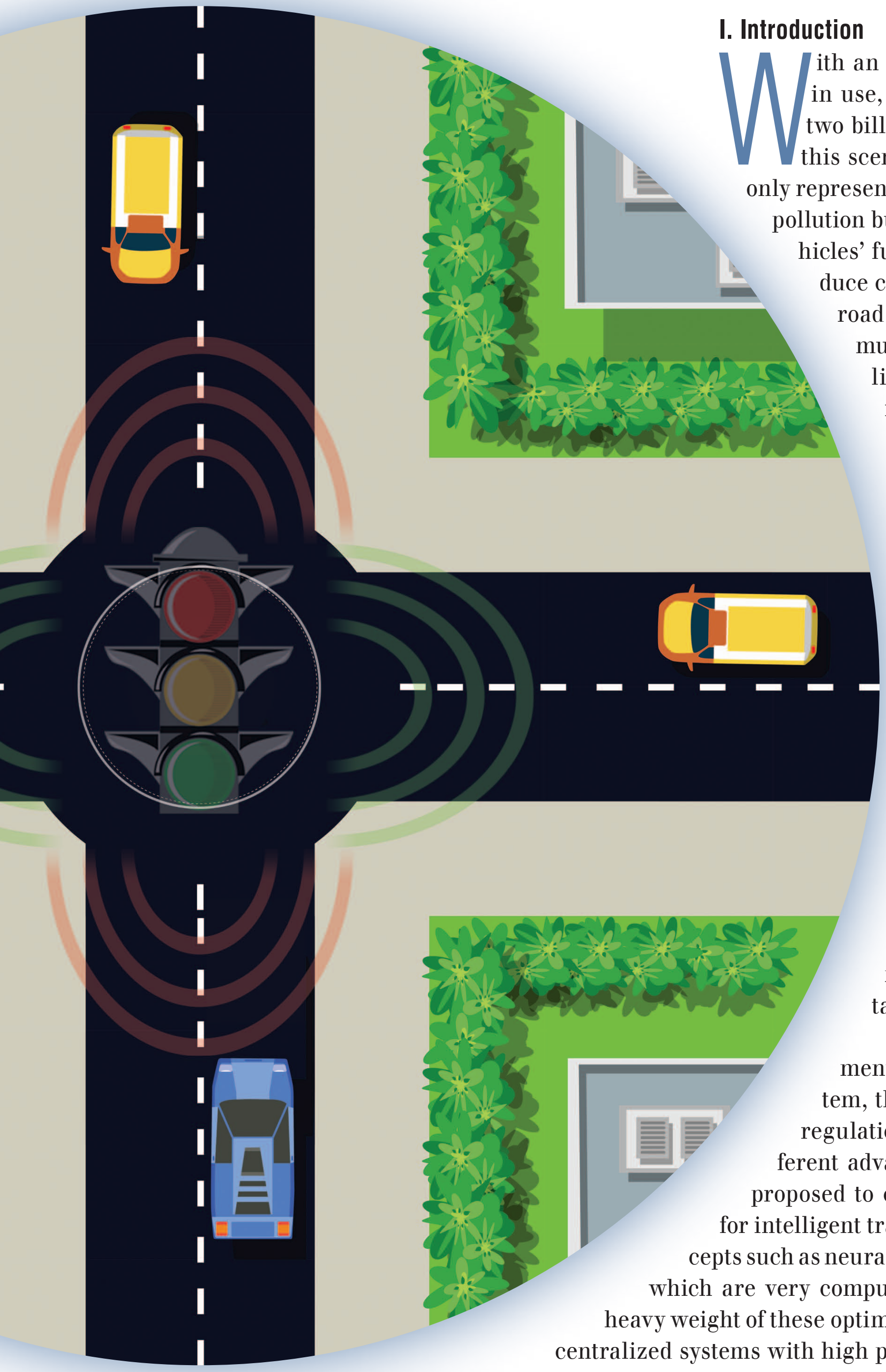
On the Similarities Between Urban Traffic Management and Communication Networks:

Application of the Random Early Detection Algorithm for Self-Regulating Intersections

Abstract—Communications are revolutionizing transport systems. Recently, novel technologies from the fields of telecommunications and data networks have been incorporated into both vehicle systems and road infrastructures, setting new frontiers in the design and development of Smart Cities and Smart Urban Mobility. Whereas each scenario, i.e., the vehicular one and the data communications one, has its own and well-defined particularities, there are also similarities that should be further explored. In this work, we address this confluence of research areas by investigating the applicability in vehicle traffic management of an active queue control algorithm employed in communication networks and called Random Early Detection (RED). RED provides congestion avoidance in data networks by controlling the average queue size of network buffers. It detects incipient congestion and, using probabilistic packet-marking or packet-dropping behavior, it is able to keep the data network at high throughput and low delay. In this work, we propose an adaption of RED to vehicle traffic management in signaled intersections, namely, RED for Vehicles (REDV). A complete characterization of REDV is provided, including the impact of several configuration parameters. We also conduct an extensive comparative performance evaluation via computer simulation. Results show that REDV is a light and simple algorithm able to significantly reduce the average trip time, waiting time, and CO_2 emissions in a signaled intersection compared to other approaches from the related literature.

Ramon Sanchez-Iborra
The Information and Communication Engineering Department, University of Murcia, Spain, E-mail: ramonsanchez@um.es

Maria-Dolores Cano
The Department of Information and Communication Technologies, Universidad Politécnica de Cartagena, Spain, E-mail: mdolores.cano@upct.es



I. Introduction

With an incessant worldwide growth of motor vehicles in use, it is expected that the world's fleet will reach two billion units by 2035 [27]. A direct consequence of this scenario is congestion and traffic jams, which not only represent an important source of atmospheric and noise pollution but also increase drivers' travelling time and vehicles' fuel consumption. As a result, all these facts reduce citizens' quality of life. Since the enlargement of road infrastructures is not always a feasible solution, much research effort is devoted to achieving Intelligent Transportation Systems (ITS) [16]. The incorporation in ITS of alternative (greener) powertrain technologies and telecommunications represents a turning point. For instance, vehicular communications has drawn the attention of the research community for several years [16]. Under this paradigm, vehicles communicate with each other (Vehicle to Vehicle, V2V) or with the signaling infrastructure (Vehicle to infrastructure, V2I) in order to optimize the traffic flow, among other aims. However, the real deployment of V2V, V2I, and greener technologies is still in an early stage, mainly due to its deployment complexity, economic limitations, and the still ongoing work on standards development [23]. Therefore, traffic management, from the perspective of traffic light control or intelligent signaling, becomes a more and more relevant short/medium-term approach to tackling the problem of traffic congestion.

Although traffic lights are not the only element to be considered within the traffic control system, they have a significant weight in the successful regulation of urban traffic. As technology evolves, different advanced mathematical methodologies have been proposed to control traffic signals [12]. Popular algorithms for intelligent traffic control are usually based on complex concepts such as neural networks, fuzzy logic, swarm intelligence, etc., which are very computationally demanding [3]. Thereby, due to the heavy weight of these optimizing operations, they are usually performed in centralized systems with high processing capacity. For that reason, sometimes

> The real deployment of Vehicle to Vehicle, Vehicle to Infrastructure and greener technologies is still in an early stage, mainly due to its deployment complexity, economic limitations, and the still ongoing work on standards development.

the processing and information-dissemination time is too long to take real-time countermeasures. Conversely, distributed approaches, understood as a set of self-governing control-nodes collaborating with each other in real time for regulating traffic [9], [11], present many advantages over traditional centralized solutions. For instance, the former systems usually present higher robustness, as they do not rely on a central computer or a wide wired communication network, are easier to deploy and manage, and show more flexibility due to their inherent modularity. In addition, with the continuous growth of both urban areas and number of vehicles, controlling an increasing number of intersections from a single central point is not the most scalable solution. For these reasons, distributed systems composed of autonomous nodes have gained momentum, and many proposals for improving the performance of isolated intersections or the coordination among them can be found in the related literature [8], [9], [13], [28]. In addition, many proposals, either centralized or distributed, often employ historical traffic records to perform their calculations instead of using real-time data for a more accurate adjustment of traffic control [19]. This approach can be problematic because historical databases do not always correlate well with real traffic conditions [4].

