Three Fog Computing Based Variants of Congestion Control in ITS

Ananya Paul, Kiton Ghosh, Sulata Mitra

Abstract: The growth of vehicles and inadequate road capacity in the urban area trigger traffic congestion and raise the frequency of road accident. Therefore the need of drastically reducing traffic congestion is a significant concern. Advancement in the technology like fog computing, Internet of Things (IoT) in Intelligent Transportation Systems (ITS) aid in the more constructive management of traffic congestion. Three IoT based Fog computing oriented models are designed in the present work for mitigating traffic congestion. The first two schemes are vehicle dependent as they control traffic congestion depending upon the number of vehicles and their direction of movement across the intersections. The third scheme is environment dependent as the agent senses the environment and controls the sequence of green signal at different routes dynamically. The performances of the three schemes in ITS are analyzed along with the comparison of storage, communication and computation overhead. The efficacy of the schemes is studied theoretically and quantitatively. The quantitative performance of the three schemes is compared with five existing schemes. On the basis of the result of the comparison, it can be concluded that the proposed schemes are capable of alleviating congestion more optimally than existing schemes due to the substantial reduction in vehicle waiting time.

Index Terms—Traffic signal control, VANET, Congestion con-trol

Keywords: The performances of the three schemes in ITS are analyzed along with the comparison of storage, communication and computation overhead.

I. INTRODUCTION

Vehicular transportation is one of the crucial means of transportation around the world. In urban areas, where the number of vehicles continuously escalates faster than the available traffic infrastructure, the traffic management becomes more inefficient and cumbersome [1], [2]. This inefficient and disorganized management has led to severe traffic congestion which has become a troublesome issue of late. Controlling or reducing traffic congestion has become one of the major important tasks around the world [3], [4]. As of late, fog computing with IoT plays an important role in the development of a sustainable traffic flow in ITS to mitigate the ever increasing traffic congestion [5].

Many works reported so far provide solutions for reducing traffic congestion in ITS. The vehicle flow rate at different lanes across an intersection is considered as fixed in [6]. The duration of green signal in the incoming lanes of an intersection is considered as fixed in [7], [8]. But the flow of traffic varies throughout the day and hence the green signal duration needs to be calculated dynamically. The green signal duration to the incoming lanes is decided dynamically in [9] unlike [7], [8] but the total duration of green signal of all the four incoming lanes is considered as fixed. So the increase in green signal duration of one lane reduces the green signal duration of the other lanes which in turn increases the waiting time of vehicles in the lanes having less green signal duration. A routing algorithm is proposed in [10] to reduce traffic congestion. But the selected route is not verified as a least congested route among all the other routes during simulation. The congestion of such a selected route is measured in [11] depending upon the number of vehicles in that route at a particular time. In [1] a lane is considered as congested if a vehicle is found in that lane for a prolonged period of time. No other parameters (e.g. waiting time, average speed of vehicles) are considered to measure the congestion in [1], [11]. The duration of green signal is not calculated in [12]. Moreover, in [12] the green signal is set to a lane as soon as the waiting time of vehicles in that lane crosses a predefined threshold. The performance of [13] is tested in a single intersection by varying the number of vehicles from 5-200 only which is not valid in real network.

All the schemes described above do not cope up with real time network. This motivates us to propose three advanced variants (Var 1, Var 2 and Var 3) based on fog computing for controlling traffic congestion of an intersection (Fig. 1) in ITS. The intersection consists of 4 incoming lanes (*$lane_2$*, *$lane_4$*, *$lane_6$*, *$lane_8$*) and 4 outgoing lanes (*$lane_1$*, *$lane_3$*, *$lane_5$*, *$lane_7$*). The intersection has 4 straight routes (2->5, 4->7, 6->1, 8->3), 4 left turn routes (2->7, 4->1, 6->3, 8->5) and 4 right turn routes (2->3, 4->5, 6->7, 8->1). The vehicles can take right turn from the intersection without considering the signal, whereas the vehicles should consider the signal while going straight by crossing the intersection distance (d1 in Fig. 1) and taking left turn by crossing the arc distance (d2 in Fig. 1). The arc distance d2 is calculated as $2*\pi*d1*(\theta/360)$, where θ is 90 . The number of vehicles in each row of a lane is assumed as two for all the three variants. In Var 1 and Var 2 an intersection controller is maintained at the intersection for controlling traffic congestion.

Manuscript received on May 21, 2021.
Revised Manuscript received on May 28, 2021.
Manuscript published on May 30, 2021.
* Correspondence Author

Ananya Paul*, Department of Computer Science and Technology, IIEST, Shibpur Howrah, India. Email: ananya.rs2017@cs.iiests.ac.in

Kiton Ghosh, Department of Computer Science and Technology, IIEST, Shibpur Howrah, India. Email: kitonghosh@gmail.com

Sulata Mitra, Department of Computer Science and Technology, IIEST, Shibpur Howrah, India. Email: sulata@cs.iiests.ac.in

Four Fog devices are present both in Var 1 and Var 2. Fog 12 is associated with $lane_1$ and $lane_2$, Fog 34 is associated with $lane_3$ and $lane_4$, Fog 56 is associated with $lane_5$ and $lane_6$, Fog 78 is associated with $lane_7$ and $lane_8$. In Var 3 an agent at the intersection controls traffic congestion.

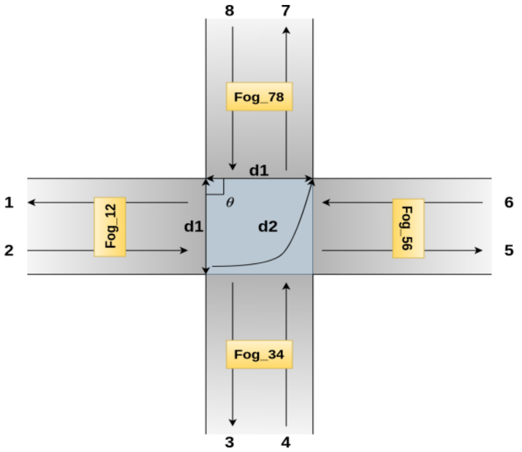


Fig. 1. Routes of an intersection

Var_1 is an extension of [14]. In Var 1_the intersection controller and Fogs maintain Table I to determine the fixed sequence of green signal for the routes as (2->7, 6->3), (4->7, 8->3), (2->5, 6->1), (4->1, 8->5). So the green signal is not on in all the straight and left turn routes corresponding to a particular incoming lane simultaneously. Let, at t^{th} instant of time, green signal is on in the routes 4->7 and 8->3. The desired route for the vehicles which are in front of $lane_8$ is 8->5 which is not in the green signal at t^{th} instant of time. Hence, such vehicles cannot cross the intersection. As a result, all the vehicles which are present behind such vehicles in $lane_8$ have to wait until the green signal is on in the route 8->5, which increases the waiting time of vehicles in $lane_8$.

In Var_2, three scheduling algorithms are proposed to control the congestion of the intersection. In Var 2 at t^{th} instant of time, green signal is on in straight and left routes of a particular incoming lane. So, green signal is on in both 8->5 and 8->3 simultaneously unlike Var_1.

Var 1 and Var 2 are vehicle dependent dynamic approaches where the traffic is controlled depending on the number of vehicles and the direction of movement of the vehicles. Each time the duration of green signal is decided based on these two factors. To cope up with this randomness and fluctuation, environment dependent Q-learning approach is used in Var 3 where the agent senses the traffic environment at the intersec- tion for controlling traffic congestion.

In the present work, the vehicle flow rate and maximum waiting time of vehicles are considered as random unlike [6], [15], [16]. Both Var 2 and Var 3 determine the sequence of green signal dynamically in contrast to the fixed sequence of green signal in Var 1. The duration of green signal to the lanesare computed appropriately for all the three variants unlike [8], [9], [17]–[19]. The effect of such dynamic variation of the green signal duration is studied during simulation by observing the waiting time unlike [10]. The performance of the proposed variants is studied qualitatively, theoretically (using queuing system) and quantitatively. The qualitative performance is evaluated in terms of communication, storage and computation overhead. Both the theoretical and quantitative performance are assessed based on the waiting time of vehicles. The sim- ulation experiments are conducted for observing the variation of waiting time of the vehicles in multiple intersections under 5 to 2600 vehicles unlike [13].

II. EXISTING WORK

Several schemes [7], [15], [16], [18]–[24] for controlling traffic congestion have already been proposed. The schemes [7], [15], [16], [20]–[22], [24] are vehicle dependent whereas the schemes [18], [19], [23] are environment dependent. Traffic Light Controller (TLC) receives all the information from the vehicles through inductive loop detectors [24], cam- eras [7], [21] or radars for controlling the traffic light dynam- ically. But this approach is expensive, has high computational complexity and latency [22]. Moreover, the camera or video recorder installed in the road might not work in bad weather. In [20], road belts are setup at each entrance and each exit of the road to detect the entering vehicles and to inform the Road Side Unit (RSU) about it. RSU counts the number of vehicles and broadcasts message for vehicles. It inserts a record of the data (speed, direction) in the database for each new vehicle. All the RSUs calculate the total load of each road and send this to TLC. TLC determines the maximum load of a road and compares the load value of other roads with the maximum load value. TLC chooses the road having maximum load value and the road whose load value has the least difference with maximum load value for sharing green signal in the next phase. The vehicles in the road having less load value are not getting a chance to cross the road which increases the waiting time. Moreover the road belt needs to be setup at the entrance and exit of each road which incurs a high cost.

In [15], TLC calculates the waiting time of each vehicle in a road in the current phase and changes the signal to green in case the waiting time of the vehicles crosses a static threshold. Moreover, TLC calculates the traffic flow data of the roads and exchanges this value with its neighbor TLCs to predict the traffic flow at the beginning of the next phase. TLC sets green signal to the pair of roads which have the largest traffic flow and sets a red signal for the other roads. The exchange of information with the neighbor TLCs increases the communication overhead. The static threshold for the maximum waiting time of vehicles might not be appropriate as the traffic flow will not be the same throughout the day.

One vehicle among the vehicles which are reaching the intersection is picked up as the leader vehicle [16]. The driver sends whether to change the traffic light or keep the green light longer to the leader vehicle. The leader vehicle estimates the priority of each route depending upon the traffic flow. If the waiting time of vehicles in a route exceeds the constant threshold value for red light duration, green light is assigned tothat route.

Retrieval Number: 100.1/ijrte.A59660510121
DOI: 10.35940/ijrte.A5966.0510121

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

If the time of green light of any route crosses the constant threshold value for the green light duration, traffic light is assigned to that route according to the traffic flow calculated by the leader vehicle. The use of such constant threshold is not suitable for fast changing traffic in urbanareas. If all vehicles on the green light route have passed the intersection, green light will be shifted from the current route to the route with the highest priority. So, the vehicles waiting in the route with less priority are not getting a chance to cross the intersection which increases the waiting time of such vehicle.

