ATSOT: Adaptive Traffic Signal Using mOTes

Punam Bedi, Vinita Jindal, Heena Dhankani, and Ruchika Garg

Department of Computer Science, University of Delhi, India
punambedi@ieee.org, vjindal@keshav.du.ac.in,
{heena.cs.du.2013,ruchika.cs.du.2013}@gmail.com

Abstract. This paper presents design and development of Adaptive Traffic Signal using mOTes (ATSOT) system for crossroads to reduce the average waiting time in order to help the commuter drive smoother and faster. Motes are used in the proposed system to collect and store the data. This paper proposes an adaptive algorithm to select green light timings for crossroads in real time environment using clustering algorithm for VANETs. Clustering algorithms are used in VANETs to reduce message transfer, increase the connectivity and provide secure communication among vehicles. Direction and position of vehicles is used in literature for clustering. In this paper, difference in the speed of vehicles is also considered along with direction, node degree, and position to create reasonably stable clusters. A mechanism to check the suitability of cluster initiator is also proposed in the paper. The proposed ATSOT system can be used for hassle free movement of vehicles across the crossroads. Prototype of the system has been designed and developed using open source software tools: MOVE for the Mobility model generation, SUMO for traffic simulation, TraCI for traffic control Interface and Python for client scripting to initiate and control the simulation. Results obtained by simulating ATSOT approach are compared with both OAF algorithm for adaptive traffic signal control and pre-timed approach to show the efficiency in terms of reduced waiting timing at the crossroads. Results are also compared with pre-timed method for single lane and multi-lane environment using Webster's delay function.

Keywords: Motes, VANETs, clustering algorithm, adaptive control, threshold, traffic signals.

1 Introduction

Traffic problem is an extremely significant issue affecting our daily life. One spends a lot of time waiting on the crossroads resulting in significant delay to reach the destination with more fuel consumption. Traffic signals are used to manage traffic on the crossroads. A traffic signal can be pre-timed, semi-actuated or fully-actuated. Pre-timed signal consists of a programmed sequence of various signal phases which is repeated after completion of a cycle. A phase is defined as an assignment of non-conflicting and simultaneous movement of traffic into separate groups to allow collision free vehicle movement [1]. Congestion on crossroads is major disadvantage

W. Chu et al. (Eds.): DNIS 2015, LNCS 8999, pp. 152–171, 2015.

of pre-timed signals when there is no traffic on the road with green light assignment and too much traffic on the road towards red signal. To solve this problem, actuated signals came into the scenario. Actuated signals are capable of altering the phase sequence with varying phase time for a signal depending on vehicle density on the road, which enable them to reduce traffic delays extensively. Actuated signals are further divided into two sub categories: semi-actuated and fully actuated.

Semi-actuated signals use the detectors present only on the major roads, whereas in case of fully actuated signals, all roads (major or minor) have detectors installed on them [1], [2]. The signal is set to provide the green light time as per traffic requirement. The effectiveness of traffic flow across a crossroad varies according to the phases, phase sequences and the timing of the traffic signals installed. Thus, in order to ensure the safety and normal traffic flow at the crossroads, design of adaptive traffic signals becomes essential [3]. The adaptive traffic signals are mainly used for adapting the timings of lights by predicting the traffic volume for smooth traffic flow without congestion.

Most of the cities in the developing countries have pre-time signals, whereas actuated signals are installed on the crossroads in most of the developed countries. There has been significant works done in the area of adaptive traffic signals for congestion free movement of traffic across the crossroads in the literature. There exist a variety of traffic light control systems implemented worldwide like Split Cycle and Offset Optimization Technique (SCOOT) [4], [5], Sydney Coordinated Adaptive Traffic System (SCATS) [6], [7], Real time Hierarchical Optimized Distributed Efficient System (RHODES) [8], Optimization Policies for Adaptive Control (OPAC) [9], Real-Time Traffic Adaptive Control Logic (RTACL) [10] etc. These systems are mainly based on the inductor loop detectors and video sensors installed on the intersections. As per our knowledge, none of these techniques uses clustering mechanism to form stable clusters to adjust the green time for traffic signals.

In this paper, motes are used to sense the traffic flow on the roads which is further used for clustering. Mote is a wireless sensor node, also known as smart dust, gathers sensory information, processes it and then communicates result to other nodes in the network [11]. Mote is a node but vice versa is not true always [12]. VANET is a special type of Mobile Ad hoc Network (MANET) having its root in traffic engineering that permits interaction among the vehicles; vehicle & infrastructure and the infrastructure itself [13], [14]. VANETs are supposed to integrate wireless technology as a type of Wi-Fi [15], Dedicated Short-Range Communication (DSRC) [16], satellite, cellular and WiMAX.

The communication range allowed for VANETs is restricted to 100 m to 300 m. The participating vehicles are turned into routers or wireless nodes for connection and data collection in the allowed range of the network. A new vehicle can join the network when it comes in the range and any vehicle can be dropped out of the network when it goes out of range. VANET can be considered as the backbone for the Intelligent Transportation System (ITS).

VANET is characterised by moving vehicles considered as nodes and fixed roads considered as edges between the nodes to form a graph like structure. There also exist fixed road side units (RSUs) alongside of road to enable the communication among the vehicles. “Fixed infrastructure belongs to the government or private network operators or service providers” [17]. Some of the benefits provided by VANETs are accident prevention, safer roads, congestion reduction and less waiting time. In VANETs messages are broadcasted through a wireless medium and hence communication is at a greater risk. Moreover due to dynamic nature of the vehicles, stale entries as well as congestion exist in the network.

Clustering algorithms ensure security, increase connectivity and reduce overall message transfer in VANETs [18], [19]. The main concern is on finding the stable clusters of vehicles in vehicular network and the job of clustering algorithms is to find not only the minimum number of clusters, but also preserve the existing configuration of the clusters with the least overhead [20]. In VANETs, cluster formation is highly challenging due to highly mobile nodes and unavailability of global topology [21]. In our proposed approach data is collected through motes and is clustered with the clustering approach discussed by Rawashdeh and Mahmud [22]. During the cluster formation process the relative speed of the vehicles along with the position and direction is considered. The clustering based on the above said parameters is found to be stable and efficient.