On the other hand, vehicular traffic management is not the only research area that suffers the problem of congestion. Telecommunication networks also need to properly dimension the communication infrastructure and, of course, they should also react to efficiently handle congestion periods. In these networks, congestion has a straightforward effect on Quality of Service (QoS). The term QoS means that the communication network should provide each service or application (e.g., video streaming, e-mail, voice over the Internet Protocol, web navigation, etc.) with its corresponding level of quality, mainly measured in terms of throughput, packet losses, delay, or jitter; all factors that strongly rely on avoiding network congestion. Similarly, we have the concept of Level of Service (LoS) in traffic science [6]. LoS stands for the level of quality that can be derived from a local (e.g., a signaled intersection) under different operation characteristics and traffic volume and it is obtained from performance metrics such as delay, speed, travel time, etc. It is clear that each scenario, i.e., vehicular and data networks, has its own and well-defined particularities. For instance, in communication networks a message is broken into small blocks of information called data packets that travel from an origin to a destination; these data packets can be individually rearranged, re-scheduled, shaped, or dropped/eliminated along their path, and vehicles cannot. Another difference is that vehicular traffic is affected by pedestrians or by the occurrence of vehicular spillbacks or gridlocks, with no direct correspondence in data networks. Nevertheless, important similarities also exist and should be further explored and exploited.

To provide QoS in telecommunication networks, queue management techniques (e.g., Tail Drop, Random Early Detection, etc.) and scheduling algorithms (e.g., Fair Queuing, Priority Queuing, etc.) are widely used solutions [25]. These techniques focus on traffic (data) engineering or traffic management from the telecommunication point of view. Therefore, in this work we propose the revision and adaption of an active queue management algorithm called Random Early Detection (RED) [17] to the vehicle traffic management arena. This new proposal is called RED for Vehicles (REDV). RED is a lightweight algorithm widely employed in telecommunication networks to deal with dense traffic flows in routers, which are devices that interconnect two or more communication networks and could be seen as similar to traffic intersections. The main goal of RED is to detect congestion before it occurs, aiming to take countermeasures to avoid it and hence mitigating its negative effects. REDV has the same objectives but applied to vehicle traffic management. Due to its low computational demands, we demonstrate that REDV is eligible to be installed in autonomous control nodes with processing restrictions, being able to provide efficient signaling timing plans for the traffic lights that comprise an intersection using real-time traffic data. Therefore, the main contributions of this work are the following:

- A state of the art that discusses the fundamental similarities between telecommunication networks and a road management system. This discussion is addressed through the exploration of common strategies adopted for avoiding or alleviating congestion events in both types of networks.
- A novel proposal based on the RED algorithm that is able to effectively manage vehicular traffic at signaled intersections. The new method is called RED for Vehicles (REDV).
- A complete characterization of REDV, exploring the impact of tuning its configuration parameters for different traffic conditions.

- A comparative performance evaluation based on extensive computer simulations in terms of overall trip time, waiting time, and CO_2 emissions. The obtained results are compared with those achieved by employing static timing plans and the algorithm proposed in [22].

An intersection can be understood as a series of queues trying to access a shared resource (the street), which is basically the same foundation a router is based on (in this case the shared resource would be the network link).

The rest of the article is organized as follows. Section 2 presents a literature review, focused on those papers proposing adaptive or intelligent traffic control methods that share foundations with QoS algorithms of communication networks. The proposed adaption of RED, REDV, is thoroughly described in Section 3. Section 4 overviews the test bench employed in the performance evaluation of REDV. The attained results are presented and discussed in Section 5. The paper ends by summarizing the most relevant findings and suggesting future research directions.

II. Related Work

Vehicular networks [21] have notably narrowed the distance between two well-differentiated research areas: transportation systems and telecommunication networks. The new wave of smart city-related services and the ultimate on-board communication technologies are enabling communication between vehicles (V2V) and between vehicles and the traffic control infrastructure (V2I) [9], [14]. Regarding traffic management, there are notable similarities between road traffic control and computer-networks management. Both systems suffer the issue of congestion, which occurs when system capacity is exceeded by traffic demand. Besides, once the road infrastructure or the network architecture is in its production stage, planning a system extension is expensive and difficult to tackle. Therefore, most efforts should focus on improving the overall system performance by using available resources or by installing cheap and simple controlling methods [19]. The main controversial points in both types of networks are those locations where several traffic flows converge, i.e., intersections in vehicle traffic and routers in telecommunication networks. An intersection can be understood as a series of queues trying to access a shared resource (the street), which is basically the same foundation a router is based on (in this case the shared resource would be the network link). However, the plethora of flexible actions that can be taken in routers is notably reduced in traffic intersections. Whereas data packets can be individually re-arranged, re-scheduled, shaped, dropped, etc., in traffic junctions the possible actions are basically re-scheduling (i.e., to provide alternative routes for vehicles) or adaptively tuning the phases of the traffic lights managing the intersection.

Although not explicitly mentioned, many techniques employed for managing vehicular traffic are similar or based on QoS strategies used for controlling communication networks. Three basic QoS approaches are: (i) best effort [25], (ii) Integrated Services (IntServ) [10], and (iii) Differentiated Services (DiffServ) [5]. In the former, all data packets are treated equally, without any distinction or priority between them, so the network does its "best" to deliver these packets but without providing any guarantees; this could be the case of common streets or arterials with no special lanes to prioritize some flows. In IntServ, packets only cross the network if a previous resource reservation has been made according to these data traffic needs (e.g., in terms of delay) via the well-known Resource ReSerVation Protocol (RSVP) [26]; this strategy could be seen in [15], where authors present a transportation system in which vehicles try to reserve some amount of space and time to cross an intersection. If the reservation request is not approved by the intersection manager, the vehicle would have to find another path towards its destination. Finally, packets are differentiated and prioritized according to their QoS needs (e.g., delay or throughput) in the DiffServ approach, creating several classes of data traffic that receive a particular treatment in the network nodes along their path; this could be the case of streets or arterials with designated lanes for specific types of vehicles (e.g., public transport, bikes, etc.) or when priority vehicles (e.g., ambulances) are detected and traffic lights along their path are synchronized for giving them access right away until their destination. For instance, this last idea has been deeply explored and numerous alternatives for emergency vehicle detection have been proposed in the related literature [31], e.g., strobe lights, acoustic systems, infrared transmitters, radio frequency communications, or GPS systems.

Furthermore, other works specialized in intelligent vehicle traffic management made use of similar concepts tackled by more advanced QoS and DiffServ strategies. The Explicit Congestion Notification (ECN) algorithm [18] is a method by which a congested router notifies this event to previous routers (from where data packets arrive) in order to request lowering the data-packet sending rate or even a change in the packet route. This strategy can be understood in the field of vehicle traffic control as detecting a congestion event and disseminating this information with the aim of suggesting (or forcing) approaching vehicles to change

Although not explicitly mentioned, many techniques employed for managing vehicular traffic are similar or based on Quality of Service strategies used for controlling communication networks.

their route towards their destination. Either the traffic control infrastructure [30] or the vehicles themselves [34] could be in charge of congestion detection and notification. In the former [30], a centralized server collects information from Road Side Units (RSU). When a congested road is detected, it is notified to the vehicles' on-board units, which remove this road from their map and recalculate a new route to the destination. In the latter, Yuan *et al.* [34] made use of Vehicular Ad-hoc NETworks (VANETs) for detecting and disseminating congestion events in urban expressways. Vehicles estimate traffic density from the analysis of status data included in the periodically transmitted beacons of surrounding vehicles. When congestion is detected, this information is broadcasted hop-by-hop to inform the upcoming vehicles, so part of the traffic flow can be diverted to auxiliary roads before reaching the congested area.

Another widely employed mechanism for managing the queues of a network router is Fair Queuing (FQ) [29]. Under this scheme, the overall bandwidth is fairly distributed among the queues composing the system according to the current length of each queue. This same methodology was employed in [22] for managing traffic intersections. In that work, the overall traffic coming into a junction is monitored, measuring the contribution of each incoming edge to the traffic volume as a whole. Thus, the signaled intersection cycle length is proportionally distributed among the different incoming roads. Due to the notable performance showed by this algorithm, we will use it for comparison purposes in the next sections. From the telecommunications perspective, one limitation of FQ is that all queues are equally treated. There are some services such as real-time communications that present stringent temporal restrictions and need to be prioritized over other services, e.g., video conference vs. e-mail. This is achieved by means of the Weighted Fair Queuing (WFQ) algorithm [24]. With WFQ, data-packet queues are categorized according to different flow classes. Thus, the bandwidth dedicated to each of these queues can be tuned according to their priorities. This strategy is similar to that employed in vehicle management by the Earliest Deadline First (EDF) method [1]. Following this scheme, the lower-priority queues (street edges) are not served until the higher-priority queues are empty. Authors suggested assigning different priorities to different types of vehicles. Thereby, those arms within an intersection having a vehicle with the highest priority (lowest deadline) are served first.