In [23], the agent captures the images of the intersection via web camera, determines the edges of the intersection using edge detection method and captures the current objects that are present on the edges to determine the vehicle density. The agent checks the current state, performs an action and receives reward or penalty. The state is changed if the agent receives reward. But the current state needs to be changed into a new state in case the agent receives a penalty as it helps the agent to know the action which is to be performed in the next state for receiving reward to minimize the penalty.

In [18] the agent identifies the current state in the environment, performs a selected action and updates the Q-table after receiving reward or penalty. After performing the action, the agent checks if the queue length in the next state has reduced from the previous state. If it is not reduced the process starts from the beginning. In [18], the states of the Q- learning algorithm are the number of vehicles at each lane of the intersection. The actions are distribution of green signals, with fixed duration to the lanes. Since, the traffic flow is unpredictable and fast changing, the fixed duration of green signal is unable to mitigate the traffic congestion efficiently.

In [19] a lane is segmented in cells. The state is composed of presence of a vehicle or not in the cell, the speed of the vehicle and the current traffic signal phase. The possible actions are the distribution of traffic signal in different lanes. The agent observes the state of the environment chooses an action and receives a reward or penalty after performing the action. But in [19], the duration of the green signal is not considered asa part of the action.

III. PRESENT WORK

All the three Fog-based variants are elaborated in this section.

Table I Four Different States

Current state	Green Signal Route (GSR)	Red Signal Route (RSR)	Next state
State 1	2->7, 6->3	(4->7, 8->3) (2->5, 6->1) (4->1, 8->5)	State 2
State 2	4->7, 8->3	(2->5, 6->1) (4->1, 8->5) (2->7, 6->3)	State 3
State 3	2->5, 6->1	(4->1, 8->5) (2->7, 6->3) (4->7, 8->3)	State 4
State 4	4->1, 8->5	(2->7, 6->3) (4->7, 8->3) (2->5, 6->1)	State 1

A. Var 1

Var_1 considers four states as shown in Table I. The sequential execution of these four states creates a cycle. Each vehicle after entering into an incoming lane sends a message (M V) to the Fog associated with that lane in the form (type id, route id), where type id and route id are message type identification and route identification respectively. Fogs keep receiving M Vs from the beginning to the end of aparticular state.

Each Fog searches Table I at the end of the current state to identify the route that enters into green signal (GSR) and red signal (RSR) in the next state. Let Route 1(IL1->OL1) is GSR and Route 2 (IL2->OL2) is RSR. IL1/(IL2) and OL1/(OL2) are the incoming lane and outgoing lane respectively whichare associated with Route 1/(Route 2).

Fog which is associated with OL1/(OL2) computes the capacity of OL1/ (OL2) as $CV_1/(CV_2)$. $CV_1/(CV_2)$ is defined as how many more number of vehicles can enter into OL1/(OL2) and is estimated as the difference of the maximum capacity (MAX CAP) and the number of vehicles present in OL1/(OL2). Fog sends $CV_1/(CV_2)$ to the Fog which is associated with IL1/(IL2) in the form of a message (M Fog1) to prevent the entry of vehicles more than $CV_1/(CV_2)$ into OL1/(OL2) for avoiding congestion in Route 1/ (Route 2).

Fog which is associated with IL1/(IL2) determines whether Route 1/ (Route 2) is congested using Algorithm 1 (CONGESTION_ DETERMINATION). The Fog counts the number of waiting vehicles in IL1/(IL2) as $Count_{IL1}/(Count_{IL2})$ that are willing to enter into OL1/(OL2) using the information of the route id field of M V. Route 1/ (Route 2) is congested if $Count_{IL1}/(Count_{IL2})$ is greater than $CV_1/(CV_2)$.

Fog which is associated with IL1

Allows $CV_1/(Count_{IL1})$ number of vehicles to enter into OL1 when Route 1 becomes green in the next state if Route 1 is congested/(not congested)

Computes the time which is required for $CV_1/(Count_{IL1})$ number of vehicles to cross the intersection as normal duration of green signal ($Normal_{green}$) in Route 1

Assumes that N_{IL1} number of vehicles will enter into IL1 when Route 1 becomes green in the next state. N_{IL1} is considered as same as $Count_{IL1}$

Computes the time which is required for (CountIL1 + NIL1) number of vehicles to cross the intersection as maximum duration of green signal (Thresholdgreen) which is also the maximum waiting time of vehicles in red signal in Route_2

Computes $Normal_{green}$ and $Threshold_{green}$ using Algo- rithm 2 (*Computation* $Normal_{green}$ $Threshold_{green}$)

Sends ($Normal_{green}$, $Threshold_{green}$) to the intersection controller in the form of a message (M Fog2) for determining the actual duration of green signal in Route 1

Fog which is associated with IL2 sends a message (M Fog3) to the intersection controller to inform whether congestion occurs (Occurrence of congestion) in Route 2.

The intersection controller reads the green signal routes in the current state (current GSRs) and in the next state (next GSRs) from Table I. current GSRs enter into red signal in the next state.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

The intersection controller computes the actual duration of green signal for next GSRs and the maxi- mum waiting time of vehicles in current GSRs after receiving M Fog2 and M Fog3 from the Fogs using Algorithm 3.

Var 1 uses the parameters as shown in Table II.

Table II Parameters Used In Var 1

Parameter	Definition
$Count_2$, $Count_4$, $Count_6$, $Count_8$	Number of waiting vehicles in incoming lane $lane_2$, $lane_4$, $lane_6$, $lane_8$ respectively
CV_1, CV_3, CV_5, CV_7	Capacity of vehicles in outgoing lane $lane_1$, $lane_3$, $lane_5$, $lane_7$ respectively
$Normal_{green\ 47}$, $Normal_{green\ 83}$, $Normal_{green\ 25}$, $Normal_{green\ 61}$, $Normal_{green\ 41}$, $Normal_{green\ 85}$, $Normal_{green\ 27}$, $Normal_{green\ 63}$	Normal duration of green signal in the route 4->7, 8->3, 2->5, 6->1, 4->1, 8->5, 2->7,6->3 respectively
$Threshold_{green\ 47}$, $Threshold_{green\ 83}$, $Threshold_{green\ 25}$, $Threshold_{green\ 61}$, $Threshold_{green\ 41}$, $Threshold_{green\ 85}$, $Threshold_{green\ 27}$, $Threshold_{green\ 63}$	Maximum duration of green signal in the route 4->7, 8->3, 2->5, 6->1, 4->1, 8->5, 2->7,6->3 respectively

State _1: It can be observed from Table I that the routes 4->7 and 8->3 enter into green signal (GSR) whereas the route 2->7 and 6->3 enter into red signal (RSR) when the state changes from State 1 to State 2. $lane_4$/($lane_8$) isIL1 and $lane_7$/($lane_3$) is OL1 for the route 4->7/(8->3). $lane_2$/($lane_6$) is IL2 and $lane_7$/($lane_3$) is OL2 for the route2->7/(6->3).

$Count_4$/($Count_8$) is $Count_{IL1}$ for the route 4->7/(8->3). Fog 34/(Fog 78) is associated with the route 4->7/(8->3) whereas Fog 12/(Fog 56) is associated with the route 2->7/(6->3). The functions of all these Fogs are elaborated below.

Functions of Fog 34/ (Fog 78):

Calls Algorithm 1 (CONGESTION DETERMINATION (State 1, $lane_3$, 4->7))/ (CONGESTION DETERMINATION

(State 1, $lane_7$, 8->3)) to determine whether the route 4->7/(8->3) is congested -State 1 is the current state, $lane_3$ is the outgoing lane which is associated with Fog 34 and $lane_7$ is the outgoing lane which is associated with Fog 78

Sets the returned value as $Count_4$= $Count_a$, CV_3= CV_a' , CV_7= CV_b, occ cong= Occurrence of congestion/ ($Count_8$= $Count_a$, CV_7= CV_a' , CV_3=CV_b, occ cong= Oc- currence of congestion)

Sends M Fog1 in the form (Fog 34, CV_3) /(Fog 78, CV_7) to (Fog 56 associated with 6->3, Fog 78 associated with 8->3) /(Fog 12 associated with 2->7, Fog 34 associated with 4->7) Calls Algorithm 2 (*Computation* $Normal_{green}$ $Threshold_{green}$ (4->7, d1,$Count_4$, CV_7, occ cong))/ (*Computation* $Normal_{green}$ $Threshold_{green}$ (8->3, d1, $Count_8$, CV_3, occ cong)) to compute $Normal_{green\ 47}$, $Threshold_{green\ 47}$/($Normal_{green\ 83}$, $Threshold_{green\ 83}$) for the route 4->7/(8->3)

- Here d1 is used as both 4->7 and 8->3 are straight routesSets the returned value as $Normal_{green\ 47}$= $Normal_{green\ ab}$ / ($Normal_{green\ 83}$= $Normal_{green\ ab}$), $Threshold_{green\ 47}$=$Threshold_{green\ ab}$ / ($Threshold_{green\ 83}$= $Threshold_{green\ ab}$)

Sends M Fog2 in the form (Fog 34, $Normal_{green\ 47}$, $Threshold_{green\ 47}$)/ (Fog 78, $Normal_{green\ 83}$, $Threshold_{green\ 83}$) to the intersection controller

Function of Fog 12/ (Fog 56):

Calls Algorithm 1 (CONGESTION DETERMINATION (State 1, $lane_1$, 2->7))/ (CONGESTION DETERMINA-

TION (State 1, $lane_5$, 6->3)) to determine whether the route 2->7/(6->3) is congested

- $lane_1$ is the outgoing lane which is associated with Fog 12 and $lane_5$ is the outgoing lane which is associated with Fog 56

Sets occ cong= Occurrence of congestion

Sends M Fog3 in the form (Fog 12, occ cong)/((Fog 56, occ cong)) to the intersection controller

Algorithm 1 the Congestion Determination Algorithm

procedure CONGESTION DETERMINATION (State, $lane_a'$, a->b)
  if State==State 1 **then**
    Out Ln= $lane_3$, $lane_7$ /* Out Ln is the outgoinglanes of GSR and RSR in the current state */
  end if
  if (State==State 2) or (State==State 4) **then**
    Out Ln= $lane_1$, $lane_3$, $lane_5$, $lane_7$
  end if
  if State==State 3 **then**
    Out Ln= $lane_1$, $lane_5$
  end if
  Estimates the number of waiting vehicles in $lane_a$ as $Count_a$ that are willing to enter into $lane_b$
  if $lane_a'$ $\in$ Out Ln **then**
    Estimates the capacity of $lane_a'$ as CV_a'
  end if
  Reads CV_b from M Fog1 after receiving it from theFog associated with $lane_b$
  if $Count_a$ > CV_b **then**
    Occurrence of congestion=1 /*Route a->b is con-gested */
  else
    Occurrence of congestion=0 /*Route a->b is notcongested */
  end if
  return ($Count_a$, CV_a' , CV_b, Occurrence of conges-tion)
end procedure

All the Fogs maintain a static database (Table III) to store Range of number of vehicles and Range of speed (Km/hr). Each Range of number of vehicles is in the form (veh min- veh max), where veh min is the lowest number of vehicles and veh max is the highest number of vehicles in that range. Similarly, each Range of speed corresponding to a particular range of number of vehicles is in the form (speed min- speed max), where speed min is the lowest speed of vehicles and speed max is the highest speed of vehicles in that range.