The clustering algorithm divides the network into clusters wherein highly mobile vehicles are positioned in one cluster and moderately low mobility vehicles are positioned in the alternative group. For the formation of stable clusters, we have devised a mechanism to test the suitability of the cluster initiator. It is required to minimize the cluster initiator transfer. Another consideration for clustering within the VANETs is that the algorithm has to be very fast to minimize time loss. After cluster formation, density is calculated and the information is fed into the proposed adaptive algorithm running on the RSUs. With the received information, signal time is decided according to the algorithm based on threshold. The main idea is to reduce the overall waiting time at the crossroads and ensure speedy movement of the vehicles in order to reduce the fuel consumption.

Rest of the paper is organized as follows. Section 2 reviews the work done in literature in the area of Intelligent Transportation System along with the clustering algorithms used in VANETs. The proposed system ATSOT is presented in section 3. Experimental setup and results are shown in section 4 followed by the conclusion in section 5.

2 Literature Review

Intelligent Transportation Systems (ITS) can be used as a proficient approach to improve the operation of VANETs. Objectives of ITS include comfortable driving, road safety and dissemination of updated road information to the commuters [23].

Adaptive traffic signals have been widely studied in the literature. Examples include the well-known SCOOT, SCATS, RHODES, OPAC, RTACL etc. SCOOT is a centralized traffic responsive system that coordinates the traffic light in a fixed green light sequence as an automatic respond to traffic flow fluctuation in urban areas [4], [5]. SCATS is a system which programs the traffic lights by providing the intelligent pre-defined traffic plans to offer vehicle delay reduction [6], [7]. It relies on loop detectors placed on the lane pavement before the intersection. RHODES uses three level hierarchies for the management of traffic signals [8]. It is designed for under saturated conditions. OPAC is designed for oversaturated conditions [9]. RTACL is designed for networks of streets [10]. RTACL did not meet expected performance measures. OPAC, increased delay and travel time in some instances, and RHODES reduced cycle lengths, but did not show any significant difference in arterial travel times.

Junping Zhang et al. [24] presented a study of multifunctional data driven intelligent transportation system, which can accumulate a massive sum of data collected from diverse resources: Vision-Driven ITS (input data collected through video sensors and used recognition including pedestrian and vehicle detection); Multisource-Driven ITS (e.g. laser radar, inductive-loop detectors and GPS); Learning-Driven ITS (effective prediction of the accidents occurrence to boost the safety of pedestrians by decreasing the effect of vehicle collision);and Visualization-Driven ITS (to assist the decision makers swiftly recognize anomalous traffic patterns and consequently acquire obligatory counteractive actions). In case of complex scenario i.e. with large number of vehicles, there exist some obstacles for object reorganization.

Car-to-car communication based adaptive traffic signal control system is discussed in [25]. The approach reduces the queue length to reduce the waiting time at the crossroads for the vehicles. Clustering is used for the density calculation of approaching vehicles, which is used by the traffic signal controls for setting the cycle timing in this system. DBCV algorithm is used to gather the desired density information by combining cluster and opportunistic dissemination technique. The direction is computed within the vehicle by engaging digital maps and Global Positioning Systems (GPS) in a geographical region and helps in the formation of clusters.

An adaptive traffic light system is designed by Gradinescu et al [26] in which each vehicle is equipped with a short range communication device. The system is centered on a wireless controller node that is positioned on the intersection to determine the optimal values for the signals. Traffic signals can respond to the communication between vehicles to estimate the density of vehicles around it and consequently adjust the signal timing. In [25] and [26] adaptive traffic light systems based on wireless connectivity between fixed controller nodes and vehicles at crossroads are described. These systems are planned and developed with the objective of improving traffic smoothness, reducing the waiting time at crossroads and helping in the prevention of accidents.

Ganesh S. Khekare et al [27] are using the traffic information to avoid accidents. Vehicles themselves are providing this information without the need of extra infra-structure. The authors have developed an adaptive traffic control system similar to the proposal given by [25], [26] in their work. The system is implementing the AODV routing algorithm without considering any clustering method.

In [22], authors have described a speed-overlapped clustering method in highways scenario. In their work, stable and unstable clustering neighbours are defined according to the relative position, movement direction and speed of the vehicles. Clusters are created only among the stable neighbours and clustering can be started only from the slowest or fastest vehicle. Concept of location services is also being used in the earlier work by various researchers in [18], [19], [28], [20], [29]. A problem is also foreseen in method described in [22] if the speed of some vehicle deviates too much from the speed of other vehicles in the cluster, then cluster formation process becomes unstable. But as compared to other methods, this method creates comparatively stable clusters and hence we have used this clustering approach in our work to propose an adaptive algorithm.

Webster’s delay function [3] and HCM 2000 [27] are the main delay functions that exist in the literature as per our knowledge. HCM 2000 method is based on the concept of lane groups. Saturation flow and delay are calculated for each lane group separately. In Webster’s method, flow is calculated for the total approach width assuming the arrival rate to be random. We have considered the traffic of mixed type with poor lane discipline as present in many developing countries. Hence, Webster's method is used in our work.

Webster’s delay model is the standard delay model with the assumption of vehicles arrival at a uniform pace [2], [30], [31], [32], [1], [33]. The expression given by Webster for delay per cycle, d, is as follows:

$$d = \frac{\frac{C}{2}\left[1-\frac{g}{C}\right]^2}{1-\frac{v}{S}} \tag{1}$$

Where g is the effective green time, C is the cycle length, v is the critical flow for the phase and S is the saturation flow. This eq. (1) computes the optimal cycle length that minimizes the average vehicle delay. Effective green time, g, is calculated by the eq. (2) as follows:

$$g = C - t_l \tag{2}$$

Where C is the cycle length and t_l is the lost time. The study done in [34] uses a new job-scheduling based online OAF (oldest arrival first) algorithm in traffic signal control. The results obtained show that the OAF algorithm reduces the delays experienced by the vehicles, as compared with Webster’s method and the pre-timed signal control approach.