As stated previously, we propose to apply the Random Early Detection (RED) algorithm [17] inherited from communication networks in traffic-signaled intersections. Basically, this algorithm monitors the involved queues aiming at detecting incipient congestion. When this occurs, preventive measures are taken in order to mitigate or avoid congestion. A detailed description of RED is presented in Section 3. There are two previous works that attempted to employ the RED foundations for managing isolated traffic intersections, but from different perspectives than that proposed in this work. First, Xiao *et al.* presented a monitoring system for detecting congestion based on wireless sensor networks [33]. Once a congestion event is detected, the vehicles coming into the conflictive area are rerouted with a certain probability following the RED algorithm. With higher levels of congestion, the probability of being rerouted to alternative paths increases. Differing from this work, our goal is not re-routing traffic but adapting the traffic light signaling so that the traffic system reacts before congestion occurs, as we will explain in the next sections. Second, work by Alabdallaoui *et al.* [2] introduced a particular adaption of RED. As stated by the authors, a previous study is compulsory in order to account for the scenario's traffic demand on a per-hour basis. The RED algorithm then uses this information to detect congestion events and to adapt the green time for evacuating a certain number of vehicles. The main drawbacks of this work are: (i) a previous study is needed, which will not always correlate well with real traffic conditions; (ii) the evacuation rate function is not rigorously defined; and iii) the presented results should be extended, aiming at validating the algorithm's performance.

III. Random Early Detection for Vehicles (REDV)

In this section, we first explain the RED algorithm used in telecommunication networks and then illustrate our proposal as an original adaption of RED to vehicle traffic management.

A. Random Early Detection

The RED (Random Early Detection) algorithm, also known as random early discard, is one of the most widely employed active queue management methods in data networks [17]. As stated in previous sections, RED performs countermeasures when the first congestion evidences are detected. It is usually configured in routers with a high volume of traffic, i.e., managing high quantities of data packets. These packets are located in the router's buffers (please note that the terms buffer and queue are exchangeable). There are several options to select in which buffer a packet is put, e.g., based on the destination address of the packet, the type of service the packet belongs to, etc. Then, packets are served (sent to the next communication router in their path) as soon as possible. It is assumed that queues follow a First In First Out (FIFO) discipline. RED monitors each individual queue as an isolated system and obtains the number of packets waiting to be served, i.e., the queue length. When this value exceeds a preset threshold (min_{th}), the next arriving packet is discarded with probability p_a. This probability is assessed as a function of the queue length; thus, the larger the queue the higher the dropping probability of an incoming packet. If the queue length is greater than a maximum preset threshold (max_{th}), every incoming packet is discarded. This behavior is depicted in Fig. 1. It is important to note that RED actually uses an exponential weighted moving average of the queue length (*avg*) and not the real (instantaneous) number of packets in the queue. The strategy of employing an exponential weighted moving average acting as a low-pass filter reduces the effect of short-term growths in the *avg* value. Consequently, the dropping probability is not highly altered. In other words, short-term variations in the queue size due to bursty traffic or highly transitory congestion do not affect the long-term system performance because RED accommodates transient congestion by controlling the *avg* value, keeping at the same time high throughput and low average delay in data networks. Note that bursty traffic is very common in data networks and it is understood as a sudden increase in data network traffic, idle periods followed by bursts of data packets.

Therefore, the RED algorithm has two well-differentiated steps. First, the exponential weighted moving average of the queue has to be computed. This task is performed as (1) indicates, where the w_Q factor determines the time constant of the low-pass filter and Q is the current queue length. By tuning w_Q, it is possible to adjust the weight given to the history (i.e., the previous average values) and to the current queue size (Q). Second, once the exponential weighted moving average *avg* is obtained, the probability of dropping a data packet (p_b) is calculated as shown in (2), where max_p is a factor that determines the maximum value for p_b. If we use this probability p_b to drop data packets, then what we get is that the number of data packets that arrive between dropped packets follows a geometric random distribution, which means that there will be several continuous data packets being discarded and then long periods without drops, as explained in [17]. If we want to have a uniform distribution of packets that arrive between dropped packets, then the dropping probability should be p_a (3). That is, the final probability of discarding a packet is adjusted to be slowly increased since the last packet was dropped, as indicated in (3), where *count* is the number of consecutive packets accepted since the last packet drop. RED nodes perform better when the packet-dropping probability changes fairly slowly as the exponential weighted moving average changes. This helps to discourage oscillations in *avg* and this is the reason to use p_a and not p_b as the packet dropping probability. For more information about RED please refer to [17], [20], [32].

$$avg = (1 - w_Q)avg + w_Q Q \tag{1}$$

$$p_b = \frac{max_p(avg - min_{\text{th}})}{(max_{\text{th}} - min_{\text{th}})} \tag{2}$$

$$p_a = \frac{p_b}{(1 - count \cdot p_b)} \tag{3}$$

B. Our Proposal REDV

Following the same strategy, the RED algorithm can be applied to manage signaled intersections in vehicular traffic. We call this new version RED for Vehicles (REDV). For simplicity, we first focus on a four-arm signaled intersection managed by four traffic lights that only allow through and right-turning movements, i.e., left-turns are not considered (see Fig. 2). Thus the full traffic light cycle is split

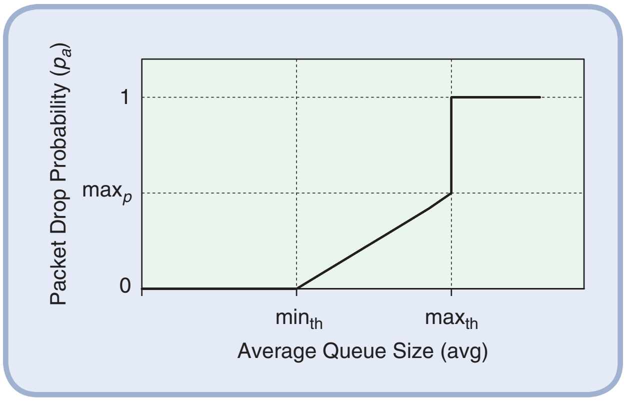


FIG 1 RED's strategy to drop data packets.

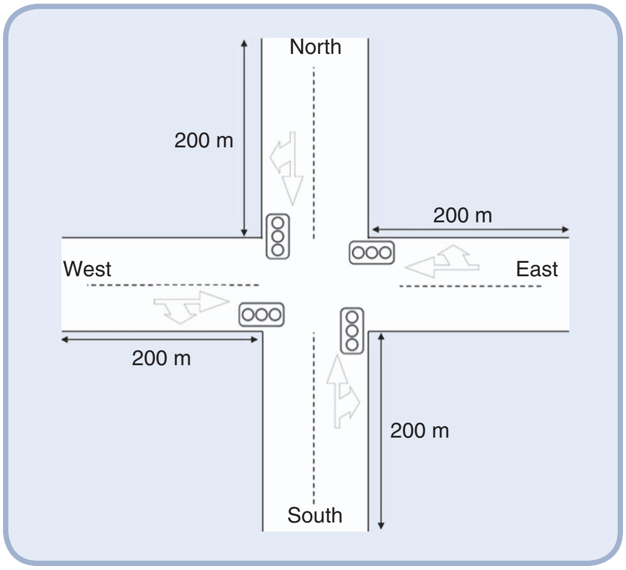


FIG 2 Scenario under consideration.