For example, (41-60)% is a range of the number of vehicles and (37-42) Km/hr is the corresponding range of speed. So, veh min, veh max, speed min, speed max are 41%, 60%, 37Km/hr, 42Km/hr respectively.

Retrieval Number: 100.1/ijrte.A59660510121
DOI: 10.35940/ijrte.A5966.0510121

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

So, 37Km/hr is the speed when the number of vehicles in the lane is 60% of MAX CAP and 42Km/hr is the speed when the number of vehicles in the lane is 41% of MAX CAP.

Table III Relation Between Number of Vehicles And Range of Speed

Range of number of vehicles	Range of speed (Km/hr)
(0-20)% of MAX CAP	65-70
(21-40)% of MAX CAP	51-56
(41-60)% of MAX CAP	37-42
(61-80)% of MAX CAP	23-28
(81-100)% of MAX CAP	9-14

Let the route IL1-> OL1 is green and N_{IL1} number of vehicles enter into IL1. Fog which is associated with IL1 searches Table III to

- Find the range of vehicles corresponding to N_{IL1}
- Read the value of veh min, veh max, speed min, speed max to calculate the average speed ($speed_{IL1}$) for N_{IL1} number of vehicles in IL1

For example, let N_{IL1} =70. So 61-80% is the range of number of vehicles for N_{IL1} and 23-28 Km/hr is the range of speed corresponding to the range of number of vehicles 61-80%. Hence, for N_{IL1}, veh min=61%, veh max=80%, speed min=23Km/hr, speed_max=28Km/hr. So,

$$speed_{IL1} = 23 + \frac{(80-70)}{(80-61)}(28-23) \qquad (1)$$

Algorithm 2 uses the following parameters. The definition of these parameters is elaborated for IL1.

start_speed → The starting speed of the waiting vehicles in IL1 when the signal of IL1 changes from red to green

T1/(T2) → Time which is required for all $Count_{IL1}$/(N_{IL1}) number of waiting vehicles in IL1 to go straight from the intersection for the straight routes after crossing distance d1 (d'=d1 in Fig. 1) or to take left turn from the intersection for the left turn routes after crossing distance d2 (d'=d2 in Fig.1)

T3/(T5) → Time which is required for the first row of $Count_{IL1}$/(N_{IL1}) number of vehicles in IL1 to cross the distance d1 (for going straight from the intersection) or d2 (for taking left turn from the intersection) with start speed/ ($speed_{IL1}$)

T4/(T6) → Time which is required for the rest of the rows (except the first row) of $Count_{IL1}$/(N_{IL1}) number of vehicles in IL1 to cross the inter vehicle distance and the length of the vehicle in its front row with start speed/($speed_{IL1}$)

Algorithm 2: T3 is computed for the vehicles in the front row of $lane_a$ to cross distance d' from the intersection (Line 2). T1 is computed for $Count_a$ number of waiting vehicles

i.e $\left\lceil \frac{Count_a}{2} \right\rceil^{th}$ row of vehicles for entering into $lane_b$

Line 4). The value of N_a is assumed as same as $Count_a$

(Line 8). T2 is computed for N_a *vehicles i.e.* $\left\lceil \frac{N_a}{2} \right\rceil^{th}$

Algorithm 2 Computation of $Normal_{green}$ and $Threshold_{green}$

1: **procedure** COMPUTATION_ $NORMAL_{GREEN}$_ $THRESHOLD_{GREEN}$(a->b, d', $Count_a$, CV_b, occ_cong)
2: T3 = (d'/start_speed)
3: T4=(vehicle length+inter vehicle distance)/(start_speed)
4: $$T1 = T3 + T4 * (\left\lceil \frac{Count_a}{2} \right\rceil - 1)$$
5: Calculate average speed of $lane_a$ ($speed_a$) using TableIII and Equ. 1
6: T5 = (total $lane_a$ distance+d')/$speed_a$
7: T6 = $\frac{\text{vehicle length+inter vehicle distance}}{speed_a}$
8: N_a = $Count_a$
9: $T2 = T5 + T6 * (\left\lceil \frac{N_a}{2} \right\rceil - 1)$
10: **if** (occ cong ==1) **then**
11: $Normal_{green\_ab}$ = T3+T4*($\frac{CV_b}{2}$ -1)
12: **else**
13: $Normal_{green\_ab}$ = T1
14: **end if**
15: $Threshold_{green\_ab}$= T1+T2
16: **return** ($Normal_{green\_ab}$, $Threshold_{green\_ab}$)
17: **end procedure**

---

row of vehicles for crossing lanea and entering into laneb (Line 9). If the value of occ cong is 1 then the route a->b is congested. Hence, only CVb number of vehicles are allowed to enter into laneb and Normalgreen_ab is calculated for $\frac{CV_b}{2}$ – row of vehicles (Line 10, 11). Otherwise, $Count_a$ number of vehicles are allowed to enter into $lane_b$ and

$Normal_{green\_ab}$ is calculated for $\left\lceil \frac{Count_a}{2} \right\rceil^{th}$ row of vehicles which is equivalent to T1 (Line 12, 13). $Threshold_{green\_ab}$ depends on the time which is required for $Count_a$+ N_a) number of waiting vehicles in $lane_a$ to enter into $lane_b$ when the route a->b is in green signal. Hence, $Threshold_{green\_ab}$ is the summation of T1 and T2 (Line 15).

State 2: It can be observed from Table I that the routes 2->5 and 6->1 enter into green signal (GSR) whereas the routes 4->7 and 8->3 enter into red signal (RSR) when the state changes from State 2 to State 3. $lane_2$/($lane_6$) is IL1 and $lane_5$/($lane_1$) is OL1 for the route 2->5/(6->1). $lane_4$/($lane_8$) is IL2 and $lane_7$/($lane_3$) is OL2 for the route 4->7/(8->3). $Count_2$/($Count_6$) is $Count_{IL1}$ for the route 2->5/(6->1). Fog 12/(Fog 56) is associated with the route 2->5/(6->1) whereas Fog 34/(Fog 78) is associated with the route 4->7/(8->3). The functions of all these Fogs are elaborated below.

Retrieval Number: 100.1/ijrte.A59660510121
DOI: 10.35940/ijrte.A5966.0510121

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

Function of Fog 12/(Fog 56):

Calls Algorithm 1 (CONGESTION DETERMINATION(State 2, $lane_1$, 2->5))/(CONGESTION DETERMINATION(State 2, $lane_5$, 6->1)) to determine whether the route 2->5/ (6->1) is congested

-State 2 is the current state, $lane_1$ is the outgoing lane which is associated with Fog 12 and *lane* is the outgoing lane which is associated with Fog_56.

- Sets $Count_2$=$Count_a$,CV_1= $CV_{a'}$,CV_5=

CV_b, occ cong=Occurrence of congestion/ ($Count_6$= $Count_a$, CV_5= $CV_{a'}$, CV_1= CV_b, occ cong= Occurrence of congestion)

. Sends M Fog1 in the form (Fog 12, CV_1) / (Fog 56, CV_5) to Fog 56 associated with 6->1 / (Fog 12 associated with 2->5)

. Calls Algorithm 2 (*Computation_Normal*$_{green}$ *Threshold*$_{green}$ (2->5, d1, $Count_2$, CV_5, occ cong)) / (*Computation* _*Normal*$_{green}$ _ *Threshold*$_{green}$(6->1, d1, $Count_6$, CV_1, occ cong)) to compute $Normal_{green\_25}$, $Threshold_{green\ 25}$ / ($Normal_{green\ 61}$, $Threshold_{green\ 61}$) for the

route 2->5 / (6->1)

-Here d1 is used as 2->5 and 6->1 are straight routes. Sets $Normal_{green\ 25}$= $Normal_{green\ ab}$ / ($Normal_{green\ 61}$= $Normal_{green\ ab}$), $Threshold_{green\ 25}$= $Threshold_{green\ ab}$ /($Threshold_{green\ 61}$= $Threshold_{green\ ab}$.)

. Sends M Fog2 in the form (Fog 12, $Normal_{green\ 25}$, $Threshold_{green\ 25}$) / (Fog_56, $Normal_{green\ 61}$, $Threshold_{green\ 61}$) to the intersection controller

Function of Fog 34/(Fog 78):

. Calls Algorithm 1 (CONGESTION DETERMINATION (State 2,$lane_3$,4->7)) /(CONGESTION_DETERMINATION (State 2, $lane_7$, 8->3)) to determine whether the route 4->7 / (8->3) is congested

. Sets CV_3=$CV_{a'}$, occ cong= Occurrence of congestion /
(CV_7= $CV_{a'}$, occ cong =Occurrence_of_congestion)

. Sends M Fog1 in the form (Fog 34, CV_3) / (Fog 78, CV_7) to Fog 78 associated with 8->3 / (Fog 34 associated with 4->7)

. Sends M Fog3 in the form (Fog 34, occ cong) / (Fog 78, occ cong) to the intersection controller.

State 3: It can be observed from Table I that the routes4->1 and 8->5 enter into green signal (GSR) whereas the routes 2->5 and 6->1 enter into red signal (RSR) when the state changes from State 3 to State 4. $lane_4$/($lane_8$) isIL1 and $lane_1$/($lane_5$) is OL1 for the route 4->1/(8->5). $lane_2$/($lane_6$) is IL2 and $lane_5$/($lane_1$) is OL2 for the route 2->5/(6->1). $Count_4$/($Count_8$) is $Count_{IL1}$ for the route 4->1/(8->5). Fog 34/(Fog 78) is associated with the route 4->1/(8->5) whereas Fog 12/(Fog 56) is associated with the route 2->5/(6->1). The functions of all these Fogs areelaborated below.

Function of Fog 12/(Fog 56):

. Calls Algorithm 1 (CONGESTION DETERMINATION (State 3, $lane_1$, 2->5)) / (CONGESTION DETERMINATION (State 3, $lane_5$, 6->1)) to determine whether the route 2->5 / (6->1) is congested

-State 3 is the current state, $lane_1$ is the outgoing lane which is associated with Fog 12 and $lane_5$ is the outgoing lane which is associated with Fog 56

Sets CV_1= $CV_{a'}$, occ cong = Occurrence of congestion / (CV_5= $CV_{a'}$, occ cong=Occurrence of congestion)

Sends M Fog1 in the form (Fog 12, CV_1) / (Fog 56, CV_5) to (Fog 34 associated with 4->1, Fog 56 associated with 6->1) / (Fog 12 associated with 2->5, Fog 78 associated with 8->5)

Sends M Fog3 in the form (Fog_12, occ_cong) / (Fog 56,occ cong) to the intersection controller

Function of Fog 34/(Fog 78):

. Calls Algorithm 1 (CONGESTION DETERMINATION (State 3, $lane_3$, 4->1)) / (congestion determination(State 3, $lane_7$, 8->5)) to determine whether the route 4->1/(8->5) is congested

- $lane_3$ is the outgoing lane which is associated with Fog 34 and $lane_7$ is the outgoing lane which is associated with Fog 78

Sets. $Count_4$= $Count_a$, CV_3= $CV_{a'}$, CV_1= CV_b, occ cong= Occurrence of congestion / ($Count_8$= $Count_a$, CV_7= $CV_{a'}$, CV_5= CV_b, occ cong = Occurrence of congestion)

. Calls Algorithm 2 (Computation $Normal_{green}$ $Threshold_{green}$(4->1, d2, $Count_4$, CV_1, occ cong)) / (Computation $Normal_{green}$ $Threshold_{green}$(8->5, d2, $Count_8$, CV_5, occ cong)) to compute $Normal_{green\ 41}$, $Threshold_{green\ 41}$ / ($Normal_{green\ 85}$, $Threshold_{green\ 85}$) for the route 4->1/ (8->5)

-Here d2 is used as 4->1 and 8->5 are left turn routes.