Literature work focuses on the clustering mechanisms that are mainly based either on direction, speed, position or their combinations. The presented work considers the parameters degree of the node and speed difference along with direction and position while forming the clusters. We also use motes to sense and collect the data that will be fed to the clustering algorithm resulting in the formation of the stable clusters.

Earlier researchers have focused only on the assignment of green time for signals in various phases. Here in this work, optimal green light assignment is done as per traffic on road either by incrementing or decrementing green light timings.

3 Proposed Adaptive Traffic Signal Using mOTes System

The proposed Adaptive Traffic Signal using mOTes (ATSOT) system is based on motes, clustering algorithm and proposed adaptive algorithm. Motes help in collecting data and feed it to the clustering algorithm which forms stable clusters. These stable clusters are used to calculate average queue length which is used by the adaptive algorithm to set the optimal green light timing as per traffic volume on roads for the signal at crossroad.

3.1 Framework of ATSOT

Framework for the proposed ATSOT system is shown in Fig. 1. In ATSOT system, traffic data is sensed and collected through the motes. This data is stored in the databases located locally at RSUs and globally at the traffic centers, and consist of the information about the traffic (e.g. position, density etc. for the moving vehicles). The data collected through motes is fed to the clustering algorithm which runs locally on RSUs and updates the results to traffic centers. For stability of clusters, a formula to check the suitability for becoming cluster initiator is also proposed. From the obtained clusters, average queue length is calculated on that crossroad. Finally, an adaptive algorithm for setting the green light timing is run on the RSUs to set the green light timing for signals at crossroads. RSU simultaneously updates the traffic centers with the results obtained. The algorithm considers minimum threshold, Th_{min} and maximum threshold, Th_{max} during the computation of green light duration for providing congestion free path with least waiting time. Detailed working of the ATSOT system is shown in Fig. 2.

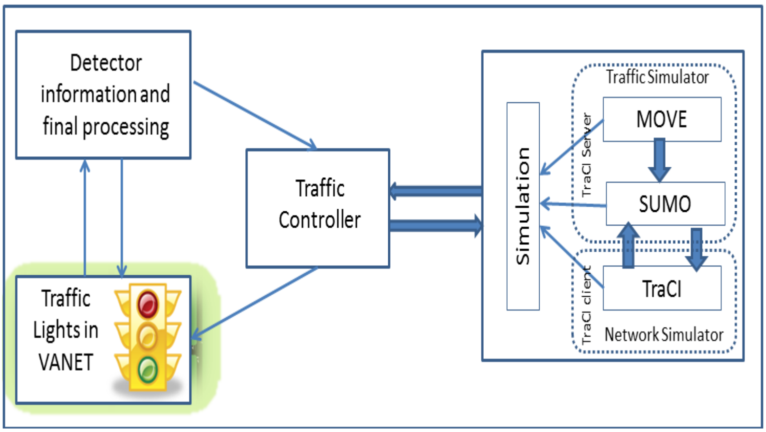

Fig. 1. Framework of ATSOT

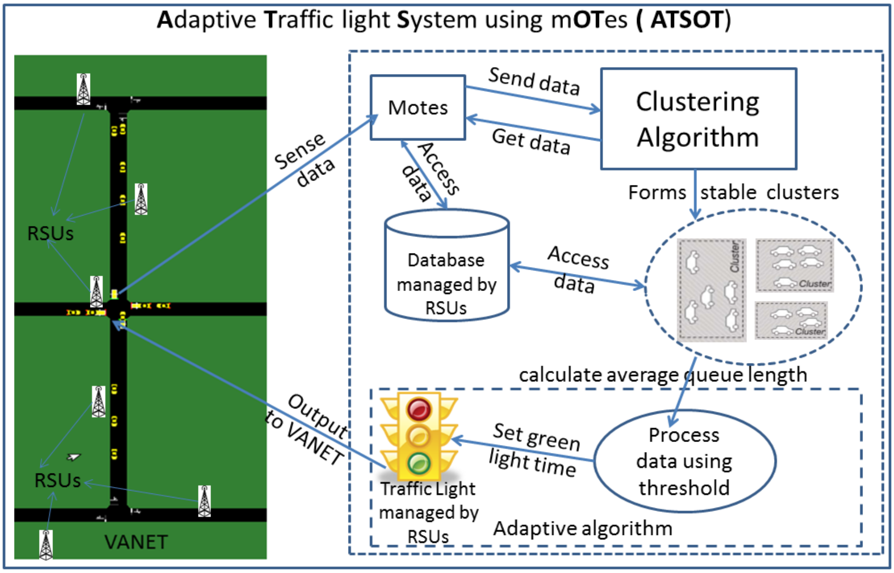

Fig. 2. Detailed working of ATSOT

We assume that each road has a sensor detector installed on the road to calculate the total number of vehicles passing through it. These sensors will transmit the information to the ATSOT system and calculate the cycle timing for next phase.

In Fig. 2, a typical VANET is shown. Motes are installed near traffic lights on the roads. These motes sense the vehicle data and store it in the databases placed at the traffic centers through nearest RSUs. Motes also disseminate the data to the clustering algorithm resulting in formation of the stable clusters of vehicles on the road. These clusters help in computing the average queue length. Clustering algorithm also checks the suitability of the cluster initiator for the formation of stable clusters through the proposed formula. The average queue length is then computed at the nearest RSU and used to select the green time for the traffic lights at crossroads based on threshold.

3.2 Clustering Process in ATSOT System

Clustering algorithm is used in ATSOT system to reduce the number of messages transfer, increase the connectivity in the network and to simplify routing process. It also offers secure communication among vehicles. During the clustering process, nodes are divided into clusters, each with a cluster initiator (CI) node that is accountable for all management and coordination tasks of its cluster. In the presented work, main concern is on finding the stable clusters of vehicles in vehicular network and the job of clustering algorithms is to find not only the minimum number of clusters, but also preserve the existing configuration of the clusters with the least overhead.