Algorithm 1 Pseudocode of REDV procedure

```
Initialization:
avg = 0; count = 1
end
for each clearance interval:
  calculate the new average queue size avg:
    if the queue is nonempty:
      avg = (1 − w_Q) avg + w_Q Q
    else
      m = current_time − Q_time
      avg = avg · (1 − w_Q)^m
    end
    end
    if (min_th ≤ avg < max_th)
      increment count
      calculate probability p_a:
        p_b = max_p (avg − min_th) / (max_th − min_th)
        p_b = p_b / (1 − count · p_b)
      end
      with probability p_a increase green time in current road:
        green_time = green_time + I
        count = 0
      end
  end
  else if (avg >= max_th) increase green time in current road:
      green_time = green_time + I
      count = 0
      end
  else
      count = −1
  end
  if (green_time > max_green_time) limit the maximum green time:
      green_time = max_green_time
  adjust the green time for the complementary road:
      green_time_complementary = cycle − green_time − 2(yellow_time + clearance_interval)
  end
  end
when queue becomes empty:
Q_time = current_time
end
```

into two phases coupling the opposite edges, and so we either have simultaneous traffic in the West-East, West-South, East-West, and East-North directions or in the North-South, North-West, South-North, and South-East directions.

The REDV procedure is as follows. Let us assume that the West-East traffic lights are green, then yellow, and then the clearance interval starts. At this point (during the clearance interval), the system will calculate an updated green time for the North and South edges. To do so, it counts the total number of vehicles waiting in the North and South edges. Let us call these values Q_N and Q_S, respectively. REDV then calculates the exponential weighted moving average value *avg* using (1). We use the maximum between Q_N and Q_S as the Q parameter employed in (1); that is, $Q = max(Q_N, Q_S)$. Please note that instead of using the maximum value we could simply add both queue lengths; the corresponding effect on the algorithm's performance will be discussed in the results section. Also, w_Q determines the weight given to the current queue size Q opposite to the history (i.e., the previous average value *avg*) as explained before. Once the exponential weighted moving average *avg* is computed, the probability p_a of modifying the traffic signal phases is calculated so that: if *avg* is below the preset min_{th} threshold, no action is performed; if *avg* is between the min_{th} and max_{th} thresholds, then the green time interval for the next green phase (green traffic lights in the North and South edges) will be increased with probability p_a; and finally, if *avg* is above max_{th}, then the green phase is increased with probability $p_a = 1$. That is, instead of dropping data packets (removing elements from the queue), REDV will probabilistically increase the green time interval of the traffic lights by a predefined amount of time (this increment, *I*, will be discussed later).

If we work with a fixed (closed) cycle length, enlarging the green phase on one street (North-South or East-West) implies decreasing the same amount of green time for the crossing phase (East-West or North-South, respectively). This can be seen as a friendly competition between the two streets or arterials for gaining a longer green time. Clearly, the larger the exponential weighted moving average *avg* of a street (North-South or East-West in the example of Fig. 2), the higher the probability of increasing its corresponding green time interval. Aiming at preventing one street from taking the entire available green time within a cycle, a maximum green time limit (*max_green_time*) is established. This limit is determined by a preset minimum green time (*min_green_time*), the yellow time (*yellow_time*), and the all-red time (*clearance_interval*) within the cycle (*cycle*) as (4) indicates.

$$\max_{green_{\text{time}}} = cycle - \min_{green_{\text{time}}} - 2(yellow_{\text{time}} + clearance_{\text{interval}}) \quad (4)$$

Algorithm 1 illustrates the pseudocode of the REDV procedure. There are several parameters in REDV that can be tuned depending on the scenario characteristics such as traffic density and traffic flow symmetry (i.e., whether or not traffic load is similar in the East-West and North-South directions). The min_{th} threshold controls when REDV starts to probabilistically update the green-time phases. The max_{th} threshold represents the limit to stop operating with probabilities and proceed deterministically, i.e., always increasing the green time if the condition is achieved. Regarding the low pass filter used in (1), w_Q allows the system's memory to be adjusted and decide which factor should have more weight in computing *avg*, either the window history (the previous exponential weighted moving average) or the current instantaneous queue length. The max_p parameter establishes the maximum probability before entering the deterministic behavior (see Fig. 1). Finally, the value of *cycle* should be set according to global traffic conditions, as we will explain later. The specific values for each of these variables will also be discussed in upcoming sections.

Besides the already-defined parameters, the following variables are also employed in REDV as shown in Algorithm 1. m represents the time that has elapsed since the last time the queue was empty (Q_{time}). $green\_time$ is the new calculated green time interval for the upcoming phase. I accounts for the predefined amount of time that is probabilistically added to the new green time interval. Finally, $green\_time_{\text{complementary}}$ is the corresponding green time interval of the complementary road, i.e., for the opposite edges; for instance, if $green\_time$ is calculated for the traffic lights in edges East-West then $green\_time_{\text{complementary}}$ is the green time interval for the traffic lights in edges North-South. The value $green\_time_{\text{complementary}}$ is adjusted once the $green\_time$ value is calculated, as explained earlier.

Therefore, observe that when one of the directions (North-South or East-West) executes the REDV procedure for updating its green time, the time increment I will be added to its last green time interval, which in turn was computed by the complementary direction in the previous iteration. The consequences are twofold: (i) the cycle duration is always maintained and (ii) the green time duration of each arterial is conditioned by the state of the complementary one. As stated previously, this leads to a friendly "competition" between both roads for the available green time. When the same traffic load from the East-West and the North-South directions arrives at the intersection, a balance between the two green-time phases will occur. Whereas in the case of asymmetry, as more traffic is detected in one direction compared to others, REDV will update the traffic light green phases accordingly, but without starvation in any queue.

IV. Test-bench and REDV Configuration

In order to evaluate the performance of the proposed algorithm, the microscopic traffic simulator SUMO has been employed [35]. The single-lane four-arm intersection shown in Fig. 2 has been considered. Each of these arms was 200 m in length and the intersection was managed by four traffic lights controlled by the REDV algorithm. The algorithm was developed in Python and communicated with the core simulator via the TRACI (TRAffic Control Interface) package [35]. REDV collects traffic data (queue lengths) during the all-red periods (clearance interval) and computes the green-time duration for the upcoming signaling phase (see Algorithm 1). These data are collected by means of the ready-to-use methods provided by the TRACI library. Please note that the only input needed by REDV is the number of vehicles waiting in the corresponding edges before the lights turn green, so in a real implementation, this operation may be addressed by multiple approaches: wireless sensor networks, cameras, or inductive loops, among others.

Aiming to explore the performance of REDV under different traffic conditions, five different traffic patterns have been used (see Table I). In this Table, the "symmetric" term refers to traffic patterns with the same number of vehicles (on average) in both arteries and "asymmetric" patterns present distinct vehicle densities in the different arteries or even in the opposite directions within the same artery. The "constant" term refers to the non-variability of the traffic densities during the simulation runs, and the "variable" traffic patterns present non-constant traffic densities during the simulated period. We assumed that only through and right-turning movements are allowed. Thus, the possible routes within the studied scenarios were the following: N➔S, N➔W, E➔W, E➔N, S➔N, S➔E, W➔E, and W➔S. Observe that all routes were approximately 400 m in length and without the possibility of being blocked by crossing traffic. For each considered scenario, 10 simulation runs of 2 hours (7200 s) were carried out. For each simulation run, a set of vehicle routes was generated according to the traffic densities associated to each scenario (see Table I). These traffic flows were randomly generated following a Poisson distribution. Thus, the final results for each scenario were obtained by averaging the outcomes attained in each simulation run. The 95% confidence intervals were calculated too. For this study, the only vehicles taken into account were cars with an average length of 5 m and maintaining a security gap of 1.5 m. Therefore, the maximum number of vehicles within an edge (Maximum Queue Capacity, MQC) was $200/(5+1.5) \approx 30$ vehicles. The maximum permitted speed in every edge was set to 13.9 m/s (50 km/h), and the vehicle acceleration to 3 m/s^2. The yellow time interval was fixed

Table I. Traffic pattern scenarios for the performance evaluation of REDV via computer simulation.

Scenario	Denomination	North-South Edges (veh/h)	East-West Edges (veh/h)
1	Symmetric. Low density	NS & SN: 500	WE & EW: 500
2	Symmetric. High density	NS & SN: 1500	WE & EW: 1500
3	Asymmetric. Constant flows	NS & SN: 500	WE & EW: 1500
4	Asymmetric. Variable flows	NS & SN: 0 s – 3600 s: 500 3600 s – 7200 s: 1500	WE & EW: 0 s – 3600 s: 1500 3600 s – 7200 s: 500
5	Asymmetric. Constant flows	NS: 1000 SN: 500	WE:1000 EW: 500

Table II. REDV parameters and cycle values.