.Sets $Normal_{green\ 41}$= $Normal_{green\ ab}$ / ($Normal_{green\ 85}$= $Normal_{green\ ab}$), $Threshold_{green\ 41}$= $Threshold_{green\ ab}$ / ($Threshold_{green\ 85}$= $Threshold_{green\ ab}$)

.Sends M Fog2 in the form (Fog 34, $Normal_{green\ 41}$, $Threshold_{green\ 41}$) / (Fog 78, $Normal_{green\ 85}$, $Threshold_{green\ 85}$) to the intersection controller

State 4: It can be observed from Table I that the routes2->7 and 6->3 enter into green signal (GSR) whereas the routes 4->1 and 8->5 enter into red signal (RSR) when the state changes from State 4 to State 1. $lane_2$/($lane_6$) isIL1 and $lane_7$/($lane_3$) is OL1 for the route 2->7/(6->3). $lane_4$/($lane_8$) is IL2 and $lane_1$/($lane_5$) is OL2 for the route 4->1/(8->5). $Count_2$/($Count_6$) is $Count_{IL1}$ for the route 2->7/(6->3). Fog 12/(Fog 56) is associated with the route 2->7/(6->3) whereas Fog 34/(Fog 78) is associated with the route 4->1/(8->5). The functions of all these Fogs areelaborated below.

Function of Fog 12/(Fog 56):

. Calls Algorithm 1 (CONGESTION DETERMINATION(State 4,$lane_1$, 2->7))/ (CONGESTION_ DETERMINATION(State 4, $lane_5$,6->3))

Retrieval Number: 100.1/ijrte.A59660510121
DOI: 10.35940/ijrte.A5966.0510121

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

to determine whether the route 2->7/ (6->3) is congested

-State 4 is the current state, $lane_1$ is the outgoing lane which is associated with Fog 12 and $lane_5$ is the outgoing lane which is associated with Fog 56

Sets $Count_2$= $Count_a$, CV_1= CV_a', CV_7= CV_b, occ cong= Occurrence of congestion / ($Count_6$= $Count_a$, CV_5= CV_a', CV_3= CV_b, occ cong= Occurrence of congestion)

Sends M Fog1 in the form (Fog 12, CV_1) / (Fog 56, CV_5) to Fog 34 associated with 4->1 / (Fog 78 associated with 8->5)

. Calls Algorithm 2 (*Computation Normal*$_{green}$ *Threshold*$_{green}$(2->7, d2, $Count_2$, CV_7, occ cong))/(*Computation Normal*$_{green}$ *Threshold*$_{green}$(6->3, d2,$Count_6$, CV_3, occ cong)) to compute Normal$_{green\ 27}$,Threshold$_{green\ 27}$ / (Normal$_{green\ 63}$, Threshold$_{green\ 63}$) for the

route 2->7 / (6->3)

- Here d2 is used as 2->7 and 6->3 are straight routes.

. Sets Normal$_{green\ 27}$= Normal$_{green\ ab}$ / (Normal$_{green\ 63}$= Normal$_{green_ab}$), Threshold$_{green\ _27}$= Threshold$_{green\ ab}$ / (Threshold$_{green_63}$= Threshold$_{green_ab}$)

. Sends M Fog2 in the form (Fog 12, Normal$_{green\ 27}$, Threshold$_{green\ 27}$) / (Fog 56, Normal$_{green_63}$, Threshold$_{green\ 63}$) to the intersection controller

Function of Fog 34/(Fog 78):

. Calls Algorithm 1 (CONGESTION DETERMINATION (State 4, $lane_3$, 4->1)) / (CONGESTION DETERMINATION(State 4, $lane_7$, 8->5)) to determine whether the route 4->1/ (8->5) is congested

- $lane_3$ is the outgoing lane which is associated with Fog 34 and $lane_7$ is the outgoing lane which is associated with Fog 78

. Sets CV_3= CV_a', occ cong =Occurrence of congestion /(CV_7= CV_a', occ cong=Occurrence of congestion)

. Sends M Fog1 in the form (Fog 34, CV_3)/ (Fog 78, CV_7)to Fog 56 associated with 6->3 / (Fog 12 associated with 2->7)

. Sends M Fog3 in the form (Fog 34, occ cong) / (Fog 78, occ cong) to the intersection controller

Function of intersection controller: It is elaborated in Algorithm 3.

Algorithm 3 Function of intersection controller

Input Two M Fog2 and two M Fog3

Output Actual duration of green signal in next GSRs,maximum waiting time of the vehicles in current GSRs

procedure
Reads *Threshold*$_{green}$ and *Normal*$_{green}$ from the two received M Fog2
Reads occ cong from the two received M Fog3
if value of the two received occ cong == 0 **then** /*current GSRs are not congested*/
Sets the maximum waiting time of the vehicles as the maximum of the two *Threshold*$_{green}$ values in current GSRs for the next state
else
Sets the maximum waiting time of the vehicles as the minimum of the two *Threshold*$_{green}$ values in current GSRs for the next state
end if
Sets the actual duration of green signal as the average of the two *Normal*$_{green}$ values in next GSRs
end procedure

Size of messages: The type id field in M V indicates that vehicle is the sender of this message and its value is 1. The size of the route id field is 4 bits as intersection has 12 routes. The maximum vehicle speed is assumed as 120 Km/hr and hence the size of the veh speed is 7 bits. So, the size of the message M V (Size M V) is 12 bits.

The size of the Fog identification field in M Fog1 is 2 bits as the intersection has 4 Fogs. The size of CV value in M Fog1 depends on the MAX CAP. During simulation MAX CAP is assumed as 200. Hence, the size of CV value is 8 bits and the size of M Fog1 (Size M Fog1) is 10 bits.

The size of both the *Normal*$_{green}$ and *Threshold*$_{green}$ fields in M Fog2 is assumed as 2 bytes. Hence, the size of M Fog2 (Size M Fog2) is 34 bits.

The occ cong field in M Fog3 indicates whether congestion occurs in the route or not. The value of this field is 1 ifcongestion occurs, otherwise it is 0. Hence, the size of M Fog3(Size M Fog3) is 3 bits.

B. Var 2:

Three scheduling algorithms are illustrated in Var 2. These three approaches are priority scheduling (Pr Sch), shortest job first scheduling (SJ Sch) and round robin scheduling (RR Sch). Each approach has four phases corresponding to the green signal in the four incoming lanes. The sequence of execution of these four phases is determined dynamically in all the three scheduling approaches and the execution of the four phases creates a cycle. The vehicles in a lane can go straight (d1 in Fig. 1) or left (d2 in Fig. 1) when the lane gets green signal.

Function of a vehicle:

The function of v^{th} vehicle (V_v) after entering into an incoming lane in k^{th} cycle is elaborated in this section. The On Board Unit (OBU) of V_v determines its position usingGPS, executes Algorithm 4 to generate a message M V.

Algorithm 4 Function of V_v

Input Position of V_v

Output M_V

procedure
Computes its distance (d) from the intersection
Adds d1 or d2 with d to compute the distance Dist
Compute the time ($Time_v$) required to cross Dist as $\frac{Dist}{start\_speed}$
Generates M_V in the form ($Time_v$)
Sends M_V to Fog associated with its current lane
end procedure

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

Function of Fogs in k^{th} cycle: Fog 12, Fog 34, Fog 56, Fog 78

- Receive M_Vs from the vehicles in $lane_2$, $lane_4$, $lane_6$, $lane_8$
- Store the received M_Vs in $Queue_2$, $Queue_4$, $Queue_6$, $Queue_8$
- Execute Algorithm 5 to determine the sequence and duration of green signal for the lanes for $(k + 1)^{th}$ cycle.

Algorithm 5 Function of Fog_12, Fog_34, Fog_56, Fog_78 in k^{th} cycle

Input M_Vs
Output Sequence and duration of green signal

1: **procedure**
2: Count the number of M_Vs in $Queue_2$, $Queue_4$, $Queue_6$, $Queue_8$ as V_2, V_4, V_6, V_8 respectively
3: Read time from V_2, V_4, V_6, V_8 M_Vs respectively
4: Compute the maximum time G_2 by comparing V_2 number of time values, G_4 by comparing V_4 number of time values, G_6 by comparing V_6 number of time values, G_8 by comparing V_8 number of time values respectively
5: Send a message (M_Fog4) in the form [Fog_12, V_2, G_2], [Fog_34, V_4, G_4], [Fog_56, V_6, G_6], [Fog_78, V_8, G_8] to the other three fogs
6: Call procedure_PR for PR_Sch, procedure_SJ for SJ_Sch, procedure_RR for RR_Sch
7: **end procedure**

Algorithm 6 procedure_PR

1: **procedure**
2: Calculate the priority of $lane_2$ (Prio_$lane_2$), $lane_4$ (Prio_$lane_4$), $lane_6$ (Prio_$lane_6$) and $lane_8$ (Prio_$lane_8$) as $\frac{V_2}{V_2+V_4+V_6+V_8}$, $\frac{V_4}{V_2+V_4+V_6+V_8}$, $\frac{V_6}{V_2+V_4+V_6+V_8}$, $\frac{V_8}{V_2+V_4+V_6+V_8}$
3: Arrange the lanes in the descending order of their priority in a list to determine the sequence of green signal of the four lanes for $(k + 1)^{th}$ cycle
4: Compute cycle duration as G_2+G_4+G_6+G_8
5: **end procedure**

For example, let the list in Procedure_PR is (Prio_$lane_2$, Prio_$lane_8$, Prio_$lane_4$, Prio_$lane_6$). So first Fog_12 schedules green signal for the routes 2->5 and 2->7 for the duration G_2, then Fog 78 schedules green signal for the routes 8->3 and 8->5 for the duration G_8, then Fog 34 schedules green signal for the routes 4->7 and 4->1 for the duration G_4, then Fog 56 schedules green signal for the routes 6->1 and 6->3for the duration G_6.