To implement the algorithm, individual vehicle is presumed to retain the IDs of the stable neighbours that are updated at regular intervals. To form clusters, neighbouring vehicles cooperate with each other. Vehicles build their neighbourhood relationship using the GPS data. Clusters are formed for the vehicles traveling in the same direction based on their position, speed and node degree. However, the speed among the vehicles varies and this variation might be very high; thus, all neighbouring vehicles are not included in one cluster. Underlying assumption is the existence of a vehicle with the slowest speed among its stable neighbours for initiation of the cluster formation process. The vehicle with slowest speed is called cluster initiating (CI) vehicle.

Since in our work, slower and faster vehicles will be in different clusters, we can start the cluster formation process either from the slowest or fastest vehicle in each direction. The neighbouring vehicles whose relative speed, with respect to the slowest (or fastest) vehicle, is greater than (or less than) the threshold, V_{th}, will not be grouped in the same cluster. In our work, we are choosing the slowest vehicle, V_s to initiate the clustering process by sending a cluster formation request, and then all of its neighbours with relatively similar speed will be in the first cluster. The remaining vehicles will then go through the same cluster formation process to create other clusters. Speed of the vehicle can be extracted from the GPS data. The clustering process starts with the explanation of cluster initiator selection parameters followed by the cluster formation algorithm and cluster maintenance procedure as follows:

Cluster Initiator Selection Parameters

The mobility information (speed, position, degree, and direction) of the nodes is extracted through GPS data. Priority of a node to become a cluster initiator is calculated by its suitability value, u, which is computed based on the mobility information of its neighbourhood.

Thus, u = f (degree, position, speed) is a function defined according to the following criteria:

- The suitability value of the vehicle is calculated by considering the mobility information of vehicle within radius, r and having relatively similar speed only.
- Nodes having higher number of neighbours in their neighbourhood, maintaining closer distances to their neighbours, and having closer speed to the average speed of their stable neighbours should have higher suitability value, thus they are more qualified to be elected as cluster-heads.

Cluster Formation Algorithm

In order to execute the algorithm, each vehicle is assumed to access the GPS data continuously. There exists a vehicle with slowest speed. The algorithm is divided into three phases specified by each of the three algorithms given below.

Algorithm 1 specifies the clustering initiation process followed by the algorithm 2 that explains the process of CI determination with competition among vehicles themselves. Final cluster formation process is described in algorithm 3.

Input : CI ← vehicle ID with slowest speed and stable neighbours.
Output: stable and minimum number of clusters

Algorithm 1: Initiating clustering process

```
if (there is no vehicle with speed less than that of the current vehicle || vehicle with speed
 less than that of the current vehicle ϵ other clusters) then
        temporary CI ← Current vehicle
        send Initiate_Cluster (CI)
end if
```

Algorithm 2: CI competition and determination

```
if there are vehicles with speed greater than that of the current vehicle then
        On Receiving Initiate_Cluster (CI)
        CI ← current vehicle
                check the suitability of the CI w.r.t its neighbouring vehicles having speed
        greater than that of the CI
        calculate the waiting time before broadcasting admissibility of the current vehicle to
        become a CI
        while waiting time > 0 do
            if Form_Cluster (CID) is received then
                if speed of the CI is greater than that CID then
                        Quit_Competition()
                        Process Form_Cluster (CID)
                end if
            else
                Decrement waiting time
            end if
        end while
        current vehicle declare itself as Custer initiator with its cluster id as  CID
        Send Form_Cluster (CID)
end if
```

Algorithm 3: Finalizing cluster formation process

```
if there exist some vehicle with speed difference of V_th then
        On Receiving Form_Cluster (CID)
        if that vehicle exists within the radius, r then
                that vehicle become a Cluster Member
                cluster id for the vehicle is ID
        else
                Cluster id will be the default i.e. the vehicle's id
                Reconstruct the set of the vehicles with speed greater than that of current
                vehicle
        end if
end if
```

Cluster Maintenance

VANETS are highly dynamic in nature. Thus, vehicles keep joining and leaving clusters frequently, causing extra overhead on the maintenance. The maintenance procedure consists of mainly three events that are described as follows:

- **Joining a Cluster:** when a non-clustered vehicle comes within the radius, r of a CI, both the CI and vehicle check whether their relative speeds are within the threshold $\pm V_{th}$ or not. If the speed difference is within $\pm V_{th}$, then that vehicle is added to the cluster members list of that CI. If there is more than one CI in the neighbourhood, then vehicle performs the suitability test to join the cluster.
- **Cluster Merging:** when two CI come within each other's radius r, and their relative speeds are within the predefined threshold $\pm V_{th}$, the cluster merging process takes place. The CI with less number of members gives up its CI role and becomes a cluster-member in the new cluster. The remaining cluster members may join that neighbouring cluster if they exist within the radius of another CI and the speed is also within the threshold. Finally, vehicles that cannot merge with any cluster nor can join a nearby cluster, start clustering process to form a new cluster.
- **Leaving a Cluster:** when a vehicle moves out of the range of cluster radius, r, that vehicle is removed from the cluster members list maintained by the CI. The vehicle changes its state to a standalone if there is no nearby cluster to join or there is no other nearby standalone vehicle to form a new cluster.

3.3 Adaptive Algorithm for Controlling Traffic Lights

After cluster formation, density is calculated and the information is fed into the intelligent traffic signals by the motes. With the received information, signal will be decided according to the proposed adaptive algorithm based on threshold. The main idea is to reduce the overall waiting time at the crossroads and ensure speedy movement of the traffic. The adaptive algorithm is given below:

Adaptive Traffic Control Algorithm

In order to execute the algorithm, clustering process for the vehicles is assumed to be completed. The average queue length is calculated from the clustering information for each direction. Further from the statistical experimentation, the minimum and maximum values for threshold are being selected as Th_{min} and Th_{max} respectively. Output of the algorithm is to find the optimal green light (GL) timing for the crossroad. Default GL time is assumed as 60 seconds in this algorithm which can be chosen as desired. In the proposed algorithm, we are taking T_{min} and T_{max} as the minimum and maximum time needed by the Th_{min} and Th_{max} number of vehicles to pass through the crossroad in one phase respectively. For the green light reduction, we are taking GL equal to T_{min} to pass all Th_{min} vehicles with loss time. Further, for green light extension, we have used an equation to allow maximum number of vehicles to pass in one phase with loss time given as follows:

$$GL = T_{max} + \frac{(n_i - Th_{max})T_{max}}{Th_{max}} \tag{3}$$

Where GL is the optimal green time, Th_{max} is the maximum threshold, T_{max} is the time taken by Th_{max} vehicles to pass through crossroad in one phase and n_i is the average queue length of the i^{th} phase. The proposed algorithm is based on the following conditions:

- If there are no vehicles in one of the phases that has been allotted the green signal but there are waiting vehicles in the other phase according to information given by detectors, green time extension is allotted to the other phase.
- If there are number of vehicles less than that of the minimum threshold value in one of the phase that has been allotted green light the green light time is set to the minimum time required by signal for passing the crossroad.
- If vehicles are waiting for the green signal at the crossroad on all the phases, then to reduce congestion check if the queued vehicles is greater than some threshold on a phase , green signal is allotted to that phase.
- In the above case when there is queuing on all the phases but queuing is less than threshold then also consider the time that a vehicle has been waiting on the crossroad. If this waiting time is greater than some threshold time limit then change to green the signal to give way to those vehicles.

The parameters used by the algorithm are defined as follows:

L_i : i^{th} phase traffic lights state. In the work, we take 0 for red traffic, 1 for green traffic and 2 for yellow traffic.

P_i : i^{th} phase where i=1, 2, 3, 4.

WT_i : waiting time of vehicle at i^{th} phase.

time_limit: maximum time that a vehicle can wait on the crossroad. This parameter is used to reduce infinite waiting of a vehicle.

Th_{max}, threshold: maximum number of vehicles that are queued on each phase.

Th_{min}, threshold: minimum number of vehicles that are queued on each phase.

T_{min} : maximum time taken by the Th_{min} vehicles to pass through the crossroad. This parameter can be determined through experiments.

T_{max} : maximum time taken by the Th_{min} vehicles to pass through the crossroad. This parameter can be determined through experiments

Steps of the algorithms are explained as follows:

Algorithm 4: Adaptive traffic control (L, P_i, WT_i)

```
While (true)
    for i = 1, 2, 3, 4
        n_i=Average Queue Length at P_i
        if n_i==0
         for all values of j such that i ≠ j
           n_j = Average Queue Length at P_j
           WT_i=waiting time of vehicle at phase P_i
                if  n_j == 0
                   go to Label
              else if  WT_i >time_limit || Th_min <= n_j  <= Th_max)
                         L_i=1 with same GL
                         go to Label
              else if  n_j <= Th_min
                       L_i=1 with GL =   T_max
                         go to Label
                    else if n_j   >= Th_max
                        L_i=1 with GL = T_max + ((n_j - Th_max)* T_max ) / Th_max
          Label:   continue the for loop
                   endif
       continue the for loop
continue the while loop
```

4 Experimental Setup and Results

The proposed system is simulated using traffic simulators: MObility model generator for VEhicular networks (MOVE) [35], Simulation of Urban MObility (SUMO) [36], [37], [38] and network simulator: Traffic Control Interface (TraCI) [39], [40]. MOVE is a tool that is adopted on the top of SUMO which is open source micro-traffic simulator. MOVE provides a GUI that allows the user to quickly generate real-world simulation scenarios without a simulation scripts. The output of MOVE is mobility trace file that contains real-world vehicle information that can be used by SUMO [41].

SUMO and MOVE are used to model the moving behavior of vehicles and produce traffic information. These two together are traffic simulators and work as server for the system. MOVE is used to simulate a realistic mobility model for VANET in the presented work and its output is given to SUMO which acts as TraCI server. TraCI is the traffic control interface that works as a client for the system and can be programmed using Python scripting language to initiate the communication for further processing in the proposed system. In the simulation, there is interaction with traffic control, which takes the detector information from motes and uses the proposed adaptive algorithm that makes use of threshold to decide the green light time for traffic lights in real time.

The objective of adaptive traffic signals is to minimize the average queue length for the vehicles at the crossroads. In our work we are using motes to sense and collect the data that is fed to the clustering algorithm. For the purpose of analysis we initially design the system to handle single lane traffic, and further extended the simulation for multi-lane traffic. The mobility of vehicles was defined by SUMO that follows the designed road. A random uniform distribution of the speed was specified amongst the vehicles between the range 3 m/s to 9 m/s for the vehicle mobility. The value of distance for the cluster formation is derived from an experimental analysis. The key parameters used for the simulation are summarized in the table 1 below:

To test the accuracy of the system, simulations were run 10 times and the average value was taken to obtain the results. ATSOT is compared with a pre-timed signal system whose cycle time is set by the simulator to be 60 seconds for a phase. This cycle time comprises of the green time and the inter-green time. Further, the measures of validation also included comparison with already existing clustering algorithm. Under the similar simulation conditions we find that ATSOT considerably reduces average queue length experienced by the cars at crossroad. We start the simulation with single lane traffic model with one crossroad as shown below in Fig. 3 and extend the work with multi-lane traffic model with several crossroads as depicted below in Fig. 4.

Table 1. Simulation Setup

Simulation Scenario	**Values**
Simulation Time	100-4000 s
Vehicle Speed	3 – 9 m/s
Transmission Range	10 m
Number of Simulations for each case	10
Driver Reaction Time	1.5 s
Mobility Model	Car Following
Cycle Time for Pre-timed system	60 s
Length of Car	5 – 8 m