Parameter	Value
min_{th}	3 veh
max_{th}	12 veh
w_Q	{0, 0.25, 0.5, 0.75, 1}
max_p	{0, 0.25, 0.5, 0.75, 1}
I	5s
cycle	{60, 90, 120} s

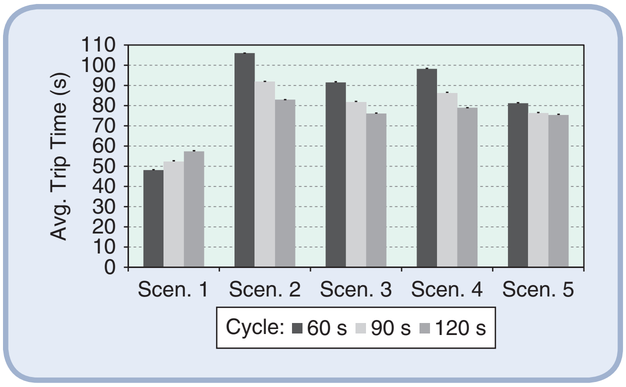


FIG 3 Average trip time and confidence intervals for the 5 traffic conditions under study. Fixed-time plan with different cycle lengths.

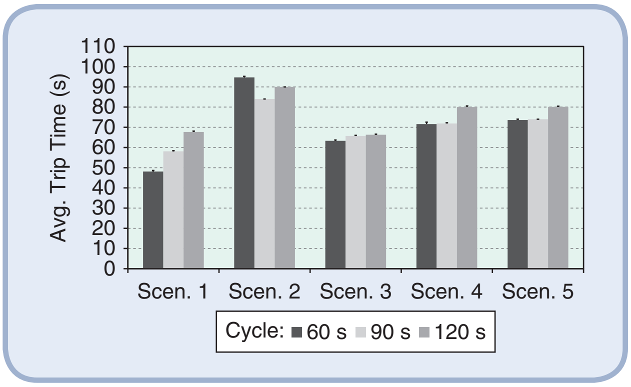


FIG 4 Average trip time and confidence intervals for the 5 traffic conditions under study. Fair Queuing-based algorithm with different cycle lengths.

to 3 s, the clearance interval to 2 s, and the minimum green time to 10 s. Consequently, the maximum green time was set as follows (see (4)): *max_green_time* = *cycle* – 10 – 2·(3+2) = *cycle* – 20 s. We used three different values for the cycle length, 60 s, 90 s, and 120s, whose effect is also discussed latter on.

Regarding REDV, there are five key parameters to be configured: min_{th}, max_{th}, w_Q, max_p, and I. After extensive simulation analysis, the min_{th} and max_{th} operation thresholds have been fixed to 10% and 40% of the MQC value, respectively, and the time increment I has been set to 5 s. The impact of the w_Q and max_p parameters on the system performance has been studied by varying their values, as indicated in Table II. In order to compare the performance of REDV, two other traffic management approaches for signaled intersections have been used: static phases (fixed-time plan) and the proposal by McKenney and White [22] that we called FQ-based due to its similarity to the so-called QoS algorithm.

For the former, the corresponding total cycle (60 s, 90 s, or 120 s) was equally shared between the green and red phases after subtracting the yellow and the clearance time intervals. Finally, widely-used key performance metrics have been taken into consideration for evaluating the three traffic management approaches: overall trip time, waiting time, and CO_2 emissions.

V. Results

In this section we discuss the results obtained from the simulated scenarios described above. Initially we study the best configuration for each traffic management mechanism, including the effect of the w_Q and max_p parameters on REDV and the approach followed for computing the exponential weighted moving average. We then compare the performance of REDV with the other evaluated approaches (fixed-time plan and FQ-based). Finally, we also investigate and present the results of REDV performance, including a self-adjust cycle.

A. Characterization of Algorithms

REDV, fixed-time plan, and FQ-based are individually studied in order to find their best configuration. First, Fig. 3 shows the average trip time and the confidence intervals for the five scenarios under study (see Table I) when managed according to the fixed-time plan strategy. For each scenario, three different cycle lengths have been tested. The obtained results confirm a well-known fact for static timing-plan strategies: With low traffic densities (Scenario 1 in Fig. 3), it is better to use short cycles, e.g., 60 s, than longer cycles [7]. In contrast, when traffic density is high (Scenario 2 in Fig. 3), the best results are obtained with the longest cycle (120 s). For high-density asymmetric traffic flows (Scenarios 3, Scenario 4, and Scenario 5 in Fig. 3), it is more appropriate to always employ the longest cycle because in these scenarios there are at least two edges with high traffic density. As shown in Fig. 4, a different behavior is observed in the FQ-based algorithm. In this case, the best results are always attained with the shortest cycle of 60 s (but Scenario 2). Finally, focusing on REDV, it is interesting to observe that contrary to the fixed-time plan, a higher traffic load does not imply using the greatest cycle length, as illustrated in Fig. 5.

Regarding the way the exponential weighted moving average is computed in REDV, we initially propose to use the maximum queue length between the two opposite edges as the instantaneous queue size Q employed in (1). For instance, if the intersection is at a clearance interval and the next green phase corresponds to North-South, then we would set Q as the maximum value between the North queue and the South queue sizes ($Q = max(Q_N, Q_S)$). Nevertheless, other approaches could be followed such as adding the lengths of opposite edges and setting Q as the resulting value ($Q = Q_N + Q_S$). The impact of these two methods is illustrated in Table III. We can observe that in most scenarios, it is better to use the maximum method. The only exception is Scenario 4, where addition seems more appropriate. When using the maximum approach, REDV is more proactive and there will be more competition more quickly between the crossing roads. On the other hand, the addition-based method is friendlier. Adding the queue sizes in Q will soften the exponential weighted moving average values and so does the probabilistic behavior. From a research point of view, we find more interesting to analyze the most proactive approach and this is the method that we use throughout this paper. Addition and other possible proposals for determining queue length are left for further study.

One key performance metric that directly impacts on drivers' mood is waiting time, which in the evaluated scenarios is represented by the time that the car is stopped at the signaled intersection waiting to cross it.

In addition to the cycle length, REDV has other parameters to be configured, which are: min_{th}, max_{th}, w_Q, max_p and I. Through extensive computer simulations we set min_{th} and max_{th} to 10% and 40% of the MQC, respectively, and I to 5 s as indicated in Table II. By using these values for the thresholds, we have adopted an aggressive strategy for increasing the green time when the first signs of congestion are detected. In the original RED algorithm for communication networks, min_{th} and max_{th} are usually fixed to more conservative figures because in data networks there is a loss of information (due to packet dropping), something that does not occur in the case of vehicle systems. Regarding w_Q and max_p, they are key in determining the weight of the vehicle traffic history versus the instantaneous size of the intersection edge queues. For that reason, Fig. 6 includes the performance of REDV in terms of average trip time for several values of w_Q and max_p, using the optimum cycle duration for each simulated scenario, namely 60 s for Scenario 1, 120 s for Scenario 2, and 90 s for Scenario 3, Scenario 4, and Scenario 5. Observe that the combination of both parameters, w_Q and max_p, has an important impact on the system's performance, especially in the asymmetric cases, e.g., Scenario 3 and Scenario 4. As stated earlier, when both crossing roads present similar traffic density, there is a balance between the "strength" of both sides in gaining green time, so the REDV configuration parameters have a lower impact on the final performance. However, when there is an imbalance between the traffic densities of both roads, especially when they are irregular (Scenario 4), the algorithm needs flexibility to self-adapt to these changes in the traffic pattern. This is achieved when w_Q and max_p are set

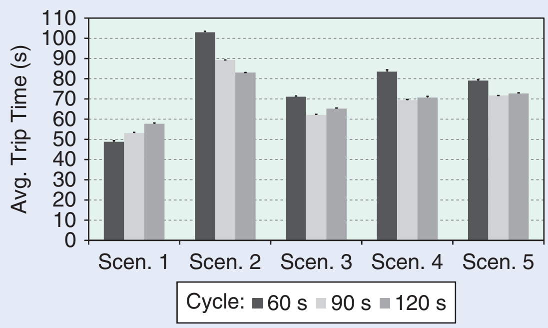


FIG 5 Average trip time and confidence intervals for the five traffic conditions under study. REDV algorithm with different cycle lengths and w_Q, = max_p = 0.5.