Algorithm 7 procedure_SJ

1: **procedure**
2: Arrange the lanes in the ascending order of green signal duration G_2, G_4, G_6, G_8 in a list to determine the sequence of green signal of the four lanes for $(k + 1)^{th}$ cycle
3: Compute cycle duration as G_2+G_4+G_6+G_8
4: **end procedure**

For example, let the list in Procedure_SJ is (G_2, G_8, G_4, G_6). So first Fog 12 schedules green signal for the routes 2->5 and 2->7 for the duration G_2, then Fog 78 schedules green signal for the routes 8->3 and 8->5 for the duration G_8, then Fog 34 schedules green signal for the routes 4->7 and 4->1 for the duration G_4, then Fog 56 schedules green signal for the routes 6->1 and 6->3 for the duration G_6.

Algorithm 8 procedure_RR

1: **procedure**
2: Arrange the lanes in the descending order of their priority in a list to determine the sequence of green signal of the four lanes in $(k + 1)^{th}$ cycle
3: Calculate time quantum T_{avg} as $\frac{G_2+G_4+G_6+G_8}{4}$
4: Compute cycle duration as 4* T_{avg}
5: **end procedure**

For example, let the list in Procedure_RR is (Prio_$lane_2$, Prio_$lane_8$, Prio_$lane_4$, Prio_$lane_6$). So first Fog_12 schedules green signal for the routes 2->5 and 2->7 for the duration T_{avg}, then Fog_78 schedules green signal for the routes 8->3 and 8->5 for the duration T_{avg}, then Fog 34 schedules green signal for the routes 4->7 and 4->1 for the duration T_{avg}, then Fog_56 schedules green signal for the routes 6->1 and 6->3 for the duration T_{avg}.

In PR_Sch, the green signal is given to the lane which has the highest priority. So, even if there is less number of vehicles in the less priority lanes, they have to wait until the lanes receive the green signal which increases the waiting time. In SJ_Sch, instead of priority, the green signal is given to the lane which requires less duration of green signal. It increases the waiting time of vehicles in the lanes where the green signal duration is high. Waiting time decreases in RR_Sch since it considers both priority and green signal duration.

Size of messages: The size of the $Time_v$ field in M_V of V_v is assumed as 2 bytes. Hence, the size of M_V (Size_M_V) is 16 bits. The size of V_2, V_4, V_6 and V_8 is assumed as 8 bits. The size of G_2, G_4, G_6 and G_8 is assumed as 2 bytes. Hence, the size of M_Fog4 (Size_M_Fog4) is (2+8+16) bits.

C. *Var 3:*

The agent knows the sets of possible states and actions. Each lane is in a particular level of vehicles depending upon the number of vehicles at any instant of time. Five levels such as - very low (VL), low (L), medium (M), high (H) and very high (VH) are considered in Var 3. The level of vehicles in a lane is VL,L,M,H,VH if the number of vehicles in that lane is (0-20)%, (21-40)%, (41-60)%, (61-80)%, (81-100)% of the total number of vehicles respectively across the intersection.

A state is a tuple of four levels corresponding to four incoming lanes. For example, let the current state is s1 <L, M, L, L > i.e. the level of vehicles in $lane_2$, $lane_4$, $lane_6$, $lane_8$ are L, M, L, L respectively. Now for a particular level of vehicles in $lane_2$, the number of level of vehicle in $lane_4$, $lane_6$, $lane_8$ is 5^3 and the number of possible states is 5^3. So for five different level of vehicle in $lane_2$, the number of possible states is 5x5^3 (=625).

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

Var 3 considers four actions - green signal in $lane_2$, $lane_4$, $lane_6$ and $lane_8$ as (g2,g4,g6,g8). The sequence of execution of these four actions is determined dynamically by the agent.

The execution of any four actions one after another creates a cycle. For example, the sequence (g2,g4,g6,g8) creates a cycle. The agent maintains a Q-table (Fig. 2) to store the records corresponding to 625 number of states. So the Q-table has 625 number of records. Each record in Q-table has four pair of attributes corresponding to four possible actions. These attributes are G_2, G_4,G_6,G_8, Q_2, Q_4,Q_6, Q_8 where G_2, G_4,G_6, G_8 are the duration of action g2, g4, g6, g8 and Q_2, Q_4,Q_6, Q_8 are Q-value of action g2, g4, g6, g8.

State	G_2	Q_2	G_4	Q_4	G_6	Q_6	G_8	Q_8

Fig. 2. Q-table

Algorithm 9 Congestion Control Algorithm

Input Current state s1

Output Updated Q-table

1: **procedure**
2: Calculates total waiting time of vehicles in each lane
3: Calculates the average waiting time (avg_wt) of the vehicles in each lane as $\frac{total\ waiting\ time\ of\ vehicles\ in\ a\ lane}{number\ of\ vehicles\ present}$
4: Calculates current waiting time as the average of avg_wt of the four lanes
5: Searches Q-table for the record (Rec_s1) correspond-ing to the current state s1
6: Reads the attributes (G_2, Q_2), (G_4, Q_4), (G_6, Q_6), (G_8, Q_8) from Rec s1
7: Sets i=2 and p=0
8: **if** i $\leq$ 8 **then**
9: **if** (G_i,Q_i) pair in Rec_s1 is empty **then**
10: Performs the action g_i (g) for duration δ_i (D)
11: Calls Algorithm10 FUNC(g,D) at the end of D
12: Sets G_i = New_D and Q_i= New_Q
13: Inserts (G_i, Q_i) in the position of i^{th} pair in Rec_s1
14: p=p+1
15: Break
16: **else**
17: i=i+2
18: Go to step 7
19: **end if**
20: **end if**
21: **if** p=0 **then**
22: Compares Q_2, Q_4, Q_6, Q_8 to find the maximumQ-value (max_Q)
23: Reads the duration (D) from the pair of attribute corresponding to max_Q
24: Performs the action (g) corresponding to the pair
25: Calls Algorithm 10 FUNC(g,D) at the end of D
26: Sets D = New_D and max_Q=New_Q
27: Updates (D, max_Q) pair of attribute in Rec_s1
28: **end if**
29: **end procedure**

The agent senses the environment to determine the number of vehicles in each lane across the intersection, computes the current state, executes the congestion control algorithm (Algorithm 9) and computes the next state.

Algorithm 10 Calculation of duration and Q-value

1: **procedure** FUNC(g,D)
2: Reads the number of vehicles in each lane across the intersection to compute next state (s2)
3: Calculates avg_wt of the vehicles in each lane in s2
4: Calculates next waiting time as the average of avg_wt of the four lanes
5: **if** next waiting time > current waiting time **then**
6: Receives penalty from environment
7: Decreases D
8: Calculates D (New_D) using Equ. 2
9: Calculates Q-value (New_Q) corresponding to action g in s1 using Equ. 4
10: **return** (New_D, New_Q)
11: **else**
12: Receives reward from environment
13: Increases D
14: Calculates D (New_D) using Equ. 3
15: Calculates Q-value (New_Q) corresponding to action g in s1 using Equ. 4
16: **return** (New_D, New_Q)
17: **end if**
18: **end procedure**

Calculation of New_D: The value of D increases in case of reward and decreases in case of penalty. The difference between the next waiting time and current waiting time is considered as diff. In case of penalty, diff is positive and hence D is reduced (diff/3) as there are three red signal lanes at a time.

$$New_D = D - \frac{diff}{3} \qquad (2)$$

In case of reward, diff is negative. New_D is calculated depending on the number of vehicles and the distance (Dist)to be covered by the vehicles in the lanes for crossing the intersection in green signal. T7 is the time taken by the first row of vehicles in an incoming lane to cross Dist. So,T7 is (Dist/ start speed). T4 is the time to cross the inter vehicle distance and the length of the vehicle in the front row as discussed in section 3.1. For example, let the number of vehicles in $lane_2$ is V_2 and so the number of rows of vehicles in $lane_2$ is $\left\lceil\frac{V_2}{2}\right\rceil$. New_D is the new duration of green signal in $lane_2$ (G_2) and it is the time which is required by the vehicles $\left\lceil\frac{V_2}{2}\right\rceil^{th}$ row of $lane_2$ to cross the intersection when

$lane_2$ gets green signal. New_D is calculated using Equ. 3.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

Calculation of Q-value: The agent calculates the Q-value [25] corresponding to the action g in s1 using Equ 4.

$$New\ Q \equiv (1-\alpha)Q(s1, g) + \alpha[R + \gamma * Q(s2, g_{max\ Q})] \quad (4)$$

where,

α = learning rate (0 < α < 1), Q(s1,g) = Q-value ofthe action g corresponding to s1, R=reward or penalty, γ= discounting factor (0< γ <1) $g_{max\ Q}$= action g which has the maximum Q-value, $Q(s2, g_{max\ Q})$ = Q-value of the action, $g_{max\ Q}$ corresponding to s2

IV. PERFORMANCE ANALYSIS

The performance of Var 1, Var 2 and Var 3 is studied qualitatively, theoretically and quantitatively. The qualitative performance is studied in terms of communication overhead (COMM OH), storage overhead (STO OH) and computation overhead (COMP OH). The theoretical and quantitative per- formances are studied in terms of waiting time. Waiting time isthe time during which the vehicles are waiting in the red signal. The increase in number of vehicles causes an increase inwaiting time which in turn escalates traffic congestion. Hence the variation of waiting time is studied by varying the number of vehicles. SUMO (Simulation of Urban MObility) is used to construct the road network, to generate traffic flows, mobility of the vehicles and to control the traffic lights whereas NS3 is used as a discrete-event network simulator. Traffic Control Interface is used in SUMO which allows to access vehicle speed, current traffic light etc. during the simulation.

A. *Qualitative performance:*

In this section COMM OH, STO OH and COMP OH ofall the three variants are studied.

1) COMM OH: **Var 1:** COMM OH in a state includes transmission of M_V from vehicles to Fogs, transmission ofM Fog1 to Fogs, transmission of M Fog2 and M Fog3 to the intersection controllers to decide the green signal duration of the routes in the next state.

Fog 12, Fog 34, Fog 56 and Fog 78 receive M V from V_2, V_4, V_6 and V_8 number of vehicles respectively. Hence, COMM OH due to the transmission of M_V is Size_M_V *(V_2 + V_4+ V_6+ V_8) bits.

Four M_Fog1 are transmitted by Fogs. So, the COMM OH due to the transmission of M Fog1 is (Size M Fog1*4) bits.Two Fogs send M Fog2 and other two Fogs send M Fog3 to the intersection controller. So, the COMM OH due to the transmission of M Fog2 is (Size M Fog2*2) and M Fog3 is (Size M Fog3*2) bits.

COMM OH of all the four states (per cycle) in Var 1 is 4*(Size M V *(V_2 + V_4+ V_6+ V_8) + (Size M Fog1*4) + (Size M Fog2*2)+ (Size M Fog3*2)) bits.

Var 2: COMM OH per cycle includes transmission of M V from vehicles to Fogs, transmission of M Fog4 to the Fogs to decide the green signal duration of the routes. COMM OH due to the transmission of M_V is Size_M_V *(V_2 + V_4+ V_6+ V_8) bits. Each Fog sends three M Fog4 to the other three Fogs. COMM OH due to the transmission ofM_Fog4 is (Size M Fog4*3*4) bits.

So, in Var 2 COMM OH per cycle is Size M V *(V_2 + V_4+ V_6+ V_8) + (Size M Fog4 * 3 * 4) bits.