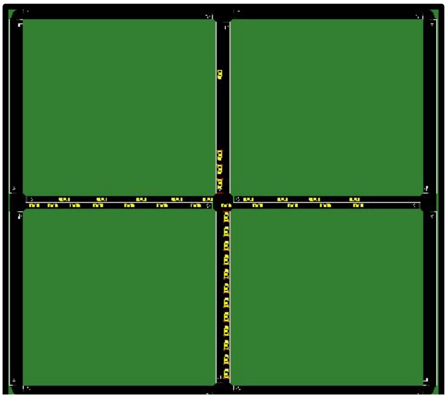

Fig. 3. Screen Shot for single lane traffic

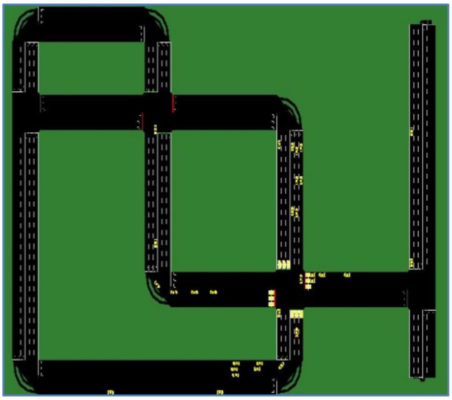

Fig. 4. Screen Shot for multi- lane traffic

In single lane traffic model, we consider the straight directions only for moving vehicles. Then extend the simulation with multi-lanes traffic model for vehicle movements in left, straight and right directions. Next, we have used two phases for single lane traffic model as phase 1 and phase 2; and four phases as phase 1, phase 2, phase 3 and phase 4 shown in Fig. 5 for multi-lane traffic models for congestion free movement across the crossroads in the simulation.

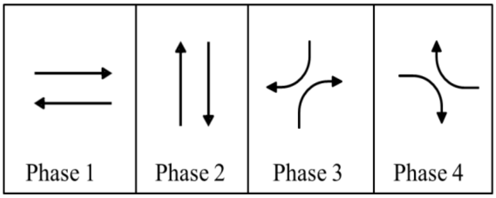

Fig. 5. Different phases used in ATSOT

Fig. 6 depicts the enlarged view of one junction with traffic signals. In the underlying system we developed a simulation in python for pre-timed traffic light control with and without Webster's method under both single lane as well as multi-lane approach. Further we simulated ATSOT system, in which data is collected through motes and fed to the clustering algorithm to form stable clusters. These clusters are used by the adaptive traffic light control algorithm to set the optimal green time for the traffic signals. We also coded the existing OAF algorithm, pre-timed model with and without Webster's delay and finally compare the results in both single lane as well as multi-lane approaches.

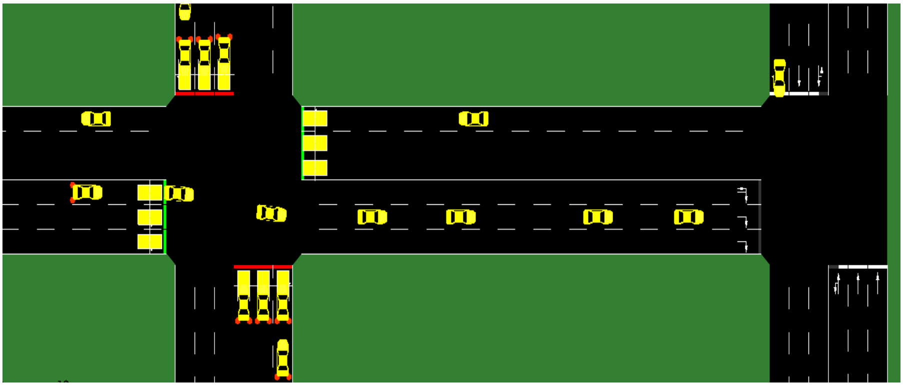

Fig. 6. Enlarged view of a junction with traffic signal in ATSOT

Results are tabulated in Table 2 and Table 3 shown below along with their graphical representation shown in Fig. 7 and Fig. 8. Initially ATSOT system is simulated for single lane approach, and then existing OAF algorithm is also implemented under similar scenario. Further, pre-timed system is simulated under same conditions with and without Webster's model. In the simulation, we gradually increase the number of vehicles and record the average queue length for each case. The results are recorded in Table 2 and their graphical representation is being shown in Fig. 7.

Experimental results show that ATSOT approach implemented in single-lane traffic model, average queue length is less as compared with both OAF and pre -timed approach with and without Webster's delay with increase in vehicles on road. ATSOT approach outperforms all the other three methods. The OAF algorithm performs almost similar as that of ATSOT for less number of vehicles. However, as the number of vehicles increases ATSOT performs better and reduces the average queue length by 1 - 11% as compared with OAF. Further, ATSOT reduced the average queue length by 14 - 83% and 17 - 92% respectively in pre-timed approach with and without Webster's approach.

Table 2. Results for Average Queue Length in ATSOT, OAF algorithm and pre-timed with or without Webster's method in Single lane

Total Vehicles in Simulation	Average Queue Length (ATSOT)	Average Queue Length (OAF algorithm)	Average Queue Length (pre-timed)	Average Queue Length (pre-timed with Webster's)
200	2	2	25	12
400	23	21	50	32
600	74	75	75	75
800	85	95	102	99
1000	139	155	177	169

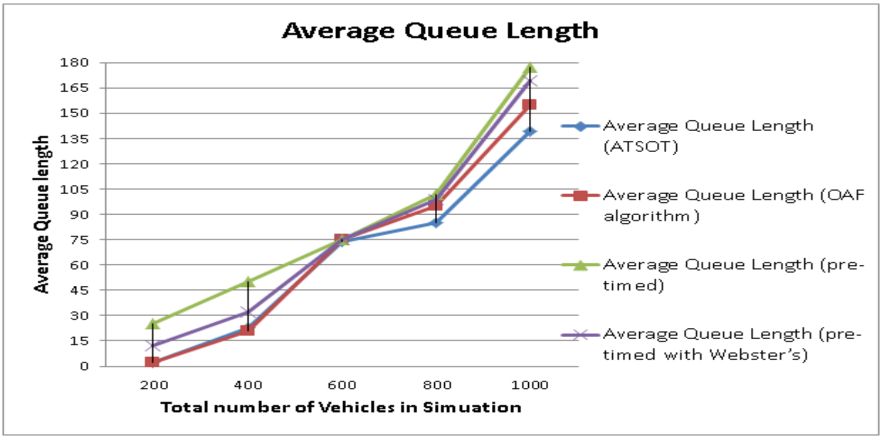

Fig. 7. Graph for Average Queue Length in ATSOT, OAF algorithm and pre-timed with or without Webster's method in Single lane

The simulation was further extended to handle heavy traffic on multi-lane roads. The ATSOT system is implemented for multi-lane approach. The performance of the system is compared with that of existing OAF algorithm under similar conditions. Further, pre-timed system is simulated under same environment with and without Webster's model. In the simulation, we gradually increase the number of vehicles and record the average queue length for each case. The results are recorded in Table 3 and their graphical representation is being shown in Fig. 8.

