

[4] M. Kamali, L. A. Dennis, O. Mcaree, M. Fisher, and S. M. Veres, "Formal verification of autonomous vehicle platooning," *Science of Computer Programming*, vol. 148, pp. 88–106, 2017.

[5] Y. Teng, L. Qi, and Y. Du, "A logic petri net-based repair method of process models with incomplete choice and concurrent structures," *Computing and Informatics*, vol. 39, no. 1-2, pp. 264–297, 2020.

[6] A. Boucherit, L. M. Castro, A. Khababa, and O. Hasan, "Petri net and rewriting logic based formal analysis of multi-agent based safety-critical systems," *Multiagent and Grid Systems*, vol. 16, no. 1, pp. 47–66, 2020.

[7] Y. Riouali, L. Benhlima, and S. Bah, "Petri net extension for traffic road modelling," *Computer Systems & Applications*, vol. 7, no. 11, pp. 7–12, 2017.

[8] Y. Liu, J. Z. Wu, and R. Qiao, "Consistency verification between goal model and process model in requirements analysis of networked software," *Journal of Computational and Theoretical Nanoscience*, vol. 11, no. 5, pp. 1248–1261, 2014.

[9] Y. Liu, J. Z. Wu, and R. Qiao, "Dynamic evolution of requirements process model deployed on networked environment with PZN," *Journal of Computational Information Systems*, vol. 9, no. 8, pp. 3329–3336, 2013.

[10] C. Liu, Q. Zeng, H. Duan et al., "Petri net based data-flow error detection and correction strategy for business processes," *IEEE Access*, vol. 8, pp. 43265–43276, 2020.

[11] R. Bruni and U. Montanari, "CCS, the calculus of communicating systems," *Models of Computation*, pp. 221–270, Springer, Berlin, Germany, 2017.

[12] R. C. Bhushan and D. K. Yadav, "Modelling a safety-critical system through CCS," *International Journal of Applied Engineering Research*, vol. 12, no. 21, pp. 11213–11217, 2017.

[13] J. Whitney, C. Gifford, and M. Pantoja, "Distributed execution of communicating sequential process-style concurrency: golang case study," *The Journal of Supercomputing*, vol. 75, no. 3, pp. 1396–1409, 2019.

[14] M. Hatzel, C. Wagner, K. Peters, and U. Nestmann, "Encoding CSP into CCS," *Electronic Proceedings in Theoretical Computer Science*, vol. 190, pp. 61–75, 2015.

[15] V. Bandur V, P. W. V. Tran-Jørgensen, M. Hasanagic et al., "Code-generating VDM for embedded devices," in *Proceedings of the 15th Overture Workshop*, London, UK, October 2017.

[16] M. Hasanagić, T. Fabbri, P. G. Larsen et al., "Code generation for distributed embedded systems with VDM-RT," *Design Automation for Embedded Systems*, vol. 23, no. 3-4, pp. 153–177, 2019.

[17] D. Sabatier, "Using formal proof and B method at system level for industrial projects," *Reliability, Safety, and Security of Railway Systems*, pp. 20–31, Springer, Berlin, Germany, 2016.

[18] P. Saratha, G. V. Uma, and B. Santhosh, "Formal specification for online food ordering system using Z language," in *Proceedings of the 2017 Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM)*, pp. 343–348, IEEE, Tindivanam, India, February 2017.

[19] G. O'Regan, "Z formal specification language," *Concise Guide to Formal Methods*, pp. 155–171, Springer, Berlin, Germany, 2017.

[20] Z. H. Muhamad, D. A. Abdulmonim, and B. Alathari, "An integration of uml use case diagram and activity diagram with Z language for formalization of library management system," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 4, p. 3069, 2019.

[21] T. Gouasmi, A. Regayeg, and A. H. Kacem, "Automatic generation of an operational CSP-Z specification from an abstract temporalZ specification," in *Proceedings of the 2012 IEEE 36th Annual Computer Software & Applications Conference Workshops*, pp. 248–253, Izmir, Turkey, July 2012.

[22] B. Mahony and S. D. Jin, "Blending object-Z and timed CSP: the semantics of TCOZ," in *Proceedings of the ICSE'98 Proceedings of the 20th International Conference on Software Engineering*, pp. 95–104, Kyoto, Japan, April 1998.

[23] Y. Liu, J. Z. Wu, R. Zhao et al., "Formal verification of process layer with petri nets and Z," *Advances in Information Sciences and Service Sciences*, vol. 5, no. 1, pp. 68–77, 2013.
[24] F. Peschanski and D. Julien, "When concurrent control meets functional requirements or Z+Petri nets," *ZB 2003: Formal Specification and Development in Z and B*, pp. 79–97, Springer, Berlin, Germany, 2003.
[25] T. Yin, Z. Li, C. Seatzu et al., "Verification of state-based opacity using petri nets," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2823–2837, 2017.
[26] X. Wu, S. Tian, and L. Zhang, "The Internet of Things enabled shop floor scheduling and process control method based on Petri nets," *IEEE Access*, vol. 7, pp. 27432–27442, 2019.
[27] Z. Ding, Y. Zhou, and M. C. Zhou, "Modeling self-adaptive software systems with learning petri nets," *IEEE Transactions on Systems Man & Cybernetics Systems*, vol. 46, no. 4, pp. 483–498, 2017.
[28] L. Wei, W. Lu, Y. Du et al., "Deadlock property analysis of concurrent programs based on petri net structure," *International Journal of Parallel Programming*, vol. 45, no. 4, pp. 1–20, 2016.
[29] M. Gaied, A. M'halla, D. Lefebvre, and K. Ben Othmen, "Robust control for railway transport networks based on stochastic P-timed Petri net models," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 233, no. 7, pp. 830–846, 2019.
[30] E. Kučera, O. Haffner, P. Drahoš et al., "New software tool for modeling and control of discrete-event and hybrid systems using timed interpreted petri nets," *Applied Sciences*, vol. 10, no. 15, p. 5027, 2020.
[31] H. B. Attia, L. Kahloul, S. Benhazrallah et al., "Using hierarchical timed coloured petri nets in the formal study of TRBAC security policies," *International Journal of Information Security*, vol. 19, no. 2, pp. 163–187, 2020.
[32] B. Aman, P. Battyányi, G. Ciobanu et al., "Local time membrane systems and time petri nets," *Theoretical Computer Science*, vol. 805, pp. 175–192, 2018.
[33] R. Cao, L. Hao, F. Wang et al., "Modelling and analysis of hybrid stochastic timed Petri net," *Journal of Control and Decision*, vol. 6, no. 3, pp. 1–21, 2018.
[34] P.-A. Bourdil, B. Berthomieu, S. Dal Zilio, and F. Vernadat, "Symmetry reduction for time Petri net state classes," *Science of Computer Programming*, vol. 132, pp. 209–225, 2016.
[35] B. Berthomieu, D. L. Botlan, and S. D. Zilio, *Petri Net Reductions for Counting Markings*, pp. 1–20, Springer, Berlin, Germany, 2018.
[36] C. J. Jiang and Z. J. Ding, *Petri Net Refinement Based System Modeling and Analysis*, pp. 91–105, Tongji University Press, Shanghai, China, 2017.
[37] H. Duan, C. Liu, Q. Zeng et al., "Refinement-based hierarchical modeling and correctness verification of cross-organization collaborative emergency response processes," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 8, pp. 2845–2859, 2018.