Var 3: COMM OH is zero as the agent is not communicating with the other entities of the network.

2) STO OH: **Var 1:** STO OH includes storage of M V and M Fog1 at Fogs, storage of M_Fog2 and M_Fog3 at the intersection controller.

Fog 12, Fog 34, Fog 56 and Fog 78 store Table III. Table III has five records, each has two attributes- Range of number of vehicles and Range of speed. Both Range of number of vehicles and Range of speed attribute contain two integer elements as veh min, veh max and speed min, speed max respectively. So the size of each record in Table III is 4*sizeOf(int) bits and STO OH to store at four Fogs is 4*5*4*sizeOf(int) bits.

Fog 12, Fog 34, Fog 56 and Fog 78 store V_2,V_4,V_6 and V_8 number of M Vs respectively. So, STO OH to store M V is Size M V * (V_2+V_4+V+V_8) bits.

Each Fog stores four M Fog1. The intersection controller stores two M Fog2 and two M Fog3.

In Var 1, STO OH per cycle is (4*5*4*sizeOf(int)+ Size M V*(V_2 + V_4+ V_6+ V_8) + (Size M Fog1 * 4) + (Size M Fog2*2)+ (Size M Fog3*2)) bits.

Var 2: STO OH per cycle includes storage of M_V andM Fog4 at Fogs. Fog 12, Fog 34, Fog 56 and Fog 78 store V_2, V_4, V_6 and V_8 number of M Vs. Each Fog stores threeM Fog4.

In Var 2 STO OH per cycle is Size M_V *(V_2 + V_4+ V_6+ V_8) + (Size M Fog4 *3*4)bits.

Var 3: STO OH is due to the maintenance of Q-table (Fig. 2). The level in a state is represented either by using a single (like L, M, H) or two characters (like VL, VH). So the size of each level is considered as 2*sizeOf(char) and the size of each state is 4*2*sizeOf(char) bits. For storing duration and Q-value of 4 actions, 8 integer elements are required.The size of the 8 integer elements is 8 * sizeOf(int) bits. So, STO OH is 625 * (4*2*sizeOf(char) +8 * sizeOf(int)) bits or 1,20,000 bits. STO OH is constant as it is independent of the number of vehicles.

3) COMP OH: **Var 1:** COMP OH of a state is as follows

- Counts the number of vehicles in $lane_2$, $lane_4$, $lane_6$, $lane_8$ with COMP OH O(V_2), O(V_4), O(V_6), O(V_8) respectively
- Estimates the capacity of lanes with COMP OH O(1)
- Predicts whether the routes are congested with COMP_OH O(1)
- Estimates $Normal_{green}$, $Threshold_{green}$ with COMP OH O(1)
- Estimates final duration of green signal for the green signal routes and maximum waiting time of vehicles in the red signal routes in the next state with COMP OH O(1)

In Var 1 COMP OH per cycle is 4*(O(V_2)+ O(V_4)+ O(V_6)+ O(V_8)+ O(1))

Retrieval Number: 100.1/ijrte.A59660510121
DOI: 10.35940/ijrte.A5966.0510121

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

Var_ 2 : COMP_ OH per cycle is as follows:

- Computes distance from the intersection and Dist with COMP _OH O(1)
- Computes the time required by the vehicles to enter tothe desired route after crossing d1 or d2 with COMP OH O(1)
- Count the number of M_ Vs in the queue with COMP OHO(V_2), O(V_4), O(V_6), O(V_8)
- Reads time values from M_Vs with COMP OH O(V_2), O(V_4), O(V_6), O(V_8)
- Compares V_2, V_4, V_6, V_8 to find G_2, G_4, G_6, G_8 with COMP OH O(V_2), O(V_4), O(V_6), O(V_8)
- Calculates the priority of each lane and arrange the lanes in descending order of their priority with COMP OH O(1) in PR Sch _
- Arranges the lanes in ascending order of G_2, G_4, G_6, G_8

values with COMP OH O(1) in SJ Sch

- Arranges the lanes in descending order of their priorityand calculates T_{avg} as the time quantum with COMP OH O(1) in RR Sch

So, COMP_ OH in PR Sch, SJ Sch and RR Sch per cycleis O(V_2)+ O(V_4)+ O(V_6)+ O(V_8)+O(1).

COMP OH in Var 3: The agent

Reads the number of vehicles as V_2, V_4, V_6, V_8 for cal- culating the current state _with COMP OH O(V_2+V_4+V_6+V_8) Calculates avg wt of each lane in the current state with

COMP OH O(1)

Calculates current waiting time with COMP OH O(1)

Searches the Q-table for the current state among 625 statesto find Rec s1 with COMP OH O(1)

Reads four pair of attributes from Rec s1 with COMP _OH O(1)

Chooses an action from the Q-table with COMP OH O(1)Calculates the next state with COMP_ OH

O(V_2+V_4+V_6+V_8)

Calculates avg wt of each lane with COMP _OH O(1) Calculates next waiting time with COMP_ OH O(1) Compares current waiting time and next waiting time with

COMP OH O(1)

Calculates duration of action (New D) and Q-value (New_Q) with COMP _OH O(1)

Inserts or updates (New _D, New_ Q) in the Q-table

corresponding to state s1 with COMP OH O(1)

In Var 3 COMP OH per action is O(V_2+ V_4+ V_6+ V_8)+

O(1) and for four actions in a cycle is 4*(O(V_2+ V_4+ V_6+V_8)+ O(1)).

The comparison of COMM OH, STO OH and COMP OHof the three variants is depicted in Table IV.

COMM OH is less in Var 2 than Var 1 as observed from

Table IV and zero in Var 3 as discussed in section IV.STO OH is less in Var 1 than Var 2 but constant in_ Var 3

Table Iv Comparison Of Comm Oh, Sto Oh And Comp Oh

-2*Number of vehicles	COMM Var 1	OH (bits) Var 2	STO_OH Var 1	(bits) Var 2	C Var 1	MP_OH Var 2	Var 3
200	10056	2712	10376	2712	199	46	39
250	12456	3312	12776	3312	197	48	45
300	14856	3912	29576	3912	198	69	47
600	29256	7512	29576	7512	192	120	79
750	36456	9312	36776	9312	196	131	83

(1,00,000 bits) as discussed in section IV. COMP OH inVar_1 is more than Var_ 2 and Var_ 3.

B. Theoretical Analysis :

Queuing system is a study of long waiting lines or queues. It is denoted by Kendall's notation [26] which is representedas A/B/C/D/E/F, where,

- A is arrival time distribution
- B is the service time distribution C is the number of
- servers
- D is the maximum number of customers allowed in the system including those in service
- E is size of population from which the customers
- come F represents the queuing discipline that is followed

Queuing system is used to estimate the waiting time of the queues [26]. The vehicles in the lanes form queues in red signals to cross the intersection. In Var 1, Var 2 and_ Var 3

- A is poisson or markov (random) arrival time distributionas the arrival time of vehicles is random
- B is also markov service time distribution as the durationof green signal is dynamic
- C is the number of intersection controller
- D is the maximum number of vehicles that can be allowedto enter into the lanes of an intersection
- E is the size of vehicle population from where vehicles can come and enter into the lanes of an intersection. So, It is assumed as infinite.
- F is queuing discipline. It is First Come First Serve (FCFS) as the vehicle which enters into a green signal lane first, goes out from that lane first.

The average waiting time (W_q) of vehicles is calculated (Equ. 5) by considering the three variants as M/M/C FCFS system. The variation of W_q with the number of vehicles for Var 1, PR Sch, SJ Sch, RR Sch and Var 3 is estimated in three different scenarios.

$$W_q = \frac{p^{c+1} * (1-p)}{(C-1)!(C-p)^2 * \lambda} \qquad (5)$$

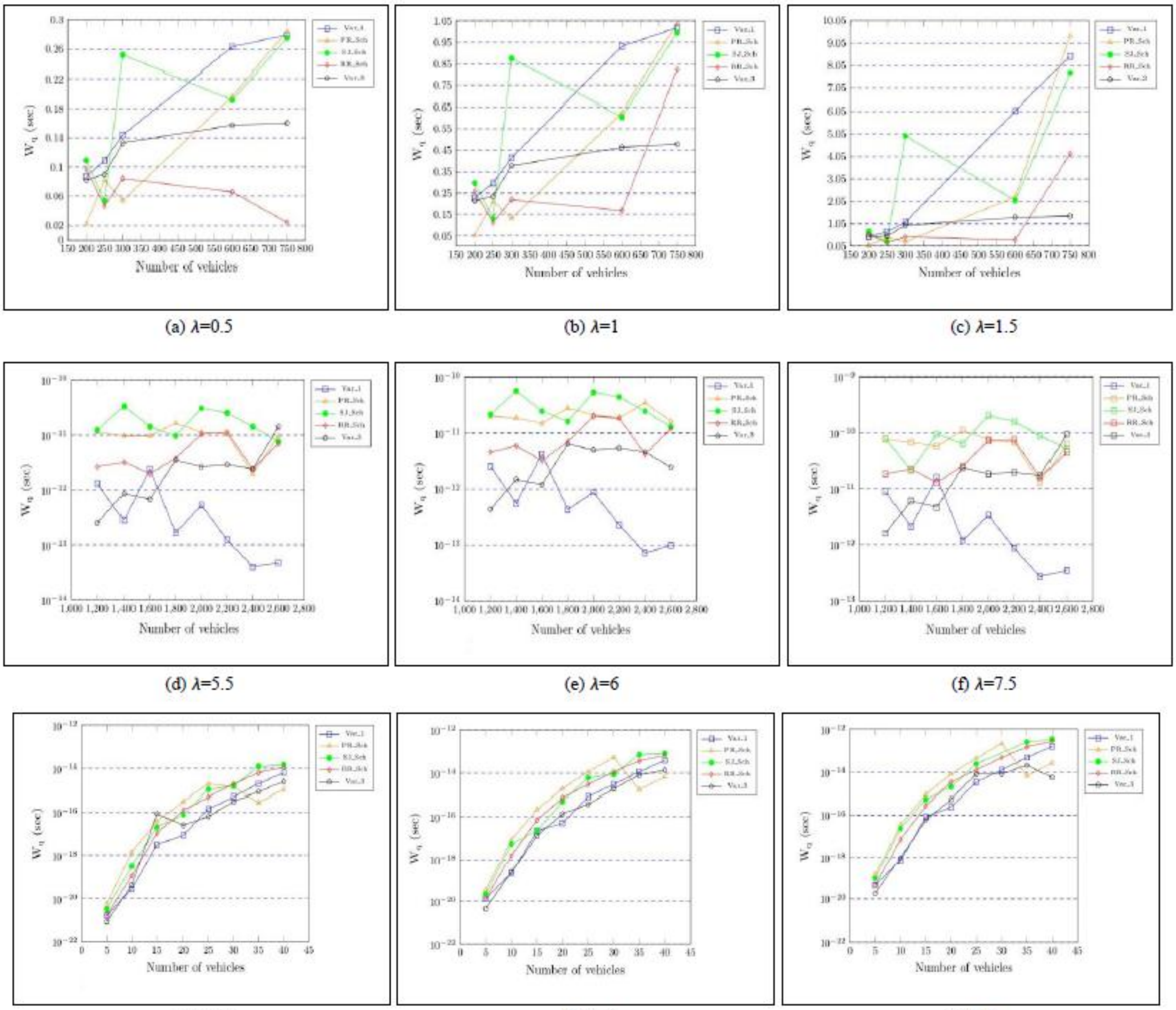
where p is $\frac{\lambda}{\mu * C}$, λ is arrival rate and μ is service rate (Equ. 6).