Table III. Performance comparison of REDV using two different approaches to determine queue length: maximum (max) vs. addition (add).

Scenario	1		2		3		4		5	
	Max	Add	Max	Add	Max	Add	Max	Add	Max	Add
Average trip time (s)	49	49	82	83.2	64.3	75	78.5	69.9	71.7	72
Average waiting time (s)	12.2	11.9	42.3	43.5	26	33.5	39	29	30.2	31.4
Average CO_2 emissions (g)	84.8	85.2	93.8	94.3	88.4	95.2	92.8	92.9	88.2	88.9

to values 0.5 and {0.5, 0.75}, respectively (see Fig. 6(c) and Fig. 6(d)). Consequently, in the following sections we will show the results obtained using REDV with $w_Q = 0.5$ and $max_p = 0.5$.

B. Comparative Performance Evaluation

In this Section, the outcomes achieved by REDV are compared with those obtained by the fixed-time plan strategy and by the FQ-based algorithm proposed in [22] under the same scenarios. Assuming the best configuration for each algorithm in terms of cycle length and REDV parameters, as explained earlier, Fig. 7 comparatively represents the average trip time and the confidence intervals for all evaluated scenarios and the three tested algorithms. In the symmetric cases (Scenarios 1 and 2), all methods present similar values for the average trip time; no matter the algorithm employed, figures of around 48 s and 83 s are achieved, respectively, for these simple traffic patterns. As stated before, symmetric traffic patterns without noticeable traffic variations are well-managed by the fixed-time plan strategy, and the adaptability of both the FQ-based and the REDV algorithms is not fully exploited. However, when the traffic pattern complexity increases, a notable difference is observed in the performance of the algorithms. Whereas the average trip times attained by the fixed-time plan strategy are 76 s (Scenario 3), 79 s (Scenario 4), and 75 s (Scenario 5), REDV reaches notable figures of 62 s, 69 s, and 71 s, respectively. This represents a reduction of around 19%, 13%, and 6%, respectively, on each vehicle's average time for crossing the evaluated scenario (Fig. 2). In turn, the FQ-based algorithm obtains better results than the fixed-time plan approach (63 s for Scenario 3, 72 s for Scenario 4, and 73 s for Scenario 5), but it is still worse than REDV.

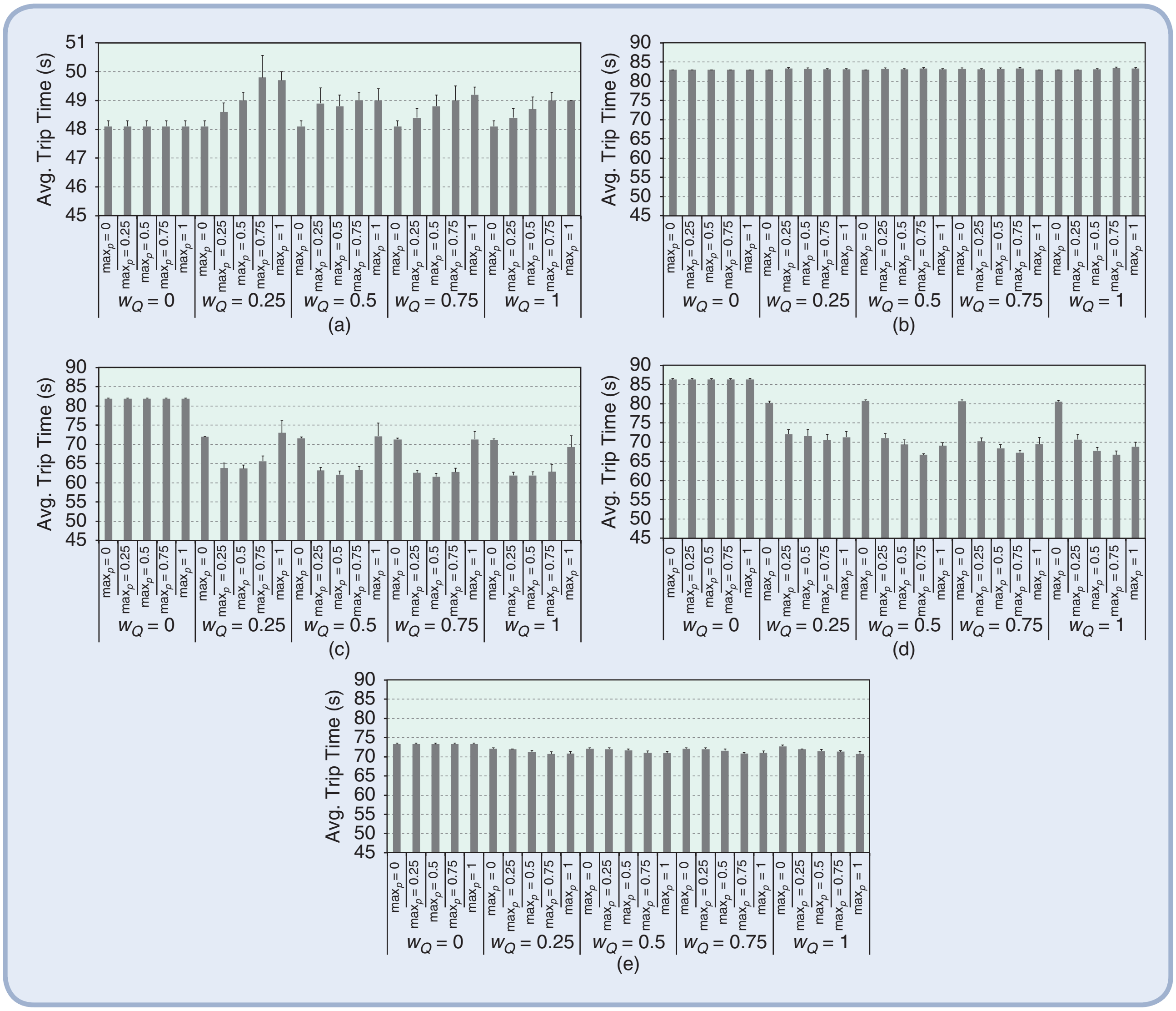


FIG 6 Average trip time and confidence intervals for the five scenarios under study. Characterization of REDV parameters w_Q and max_p. (a) Scenario 1, cycle = 60 s, (b) Scenario 2, cycle = 120 s, (c) Scenario 3, cycle = 90 s, (d) Scenario 4, cycle = 90 s and (e) Scenario 5, cycle = 90 s.

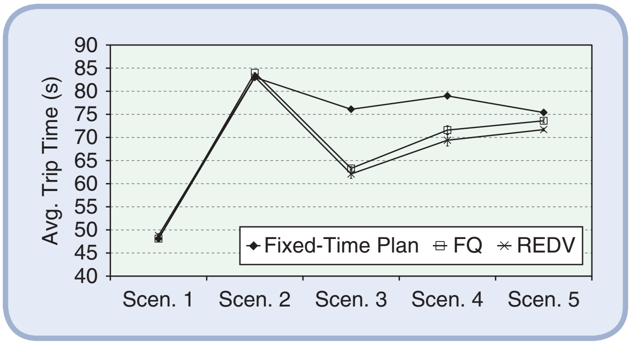


FIG 7 Average trip time and confidence intervals for the five traffic conditions under study attained by using a fixed timing-plan strategy, the Fair Queuing-based algorithm, and REDV.

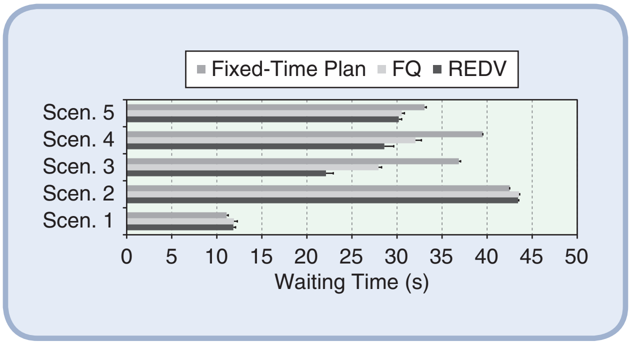


FIG 8 Average waiting time and confidence intervals for the 5 traffic conditions under study attained by using a fixed timing-plan strategy, the FQ-based algorithm, and REDV.