Experimental results show that ATSOT approach implemented in multi-lane approach, average queue length is less as that of existing OAF algorithm and pre-timed approach with or without Webster as the number of vehicles increases on the road. ATSOT approach outperforms all the other three methods. The ATSOT reduces the average queue length by 15 - 58% as compared with OAF. Further, ATSOT reduced the average queue length by 31 - 78% and 38 - 86% respectively in pre-timed approach with and without Webster's approach.

Table 3. Results for Average Queue Length in ATSOT, OAF algorithm and pre-timed with or without Webster's method in Multi-lane

Total Vehicles in Simulation	Average Queue Length (ATSOT)	Average Queue Length (OAF algorithm)	Average Queue Length (pre-timed)	Average Queue Length (pre-timed with Webster's)
200	3	3	21	10
400	8	19	44	28
600	12	27	62	55
800	34	40	76	62
1000	53	75	86	77

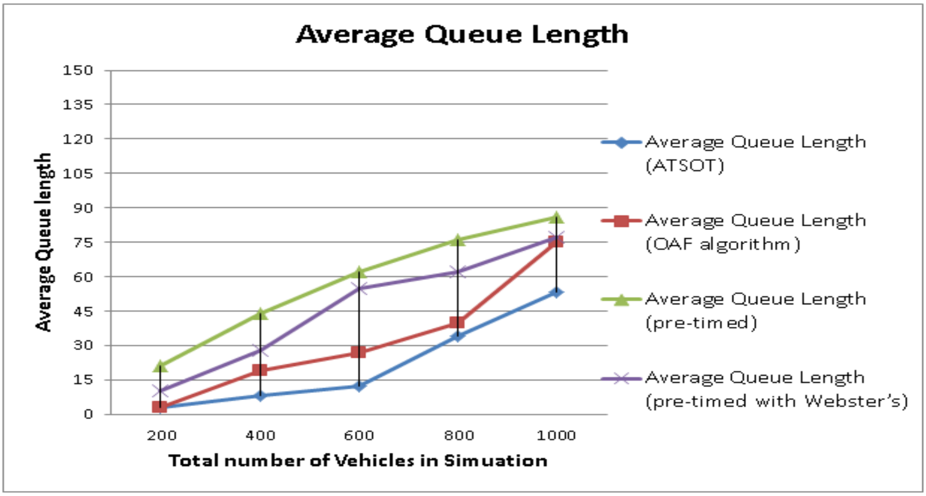

Fig. 8. Graph for Average Queue Length in ATSOT, OAF algorithm and pre-timed with or without Webster's method in Multi-lane

5 Conclusion

Design of Adaptive Traffic Signal using mOTes (ATSOT) system is presented in this paper. The traffic data is sensed through motes and information is collected about the traffic (e.g. position, density etc.) from the moving vehicles. This information is then used by the clustering algorithm to calculate the average queue length. The adaptive algorithm proposed in this paper is based on clustering to find the average queue length. Clustering algorithm also checks suitability of the cluster initiator for formation of stable clusters. Vehicle to vehicle, vehicle to infrastructure and within infrastructure communication is used in this work to set the green light time for crossroads in VANETs. The ATSOT system can help the commuters to get a congestion free path with least waiting time, resulting in overall trip reduction time, dip in the pollution levels, and reduction in fuel consumption. The ATSOT system has been simulated using open source softwares: MOVE, SUMO, TraCI and Python. The results obtained were compared with the pre-timed approach for single lane and multi-lane traffic without and with Webster's delay method and existing OAF algorithm for adaptive traffic signals. Experimental results show that, the proposed ATSOT approach reduces the average waiting time of vehicles at crossroads, as compared with the existing approaches.

Acknowledgement. The authors duly acknowledge the University of Delhi for the support in work on this paper under the research grant number RC/2014/6820.

References

1. Mathew, T.V.: Transportation Engineering I. Mumbai, India: Civil Engineering–Transportation Engineering. IIT Bombay, NPTEL ONLINE (2006)
2. Bell, M.G.H., et al.: Transport Planning and Traffic Engineering. John Wiley & Sons, NY (1997)
3. Webster, F.V.: Traffic signal settings. London: Road Research Laboratory Techn. Rep. No. 39, H.M.S.O (1958)
4. Hunt, P.B., Robertson, D.I., Bretherton, R.D., Winton, R.I.: SCOOT—A Traffic Responsive Method of Coordinating Signals, Transport and Road Research Laboratory, Crowthorne, Berkshire, England, Report TRRL 1014 (1981)
5. Mimbela, L.: Summary of vehicle detection and surveillance technologies use in intelligent transportation systems, Federal Highway Administration, USA, Intelligent Transportation Systems Joint Program Office (2000)
6. Gordon, R.L., et al.: Traffic control systems handbook,Federal Highway Administration, USA, Report FHWA-SA-95032 (1996)
7. Lowrie, P.R.: The Sydney Co-ordinated Adaptive Traffic System—Principles, Methodology, Algorithms. In: Proceedings of the International Conference on Road Traffic Signaling, Institution of Electrical Engineers, vol. 207, vol. 407, London, U.K, pp. 67–70 (1982)
8. Mirchandani, P., Head, L.: A Real-Time Traffic Signal Control System: Architecture, Algorithms, and Analysis. Transportation Research Part C: Emerging Technologies 9(6), 415–432 (2001)
9. Gartner, N.H.: Demand-Responsive Decentralised Urban Traffic Control, U.S.Department of Transportation, Washington, D.C., Report DOT/RSPA/DPB-50/81/24 (1982)
10. Turner-Fairbank Highway Research Center, Adaptive Control Software, Federal Highway Administration., USA, Report HRTS-02-037 (2005)
11. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: A survey. Computer Networks 38, 393–422 (2002)
12. Kahn, J.M., Katz, R.H., Pister, K.S.J.: Next century challenges: mobile networking for "Smart Dust". In: MobiCom 1999 Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, Washington, USA, pp. 271–278 (1999)
13. Bedi, P., Jindal, V.: Use of Big Data Technology in Vehicular Ad-hoc Networks. In: International Conference on Advances in Computing, Communications and Informatics (ICACCI 2014), Greater Noida, India, pp. 1677–1683 (2014)
14. Mohapatra, P., Krishnamurthy, S.V.: AD HOC Networks: Technologies and Protocols. Springer, Boston (2005)
15. Li, Z., Wang, Z., Chigan, C.: Security of Vehicular Adhoc Networks. In: Wireless Technologies in Intelligent Transportation Systems, pp. 133–174 (2011)
16. Dedicated Short Range Communication (DSRC) Home. (February 2014), `http://www.leearmstrong.com/DSRC/DSRCHomeset.htm`
17. Car-to-Car Communication Consortium. (April 2014), `http://www.car-to-car.org`
18. Priyanka, T., Sharma, T.P.: A Survey on Clustering Techniques used in Vehicular Ad Hoc Network. In: Proceedings of 11th IRF International Conference, Pune, Idia, pp. 174–180 (2014)

19. Balia, R.S., Kumar, N., Rodrigues, J.J.P.C.: Clustering in vehicular ad hoc networks: Taxonomy, challenges and solutions. Vehicular Communications 1, 134–152 (2014)
20. Vodopivec, S., Bester, J., Kos, A.: A Survey on Clustering Algorithms for Vehicular Ad-Hoc Networks. In: 35th Iternational Conference on Telecommunications and Signal Processing (TSP), pp. 52–56 (2012)
21. Sood, M., Kanwar, S.: Clustering in MANET and VANET: A Survey. In: International Conference on Circuits, Systems, Communication and Information Technology Applications (CSCITA), Mumbai, India, pp. 375–380 (2014)
22. Rawashdeh, Z.Y., Mahmud, S.M.: A novel algorithm to form stable clusters in vehicular ad hoc networks on highways. EURASIP Jounal on Wireless Communications and Networking 15, 2012 (2012)
23. Shivani, Singh, J.: Route Planning in Vanet By Comparitive Study of Algoriths. International Journal of Advanced Research in Computer Science and Software Engineering 3(7), 682-689 (2013)
24. Zhang, J., et al.: Data-Driven Intelligent Transportation Systems: Survey. IEEE Transactions on Intelligent Transportation Systems 12(4), 1624–1639 (2011)
25. Maslekar, N., Boussedjra, M., Mouzna, J., Labiod, H.: VANET based Adaptive Traffic Signal Control. In: IEEE 73rd Vehicular Technology Conference (VTC Spring), Budapest, pp. 1–5 (2011)
26. Gradinescu, V., Gorgorin, C., Diaconescu, R., Cristea, V., Iftode, L.: Adaptive Traffic Light Using Car-to-Car communications. In: IEEE 65th Vehicular Technology Conference, VTC2007-Spring, Dublin, pp. 21–25 (2007)
27. Khekare, G.S., Sakhare, A.V.: Intelligent Traffic System for VANET: A Survey. International Journal of Advanced Computer Research 2(4(6)), 99–102 (2012)
28. Fan, P., Haran, J.G., Dillenburg, J., Nelson, P.C.: Cluster-Based Framework in Vehicular Ad-Hoc Networks. In: Syrotiuk, V.R., Chávez, E. (eds.) ADHOC-NOW 2005. LNCS, vol. 3738, pp. 32–42. Springer, Heidelberg (2005)
29. Arkian, H.R., Atani, R.E., Pourkhalili, A., Kamali, S.: Cluster-based traffic information generalization in Vehicular Ad-hoc Networks. Vehicular Communications, 11 (2014)
30. Emmelmann, M., Bochow, B., Kellum, C.C.: Vehicular Networking: Automotive Applications and Beyond. John Wiley & Sons Ltd., United Kingdom (2010)
31. Gordon, R.L., Warren, T.: Traffic Control Systems Handbook. Washington, DC: Federal Highway Administration Report FHWA-HOP-06-006 (2005)
32. Roess, R.P., Prassas, E.S., McShane, W.R.: Traffic Engineering, 4th edn. Pearson Higher Education, NY (2011)
33. Wehrle, K., Güne, M., Gross, J.: Modeling and Tools for Network Simulation. Springer, Heidelberg (2010)
34. Kartik Pandit , Dipak Ghosal, H. Michael Zhang, and Chen-Nee Chuah, "Adaptive Traffic Signal Control With Vehicular Ad hoc Networks," IEEE Transactions on Vehicular Technology, vol. 62, no. 4, pp. 1459-1471, May 2013.
35. Lan, K.C., Chou, C.M.: An Open Source Vehicular Network Simulator. In: OSDOC 2011, Lisbon, Portugal, pp. 1–2 (2011)
36. DLR and contributors SUMO-Homepage (2014, October), http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/
37. Behrisch, M., Bieker, L., Erdmann, J., Krajzewicz, D.: SUMO - Simulation of Urban MObility - an Overview. In: The Third International Conference on Advances in System Simulation, Barcelona, SIMUL 2011, Spain, pp. 55–60 (2011)

38. Vaza, R.N., Parmar, A.B., Kodinariya, T.M.: Implementing Current Traffic Signal Control Scenario in VANET Using Sumo. In: ETCEE - 2014, National Conference on Emerging Trends in Computer & Electrical Engineering, India, pp. 112–116 (2014)
39. Wegener, A., et al.: TraCI: An Interface for Coupling Road Traffic and Network Simulators. In: Proceedings of the 11th Communications and Networking Simulation Symposium, CNS 2008, Ottawa, Canada, pp. 155–163 (2008)
40. TraCI/Change Traffic Lights State (June 2014), http://sumo.dlr.de/doc/current/docs/userdoc/TraCI/Change_Traffic_Lights_State.html
41. Vaishali, D.: Mobility Models for Vehicular Ad-hoc Network Simulation. International Journal of Computer Applications 11(4), 8–12 (2010)
42. Google's python class - Google for education - Google Developers, https://developers.google.com/edu/python/