The simulation experiment is conducted to determine the variation of μ depending upon the three different values of $\lambda$and the number of vehicles in each scenario.

$$\mu = \frac{total\ simulation\ time}{number\ of\ vehicles\ across\ the\ intersection} \qquad (6)$$

Retrieval Number: 100.1/ijrte.A59660510121
DOI: 10.35940/ijrte.A5966.0510121

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication



(a) λ=0.5 (b) λ=1 (c) λ=1.5

(d) λ=5.5 (e) λ=6 (f) λ=7.5

(g) λ=7 (h) λ=9 (i) λ=11

Fig. 3. W_q vs number of vehicles

In the first scenario the network has a single intersection. The intersection has 4 pair of lanes (C=4). The variation of μ is 1.6 to 4.93 for λ equal to 0.5, 1, 1.5 and for the variationof the number of vehicles from 200 to 750 as in [16].

Fig 3a- Fig 3c show the variation of W_q vs. vehicles for λ=0.5, 1, 1.5 respectively. It can be observed that W_q is lessin RR Sch than Var 1, PR Sch, SJ Sch and Var 3.

In the second scenario the network has 6 intersections. Each intersection has 4 pair of lanes (C=4). The 6 intersections are comprised of 34 lanes. The variation of μ is 8.64 to 22.38 for λ equal to 5.5, 6, 7.5 and for the variation of the number of vehicles from 1200 to 2600 as in [27].

Fig 3d- Fig 3f show the variation of W_q vs. vehicles for λ=

5.5, 6, 7.5 respectively. It can be observed that W_q is less inVar 1 than PR Sch, SJ Sch, RR Sch and Var 3.

In the third scenario the network has 7 intersections. Each intersection has 4 bi-directional lanes (C=4). The 7 intersections are comprised of 40 lanes. The variation of μ is 11.22 to 90 for λ equal to 7,9,11 and for the variation of the numberof vehicles from 5 to 40 as in [28].

Fig 3g-Fig 3i variation of W_q vs. vehicles for λ = 7,9,11 respectively. It can be observed that W_q is less in Var 3 than Var 1, PR Sch, SJ Sch and RR Sch.

C. Quantitative Performance :

In this section, the quantitative performance of Var 1, Var 2and Var 3 is elaborated.

1) Simulation environments and result: The simulation ex- periment is conducted to observe the variation of waiting time (Wt) by varying the number of vehicles and compare with the existing schemes [16], [27], [28]. The total simulation timeis divided into some intervals. The arrival rate of vehicles in each interval is random and hence the waiting time ofvehicles in the intervals increases or decreases randomly.

Retrieval Number: 100.1/ijrte.A59660510121
DOI: 10.35940/ijrte.A5966.0510121

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

Hence the simulation experiment is also conducted to observe the variation of waiting time with intervals and compare with the existing schemes [29], [30]. In Var 3, Dist is assumed as (d1+d2)/2.

First experiment is conducted for observing the variation of Wt with the number of vehicles for Var 1, PR Sch, SJ Sch, RR Sch, Var 3 and [16].

Second experiment is conducted for observing the variation of Wt with the number of vehicles for Var 1, PR Sch,SJ Sch, RR Sch, Var 3 and [27].

Third experiment is conducted for observing the variation of Wt with the number of vehicles for Var 1, PR Sch, SJ Sch, RR Sch, Var 3 and [28].

Fourth experiment is conducted for observing the variationof Wt with interval for Var 3 and [29].

Fifth experiment is conducted for observing the variation of Wt with interval for Var 3 and [30].

First experiment: It is conducted in an intersection with

4 pair of lanes in the presence of 200 to 750 vehicles as considered in [16]. The length of each lane is 1 Km. The length of the intersection is 200 meter. The maximum speed limit of vehicles is 60Km/hr and inter vehicle distance is assumed as 2.5m.

Fig 4a shows the plot of Wt vs. the number of vehicles for PR Sch,SJ Sch and RR Sch. Fig 4b shows the plot of Wt vs. the number of vehicles for Var 1,RR Sch,Var 3 and [16].

Second experiment: It is conducted in 6 intersections comprised of 34 lanes in the presence of 1200 to 2600 vehicles as considered in [27]. The speed of the vehicles varies between 10 Km/hr to 45 Km/hr and the length of lanes varies between 500m to 1500m as considered in [27].

Fig 4c shows the plot of Wt vs. the number of vehicles for PR Sch,SJ Sch and RR Sch. Fig 4d shows the plot of Wt vs. the number of vehicles for Var 1, RR Sch, Var 3 and [27].

Third experiment: It is conducted in 7 intersections with bi-directional lanes in the presence of 5 to 40 vehicles as considered in [28]. The speed of the vehicles varies between the range 8.5m/s to 14m/s and the length of the intersectionis 20m as considered in [28].

Fig 4e shows the plot of Wt vs. the number of vehicles for PR Sch, SJ Sch and RR Sch. Fig 4f shows the plot of Wt vs. the number of vehicles for Var 1, RR Sch and, Var 3 and [28].

Fourth experiment: It is conducted in two intersections in the presence of 3200 and 6000 vehicles as considered in [29].

Fig 4g and Fig 4h show the plot of Wt vs. interval for Var 3 and [29] for 3200 and 6000 number of vehicles respectively. **Fifth experiment:** It is conducted in one intersection with four 1000 feet bi-directional lanes and the numbers of vehicles entering each lane per hour is almost 1000 as considered in [30].

Fig 4i shows the plot of Wt vs. interval for Var 3 and [30]. **Observation from simulation results:** It can be observed from Fig 4a, Fig 4c, Fig 4e that Wt in RR Sch is much less than PR Sch and SJ Sch as discussed in section III(b).

It can be observed from Fig 4b that Wt increases with the number of vehicles for Var 1, RR Sch,Var 3 and [16] but Wtin Var 3 is much less than [16],Var 1 and RR Sch.

It can be observed from Fig 4d that Wt increases with the number of vehicles for Var 1,RR Sch,Var 3 and [27] but it is less in Var 3 than [27],Var 1 and RR Sch.

It can be observed from Fig 4f that Wt increases with the number of vehicles for Var 1, RR Sch, Var 3 and [28] but it is less in RR Sch than [28], Var 1 and Var 3.

It can be observed from both Fig 4g and Fig 4h that Wt is less in Var 3 than [29].

It can be observed from Fig 4i that Wt is less in Var 3 than [30].

Discussion of results The waiting time in theoretical result (Fig 3a-Fig 3i) is much less than the waiting time in simulation result (Fig 4a, Fig 4b-Fig 4e, Fig 4f). In the theoretical result, the service rate of each vehicle is more or less same. But,in simulation scenario, the service rate of each vehicle is not same. Some vehicles can pass the green signal at one go, but some vehicles have to wait in the red signal. Thus, the service rate of vehicles differs.

In [16] the distribution of vehicles is assumed as same in theopposite pair of routes and the number of vehicles going fromsouth to north and north to south is assumed as much higher when the total number of vehicles is 600. Such assumptionsreduce Wt to 13 secs for 600 vehicles. When Wt of vehiclesin a lane reaches the threshold, green signal is set to that lane. If the number of vehicles is more, all waiting vehicles cannot pass the lane at one go. Hence, some vehicles have to waitagain for the next green signal which increases Wt of vehicles. Both in [27], [28] traffic congestion is controlled by forming cluster of vehicles. In [27] the vehicles which are waiting in alane form standing cluster whereas the new vehicles entering into the same lane in red signal form moving cluster. Hencethe number of vehicles waiting in red signal increases whichcauses an increase in Wt in [27]. In [28], the duration of greensignal in a lane depends upon the size of cluster. If the size ofthe cluster in one lane is large, the vehicles in the other lanes have to wait in the red signal which increases Wt in [28].

In Var 1, the upper bound of Wt i.e. the threshold value is calculated dynamically for the waiting vehicles. While calculating this threshold, both the number of vehicles present in the red signal lane and the occurrence of congestion in the green signal lane are considered. Thus, the threshold maintains a balance between the green signal lane and red signal lane,so that no vehicles have to wait for a long time. This allows

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

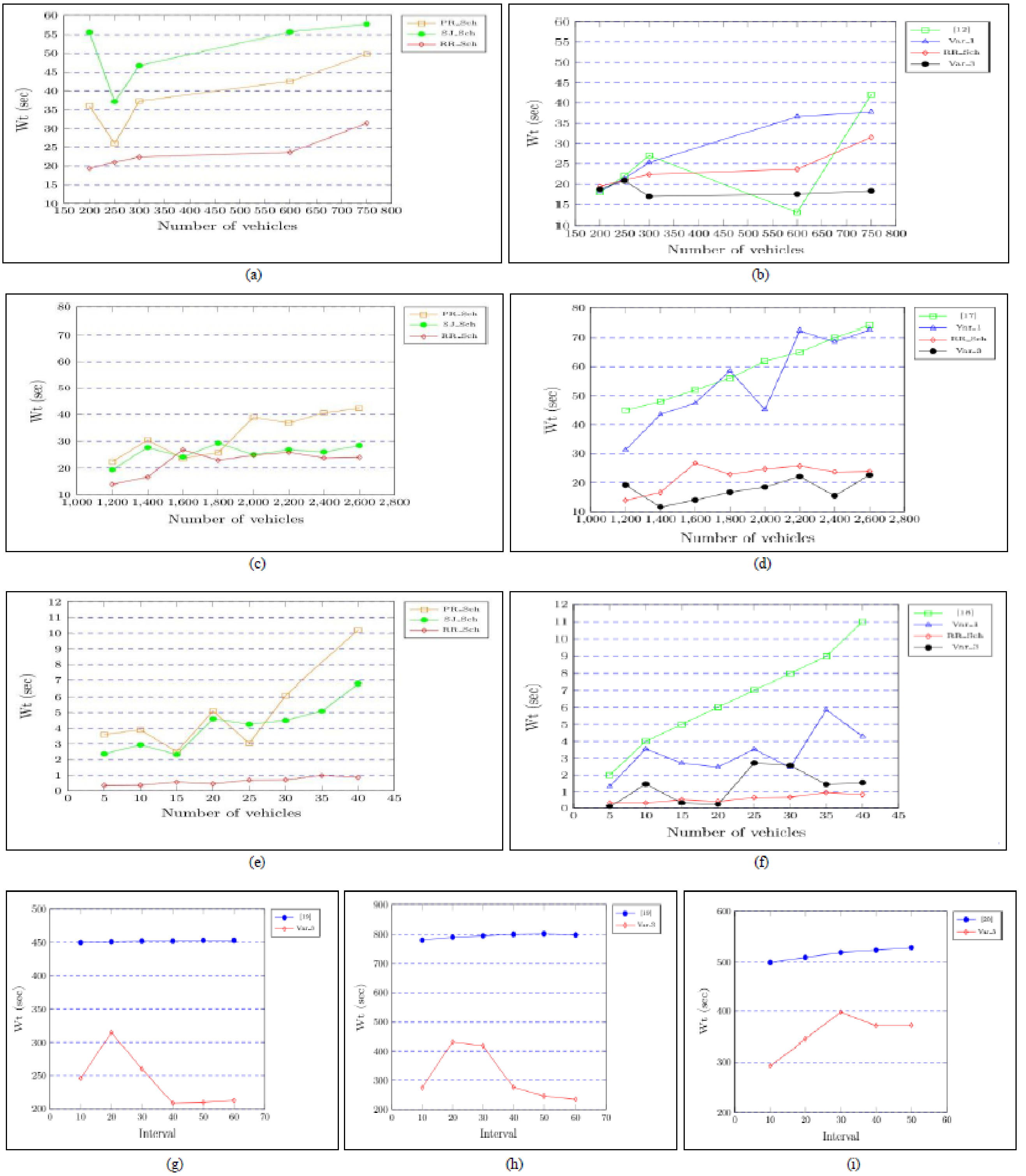


Fig. 4. Wt vs Number of vehicles

sufficient number of vehicles to pass the lane which helps to decrease Wt than [16].