One key performance metric that directly impacts on drivers' mood is waiting time, which in the evaluated scenarios is represented by the time that the car is stopped at the signaled intersection waiting to cross it. Observe in Fig. 8 how the waiting times are similar in Scenario 1 and Scenario 2, no matter the algorithm employed. Nevertheless, in the asymmetric scenarios (Scenario 3, Scenario 4, and Scenario 5), there is a substantial difference among the approaches. With REDV, a notable reduction of about 40% (Scenario 3), 27% (Scenario 4), and 9% (Scenario 5) is achieved in terms of waiting time compared with the values reached by the fixed-time plan strategy. In addition, this metric is reduced by about 22% (Scenario 3), 12% (Scenario 4), and 2% (Scenario 5) if we compare REDV with the FQ-based algorithm. Another performance measurement of importance is the level of pollution generated by road vehicles. Due to the high volume of traffic within highly populated cities, it is necessary to control the level of harmful gases emitted into the atmosphere. Therefore, Table IV shows a comparison of the average CO_2 emissions (in g) measured during the complete set of simulations (10 runs of 7200 s for each scenario). Please note that, as in the previous discussion, the presented results have been obtained from the best configuration of each algorithm in terms of cycle and REDV parameters. There is a slight reduction in CO_2 emissions when employing the REDV algorithm in comparison with the fixed-time plan strategy. Again, the biggest difference between both approaches is found in the case of complex traffic patterns (Scenario 3, Scenario 4, and Scenario 5). Comparing REDV and the FQ-based approach, the latter obtains better results in the symmetric scenarios (Scenarios 1 and Scenario 2), but its performance worsens in most complex traffic conditions (Scenario 3, Scenario 4, and Scenario 5) in terms of CO_2 emissions.

C. REDV with Self-Adjusted Cycle

So far, we have seen in the provided results that one determinant parameter that requires manual tuning to adapt it to current traffic conditions is cycle duration. Depending on the traffic pattern and on the specific traffic management algorithm in use, different cycle durations could be recommended. We therefore propose to add a new feature to REDV that consists in the self-adaption of the cycle length according to real-time traffic density. Thus, fixing the upper and lower cycle-length limits to 60 s and 120 s, respectively, we allow the free adjustment of the cycle by steps of {5, 10, 20, and 30} s.

Table IV. Average CO_2 emission (g).

Scenario	1	2	3	4	5
Fixed-time plan strategy	84.93	94.35	92.31	93.29	87.93
FQ-based algorithm	84.37	93.97	92.65	96.54	90.48
REDV	84.93	94.33	91.14	92.96	87.88

It was previously demonstrated that, in general, higher traffic volumes require longer cycles. The following strategy has hence been adopted. First, REDV automatically decreases the cycle length if the exponential weighted moving average *avg* is below the min_{th} threshold a predefined number of consecutive times, considering both arterials. We call this event *decr_occurr*. Second, if the green-phase interval has been successfully increased a predefined number of consecutive times, taking into account both arterials, the cycle length is increased. We call this event *incr_occurr*. Through computer simulations, we conclude that proper values for *decr_occurr* and *incr_occurr* are 4 and 6, respectively. Please note that with these figures it is easier to perform a decrease than an increase action. With this strategy, we intend to facilitate shorter cycles aiming at both speeding up traffic and increasing the mobility of the traffic flows.

Fig. 9 shows the results attained in the considered scenarios when this modification is incorporated in REDV. At each simulation run, the cycle is always initialized to 60 s and will be automatically adjusted by REDV following the previous procedure in the course of the simulation. For comparison

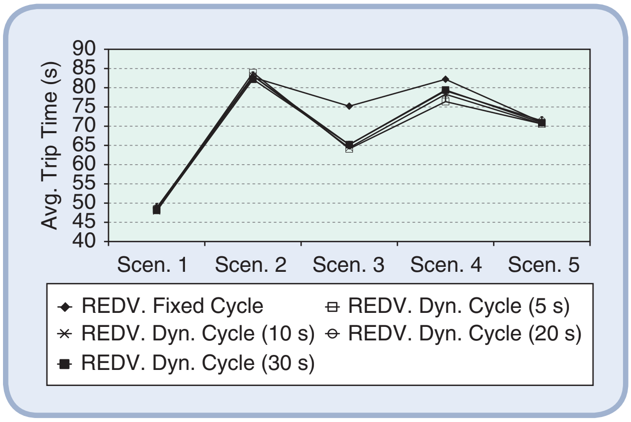


FIG 9 Average trip time and confidence intervals for the five traffic conditions under study. Comparison between REDV with a fixed cycle (the best one for each scenario) and REDV with self-adjustment.

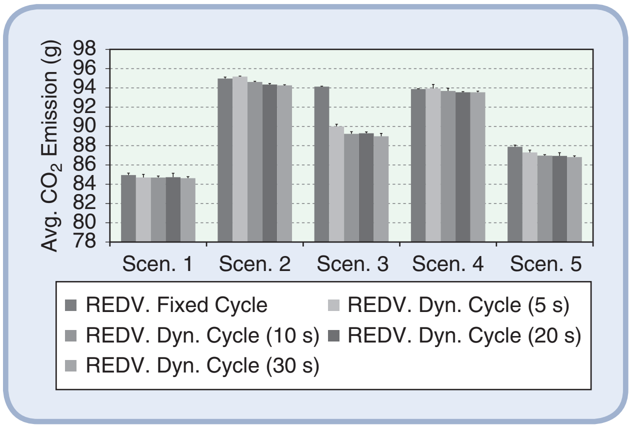


FIG 10 Average CO_2 emission and confidence intervals for the five traffic conditions under study. Comparison between REDV with a fixed cycle (the best one for each scenario) and REDV with self-adjustment.

purposes, Fig. 9 also represents the results obtained with REDV employing a fixed cycle (the best cycle for each scenario). Observe how the strategy of self-adjusting the cycle succeeds, evidenced by the shorter average trip times obtained in all the evaluated scenarios, above all by using the shortest step of 5 s. These promising results are further confirmed in terms of pollution emission. Observe in Fig. 10 that the CO_2 emissions are always lower using the self-adjusted cycle approach than using the manually configured cycles. Finally, this modification does not alter the computational advantages of REDV; low memory requirements are needed and very simple programming instructions are used, as show in the pseudo-code included in Algorithm 1.

VI. Conclusion

As well as transportation systems, telecommunication networks also suffer the negative impact of congestion events. The existing similarities between these two areas of research, in particular to address the congestion issue, led us to propose the incorporation of an active queue management technique inherited from data networks into vehicle traffic management. After describing the resemblances of several methods from the related literature used in both data networks and transport systems to improve traffic control, we presented the adaption of the Random Early Detection (RED) algorithm to be used in vehicle traffic management. Our proposal, so-called RED for vehicles (REDV), was characterized and the effect of several configuration parameters in its performance was shown. REDV was then compared via computer simulation with both a classic fixed-time plan strategy and an adaptive traffic signal control from the related literature. Within an isolated signaled intersection, several traffic patterns were considered in the performance evaluation. The results showed the superior performance of REDV in terms of average vehicle trip time, waiting time, and CO_2 emission, with improvement figures up by 19%, 40%, and 4%, respectively. In addition, we introduced a new feature for REDV: the dynamical adaption of the cycle length to current traffic conditions. This improvement allowed the efficiency of the algorithm to be further increased, especially in scenarios with non-constant traffic patterns. In summary, REDV shows consistency, efficiency, and reliability in managing an urban intersection without high computational demands. As a future work, we plan to extend the REDV operation to several intersections. It will be necessary to identify if an urban network applying REDV is able to self-synchronize, given that each intersection is self-configured and adapts to traffic conditions by itself. On the other hand, we are working on an affordable implementation of the complete system. It is necessary to evaluate different methodologies to obtain the real traffic data that serve as input to REDV, e.g., cameras and image processing versus sensor systems, taking into account that complexity should be kept low and efficiency and accuracy high, so that REDV can be deployed and tested in real scenarios. In our humble opinion, this work opens new and interesting possibilities in adaptive traffic control and congestion management by highlighting how mechanisms from data networks are transferable to Intelligent Transportation Systems.

Acknowledgment

This work was supported by the Dirección General de Tráfico (DGT), Ministerio del Interior (Spain), project grant SPIP2015-01780 (PROA).

About the Authors

Ramon Sanchez-Iborra is an Assistant Professor and Researcher at the Information and Communication Engineering Department in the University of Murcia, Spain. From the Technical University of Cartagena (Spain), he received the BSc degree in telecommunication engineering in 2007 and the MSc and PhD degrees in information and communication technologies in 2013 and 2016, respectively. He has been a visiting researcher at the NetGNA group in the Universidade da Beira Interior, Portugal, and the Quality and Usability Lab in the Technische Universität Berlin/Telekom Innovation Laboratories, Germany. His main research interests are evaluation of QoE in multimedia services, management of wireless mobile networks, green networking techniques and IoT/M2M architectures. He has published several papers in national and international conferences and journals. He has collaborated as a TPC member and reviewer for international journals and conferences and is a member of many IEEE technical committees.

Maria-Dolores Cano received the Telecommunications Engineering degree in 2000 from Universidad Politécnica de Valencia, Spain, and the Ph.D. degree from the Universidad Politécnica de Cartagena (UPCT), Spain, in 2004. She joined UPCT in 2000, where she is currently an Associate Professor in the Dept. Information Technologies and Communications. She has published numerous papers in international journals and conferences in the areas of traffic engineering, Quality of Service (QoS)/Quality of user Experience (QoE), security provisioning in telecommunications networks, and engineering education, which are also her research interests. Dr. Cano was awarded a Fulbright grant as a Postdoctoral Researcher in 2006 at Columbia University, New York, USA, and has also collaborated since 2007 with several universities in South America. She received the Best Paper Award at the 10th IEEE ISCC'05 and was given Exemplary Reviewer Recognition by IEEE Communications Letters in 2010. She is member of the Editorial Board of the IET Journal on Wireless Sensor Systems, as well as of the Technical Program Committee of several IEEE, IFIP, and IARIA international conferences. She has participated in several national and international research projects and also collaborates as a reviewer for international funding research programs and international journals included in the JCR.

References

[1] A. Ahmad, R. Arshad, S. A. Mahmud, G. M. Khan, and H. S. Al-Raweshidy, "Earliest-deadline-based scheduling to reduce urban traffic congestion," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1510–1526, 2014.

[2] S. Alabdallaoui, A. Berraissoul, and N. Idboufker, "Performance evaluation of RED approach for traffic lights management," in *Proc. 15th Int. Conf. Intelligent Systems Design and Applications*, 2015, pp. 358–362.

[3] S. Araghi, A. Khosravi, and D. Creighton, "A review on computational intelligence methods for controlling traffic signal timing," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1538–1550, 2015.

[4] M. C. Bell and R. D. Bretherton, "Ageing of fixed-time traffic signal plans," in *Proc. 2nd IEEE Int. Conf. Road Traffic Control*, 1986, pp. 77–80.

[5] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," IETF RFC 2475, 1998.

[6] T. R. Board, *Highway Capacity Manual: A Guide for Multimodal Mobility Analysis*, 6th ed. Washington, D.C., 2016.

[7] T. R. Board and N. R. Council, Highway Capacity Manual, 2000.

[8] M.-D. Cano, R. Sanchez-Iborra, F. Garcia-Sanchez, A.-J. Garcia-Sanchez, and J. Garcia-Haro, "Coordination and agreement among traffic signal controllers in urban areas," in *Proc. 7th Workshop on Communication in Transportation Systems and 18th Int. Conf. Transparent Optical Networks*, 2016, pp. 1–4.

[9] L. Chen and C. Englund, "Cooperative intersection management: A survey," *Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 570–586, Feb. 2016.

[10] D. Clark, R. Braden, and S. Shenker, "Integrated services in the internet architecture: An overview," IETF RFC 1633, 1994.

[11] S.-B. Cools, C. Gershenson, and B. D'Hooghe, "Self-organizing traffic lights: A realistic simulation," in *Advances in Applied Self-Organizing Systems*, M. Prokopenko, Ed. London, U.K.: Springer, 2013, pp. 45–55.

[12] D. Zhao, Y. Dai, and Z. Zhang, "Computational intelligence in urban traffic signal control: A survey," *IEEE Trans. Syst. Man, Cybern. C: Appl. Rev.*, vol. 42, no. 4, pp. 485–494, July 2012.

[13] P. Desai, S. W. Loke, A. Desai, and J. Singh, "Multi-agent based vehicular congestion management," in *Proc. IEEE Intelligent Vehicles Symp.*, 2011, pp. 1031–1036.

[14] J. A. F. F. Dias, J. J. P. C. Rodrigues, and L. Zhou, "Cooperation advances on vehicular communications: A survey," *Veh. Commun.*, vol. 1, no. 1, pp. 22–32, 2014.

[15] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *J. Artif. Intell. Res.*, vol. 31, no. 1, pp. 591–656, Jan. 2008.

[16] R. Florin and S. Olariu, "A survey of vehicular communications for traffic signal optimization," *Veh. Commun.*, vol. 2, no. 2, pp. 70–79, Apr. 2015.

[17] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Netw.*, vol. 1, no. 4, pp. 397–413, 1993.

[18] S. Floyd and K. Ramakrishnan, "A proposal to add explicit congestion notification (ECN) to IP," IETF RFC 2481, 1999.

[19] C. Gershenson, "Self-organizing traffic lights," *Complex Syst.*, vol. 16, no. 1, pp. 29–53, Nov. 2004.

[20] L. Guan, I. U. Awan, and M. E. Woodward, "Stochastic modelling of random early detection based congestion control mechanism for bursty and correlated traffic," *IEE Proc. Softw.*, vol. 151, no. 5, pp. 240, 2004.

[21] P. M. Khilar and S. K. Bhoi, "Vehicular communication: A survey," *IET Netw.*, vol. 3, no. 3, pp. 204–217, 2014.

[22] D. McKenney and T. White, "Distributed and adaptive traffic signal control within a realistic traffic simulation," *Eng. Appl. Artif. Intell.*, vol. 26, no. 1, pp. 574–583, Jan. 2013.

[23] "The road to 2020 and beyond: What's driving the global automotive industry," McKinsey & Company, 2013.

[24] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case," *IEEE/ACM Trans. Netw.*, vol. 1, no. 3, pp. 344–357, Jun. 1993.

[25] K. I. Park, *QoS in Packet Networks*, vol. 779. London, U.K.: Springer-Verlag, 2005.

[26] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource reservation protocol," IETF RFC 2205, 1997.

[27] N. Research, "Transportation forecast: Light duty vehicles," 2015.

[28] R. Sanchez-Iborra, J. F. Ingles-Romero, G. Domenech-Asensi, J. L. Moreno-Cegarra, and M.-D. Cano, "Proactive intelligent system for optimizing traffic signaling," in *Proc. 14th IEEE Int. Conf. Pervasive Intelligence and Computing*, 2016, pp. 544–551.

[29] D. Stiliadis and A. Varma, "Efficient fair queueing algorithms for packet-switched networks," *IEEE/ACM Trans. Netw.*, vol. 6, no. 2, pp. 175–185, Apr. 1998.

[30] K. Sue and C. Kao, "To go or not to go: n-hop congestion notice with experience support for dynamic detouring," *Adv. Comput. Sci. Int. J.*, vol. 4, no. 3, pp. 54–63, 2015.

[31] O. K. Tonguz and W. Viriyasitavat, "A self-organizing network approach to priority management at intersections," *IEEE Commun. Mag.*, vol. 54, no. 6, pp. 119–127, June 2016.

[32] R. Verma, A. Iyer, and A. Karandikar, "Active queue management using adaptive RED," *J. Commun. Netw.*, vol. 5, no. 3, pp. 275–281, Sept. 2003.

[33] L. Xiao, X. Peng, Z. Wang, B. Xu, and P. Hong, "Research on traffic monitoring network and its traffic flow forecast and congestion control model based on wireless sensor networks," in *Proc. Int. Conf. Measuring Technology and Mechatronics Automation*, 2009, vol. 1, pp. 142–147.

[34] Q. Yuan, Z. Liu, J. Li, J. Zhang, and F. Yang, "A traffic congestion detection and information dissemination scheme for urban expressways using vehicular networks," *Transp. Res. C: Emerg. Technol.*, vol. 47, pp. 114–127, Oct. 2014.

[35] "SUMO traffic simulator," 2016.