In Var_1, the green signal is not on in all the routes corresponding to a particular incoming lane simultaneously as discussed in section III.1. This increases Wt in Var 1. In RR Sch, green signal is on in all the routes of an incoming lane simultaneously unlike Var 1 which decreases Wt in RR Sch than Var 1. RR Sch schedules green signal to the lanes as per their priority and maintains a fixed time quantum which may not be sufficient for the lane having more number of vehicles to cross the intersection or the arc distance in one go. Var 3 schedules green signal to the lanes dynamically in contrast to [29] and [30] depending upon the number of vehicles and the action in the current state. So, the scheduling criterion is much more realistic in Var 3 than RR Sch, [29] and [30]. It helps to mitigate Wt in Var 3 than RR Sch, [29] and [30].

Retrieval Number: 100.1/ijrte.A59660510121
DOI: 10.35940/ijrte.A5966.0510121

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

V. CONCLUSION AND FUTURE WORK

Two vehicle dependent and one environment dependent Fog computing oriented schemes for controlling traffic congestion in ITS are analyzed in the present work.

The robustness and efficacy of the schemes are compared with each other qual- itatively and quantitatively. All the three proposed schemes outperform the existing schemes in terms of waiting time of vehicles.

The simulation experiment may be conducted in real time environment to compare the performance of the three proposedschemes.

REFERENCES

1. Cynthia Jayapal and S Sujith Roy. Road traffic congestion management using vanet. In International Conference on Advances in Human Machine Interaction (HMI), pages 1–7. IEEE, 2016.
2. Ananya Paul and Sulata Mitra. Deep Reinforcement Learning based Traffic Signal optimization for Multiple Intersections in ITS. In In- ternational Conference on Advanced Networks and Telecommunications Systems (ANTS), pages 1–6. IEEE, 2020.
3. En zhan Zhang and Xia Zhang. Road traffic congestion detecting by vanets. In Proceedings of the 2nd International Conference on Electrical and Electronic Engineering (EEE 2019), pages 242–248. Atlantis Press.
4. Ananya Paul and Sulata Mitra. Management of Traffic Signals using Deep Reinforcement Learning in Bidirectional Recurrent Neural Net- work in ITS. In 5th International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence (ISMSI), pages 436–440. ACM, 2021.
5. Ananya Paul and Sulata Mitra. Real-Time Routing for ITS Enabled Fog Oriented VANET. In 17th India Council International Conference (INDICON), pages 1–7. IEEE, 2020.
6. Nazmus S Nafi and Jamil Y Khan. A vanet based intelligent road traffic signalling system. In Australasian Telecommunication Networks and Applications Conference (ATNAC) 2012, pages 1–6. IEEE, 2012.
7. Anurag Kanungo, Ayush Sharma, and Chetan Singla. Smart traffic lights switching and traffic density calculation using video processing. In Engineering and computational sciences (RAECS), 2014 recent advances in, pages 1–6. IEEE, 2014.
8. Chakkaphong Suthaputchakun and Zhili Sun. A Novel Traffic Light Scheduling Based on TLVC and Vehicles' Priority for Reducing Fuel Consumption and CO2 Emission. IEEE Systems Journal, 12(2):1230– 1238, 2015.
9. Leena A. Abdalla, Reem M. Taha, Nejod S. Mohammed, and Niemah I. Osman. Reducing delay and co2 emissions in multi-lane intersections using vehicular ad-hoc networks. In International Conference on Computer and Information Sciences (ICCIS), pages 1–6, 2019.
10. Abhishek Srivastava, Gaurav Kapoor, and Aman Gupta. Solving traffic congestion—An application of VANET. In International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH), pages 323–326. IEEE, 2016.
11. Badreddine Cherkaoui, Abderrahim Beni-Hssane, Mohamed El Fissaoui, and Mohammed Erritali. Road traffic congestion detection in VANET networks. Procedia Computer Science, 151:1158–1163, 2019.
12. Jihene Rezgui, Mamadou Barri, and Reiner Gayta. Smart traffic light scheduling algorithms. In International Conference on Smart Applications, Communications and Networking (SmartNets), pages 1– 7. IEEE, 2019.
13. Majed Al-qutwani and Xingwei Wang. Smart traffic lights over vehicular named data networking. Information, 10(3):83, 2019.
14. Ananya Paul and Sulata Mitra. Dynamic traffic light control mechanism in vanet. In 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pages 436–440. IEEE, 2018.
15. Qin Zhu, Chao Peng, Jingmin Shi, Pengfei Duan, Yu Bao, and Mengjun Xie. Cooperative traffic light control based on semi-real-time processing. Journal of Automation and Control Engineering Vol, 4(1), 2016.
16. Jingmin Shi, Chao Peng, Qin Zhu, Pengfei Duan, Yu Bao, and Mengjun Xie. There is a will, there is a way: A new mechanism for traffic control based on vtl and vanet. In High Assurance Systems Engineering (HASE), 2015 IEEE 16th International Symposium on, pages 240–246. IEEE, 2015.
17. Trung-Thanh Ngo, Thien Huynh-The, and Dong-Seong Kim. A novel vanets-based traffic light scheduling scheme for greener planet and safer road intersections. IEEE Access, 7:22175–22185, 2019.
18. Yit Kwong Chin, Wei Yeang Kow, Wei Leong Khong, Min Keng Tan, and Kenneth Tze Kin Teo. Q-learning traffic signal optimization within multiple intersections traffic network. In Computer Modeling and Simulation (EMS), 2012 Sixth UKSim/AMSS European Symposium on, pages 343–348. IEEE, 2012.
19. Wade Genders and Saiedeh Razavi. Using a deep reinforcement learning agent for traffic signal control. arXiv preprint arXiv:1611.01142, 2016.
20. Maythem Kamal Abbas, Mohammad Noh Karsiti, Madzlan Napiah, and Brahim B Samir. Traffic light control via vanet system architecture. In Wireless Technology and Applications (ISWTA), 2011 IEEE Symposium on, pages 174–179. IEEE, 2011.
21. Pallavi Choudekar, Sayanti Banerjee, and MK Muju. Implementation of image processing in real time traffic light control. In Electronics Computer Technology (ICECT), 2011 3rd International Conference on, volume 2, pages 94–98. IEEE, 2011.
22. Laura Garelli, Claudio Casetti, Carla-Fabiana Chiasserini, and Marco Fiore. Mobsampling: V2v communications for traffic density estimation. In Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd, pages 1–5. IEEE, 2011.
23. Namrata S Jadhao and Ashish S Jadhao. Traffic signal control using reinforcement learning. In Communication Systems and Network Tech- nologies (CSNT), 2014 Fourth International Conference on, pages 1130– 1135. IEEE, 2014.
24. Yanmin Zhu, Xuemei Liu, Minglu Li, and Qian Zhang. Pova: Traffic light sensing with probe vehicles. IEEE Transactions on Parallel and Distributed Systems, 24(7):1390–1400, 2013.
25. Q-learning - wikipedia. https://en.wikipedia.org/wiki/Q-learning.
26. Lecture 4.pdf. https://nptel.ac.in/courses/110106046/Module
27. Hossein Rashid, Mohammad Javad Fazel Ashrafi, Mohsen Azizi, and Mohammad Reza Heydarinezhad. Intelligent traffic light control based on clustering using vehicular ad-hoc networks. In Information and Knowledge Technology (IKT), 2015 7th Conference on, pages 1–6. IEEE, 2015.
28. Nitin Maslekar, Mounir Boussedjra, Joseph Mouzna, and Houda Labiod. Vanet based adaptive traffic signal control. In Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd, pages 1–5. IEEE, 2011.
29. Yujie Dai, Jinzong Hu, Dongbin Zhao, and Fenghua Zhu. Neural network based online traffic signal controller design with reinforcement training. In Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on, pages 1045–1050. IEEE, 2011.
30. Dai Yujie and Dongbin Zhao. A traffic signal control algorithm for isolated intersections based on adaptive dynamic programming. In Net- working, Sensing and Control (ICNSC), 2010 International Conference on, pages 255–260. IEEE, 2010.

AUTHORS PROFILE

Ananya Paul, received B.Tech degree in Computer Science and Technology from St. Thomas College of Engineering and Technology in 2015 and M.Tech degree in Mobile Computing and Communications From Jadavpur University in 2017. She is currently pursuing her research from Indian Institute of En- gineering, Science and Technology, Shibpur. Her current research interest is VANET, ITS, IoT, Fog Computing.

Kiton Ghosh, received B.Tech degree in Information Technology from Kalyani Govt. Engg. College in 2017. He is currently pursuing his M.Tech degree in Computer Science and Technology from Indian Institute of Engineering, Science and Technology, Shibpur. His research interests include VANET, ITS, IoT, Fog Computing.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

Sulata Mitra, received B.E. degree from Bengal Engineering College (India) in 1986 and PhD de- gree in Mobile Computing from Bengal Engineering College (D.U.), Shibpur (India) in 2005. She joined the Indian Institute of Technology, Kharagpur in1989 as Senior Research Assistant and moved to the Regional Institute of Technology, Jamshedpur (India) in 1991 as Lecturer. Dr. Mitra has published 50 technical papers in journal and international con- ference proceedings. Her current research interestis QoS issues in 3G/4G cellular network, VANET,
Multihomed mobile network. She is currently with the Computer Science and Technology Department of Indian Institute of Engineering, Science and Technology, Shibpur as Professor.

Retrieval Number: 100.1/ijrte.A59660510121
DOI: 10.35940/ijrte.A5966.0510121

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication