Passive, Decentralized, and fully Autonomous Intersection Access Control

Conference Paper · October 2014

DOI: 10.1109/ITSC.2014.6958176

2 authors:

John Khoury
Jacobs Engineering Group Inc

Joud Khoury
Raytheon Technologies

Passive, Decentralized, and fully Autonomous Intersection Access Control

John Khoury* and Joud Khoury†

***Abstract*— Current research on autonomous intersection management makes a set of assumptions including active Vehicle-to-Vehicle (V2V) or Vehicle-to-Infrastructure (V2I) communications, and/or centralized control. While they enhance the efficiency of the solution, such assumptions have inherent security and privacy drawbacks and require high infrastructure costs. This paper sets to investigate an alternative solution to autonomous intersection management that is decentralized (no centralized controller) and passive (no vehicle communications). Our scheme permits autonomous vehicles approaching an intersection to make localized collision-free access decisions based purely on sensing and beacon information. Besides demonstrating the feasibility of a fully autonomous and decentralized approach, we show that our scheme operationally outperforms a standard actuated signal and all-way stop control. Our decentralized approach trades off optimality for low cost, and enhanced security, privacy, and practicality.**

I. Introduction

More than 90% of all vehicle crashes are attributable to human error. The National Highway Traffic Safety Administration (NHTSA) highlights that intersection and/or intersection related crashes account for more than 50% of all traffic crashes [1]. Besides their safety issues, intersections are also a major source of urban congestion and delay. In its 2012 Urban Mobility Report, the Texas Transportation Institute estimated that congestion in 2011 cost Americans $121 billion in combined delay and fuel costs [2]. The trend is anticipated to keep increasing. As a result, interest in intelligent transportation systems (ITS) to mitigate the ever-growing congestion and safety issues has been increasing. The pursuit of improved vehicle safety and mobility has spurred the NHTSA to focus attention on crash-less autonomous vehicles [2]. The concept of autonomous vehicles is becoming a reality given the remarkable advances in vehicle technology, wireless communications, artificial intelligence, global positioning systems (GPS) and sensing. For example, the fully autonomous vehicle of Stanford won the DARPA Grand Challenge in 2005 [3], and the Google Driverless Car has successfully completed 300,000 miles of truly autonomous accident-free driving as of 2012.

With the advent of autonomous vehicles, recent work on intelligent intersections has demonstrated significant improvements in terms of efficiency and safety. Dresner and Stone [4], [5] present an intersection control protocol (and simulator) called Autonomous Intersection Management (AIM) that uses a centralized controller. Vehicles use reliable low-latency Vehicle-to-Intersection (V2I) communications to reserve space and time in the intersection by communicating with a centralized intersection manger. The latter controls the autonomous vehicles accepting or rejecting their requests to pass through the intersection. Zohdy and Rakha [6] similarly proposed a centralized controller and custom simulator optimizing the movements of vehicles to reduce the total delay for the entire intersection and prevent crashes. VanMiddlesworth *et al.* propose a decentralized peer-to-peer approach for small intersections which eliminates the central controller [7]. The approach relies on sensing and requires reliable low-latency Vehicle-to-Vehicle (V2V) communications. Recently, Carlino *et al.* proposed an auction based management scheme that improves fairness of intersection access. The auction scheme was simulated using an agent based microscopic simulator called AORTA (approximately orchestrated routing and transportation analyzer) [8].

So far, such research requires centralized controllers which reduces the autonomy of the overall solution and comes at a high cost especially in medium to small intersections. Additionally, the research assumes a secure, efficient, reliable, an low-latency V2I or V2V communication system is available and is capable of delivering the significant amount of real-time communications between the participants (vehicles and intersection manager). Building centralized controllers and securing the communication systems and the controllers at the intersections face several technical and economic challenges [9]. First, the dedicated Radio Frequency (RF) spectrum must be made available, and the RF spectrum should be allocated to the participants (vehicles and controllers) in real time and under both low and high contention scenarios. Second, the communications infrastructure must be secured[1] and user privacy must be protected. The latter is critical anytime V2I or V2V communications is proposed. In fact, experts anticipate that securing the communication systems is the bulk of the infrastructure cost [9], not to mention that current intersection infrastructure is already expensive.

This paper sets to investigate an alternative approach to autonomous intersection management which is *decentralized* (no centralized control) and *passive* (no V2V or V2I communications). We present the Decentralized Autonomous Intersection Access Control (DAIAC) scheme. DAIAC permits autonomous vehicles approaching an intersection to make localized access decisions based purely on sensing

*Dr. John El Khoury is with the Faculty of Civil Engineering, Lebanese American University, Byblos, Lebanon `jkhoury@lau.edu.lb`

† Dr. Joud Khoury is with the Advanced Networking Department, Raytheon BBN Technologies, Boston MA, USA `jkhoury@bbn.com`

[1] For example, communications are fundamentally vulnerable to jamming and other active attacks.

information. Vehicles sense contention through an emitted signal and individually decide on the next action: proceed or stop. The outcome allocation is collision-free. Operationally, our scheme outperforms a standard actuated signal but is less optimal than a centralized controller. While we tradeoff optimality, our approach is more secure and practical as it (1) eliminates the single point of failure associated with centralized controllers, (2) eliminates the communications system security vulnerabilities and privacy concerns, and (3) considerably reduces the vehicle and infrastructure costs.

We present the DAIAC scheme that aims to satisfy the following requirements:

- *Safety* collision-free access to the intersection
- *Efficiency* the objective is maximize the users' utilities and hence to minimize average delay over the traffic demand
- *Full Autonomy* vehicles make local *uncoordinated* decisions based on sensory information and need not communicate with each other or with the infrastructure (*passive*) i.e., the intersection is *unmanaged* and no V2V or V2I communication is needed
- *Decentralization* we eliminate any centralized controller needed and accordingly the single point of failure
- *Fairness* no lane has a preference over the other, and the queues are fair

The rest of the paper is organized as follows. Section II presents the intersection model and the DAIAC set-up. The DAIAC algorithm is detailed in section III. Section IV presents the evaluation metrics and results before presenting our conclusions and directions for future work in section V.

II. Intersection and Traffic Models

Our simplified intersection model is shown in Figure 1. Each lane has length D_L and width W_L. Each lane additionally has three non-overlapping zones: a *contention zone*, a *stop zone*,and a *safety zone*. Each zone is defined by two parameters: its distance from intersection, and its length. The *contention zone* has distance from intersection D_{cz} and length L_{cz}. The *stop zone* has distance from intersection D_{sz} and length L_{sz}, and the *safety zone* has distance from intersection D_{safe} and length L_{safe} where $D_{safe} = L_{safe}$.

The contention zone is intended to allow an incoming vehicle to determine if other vehicles from the other lanes are contending for the intersection. The stop zone is intended to allow a contending vehicle to safely stop if it determines that it has to. We shall explain how a vehicle makes that decision later in section III. The safety zone may be thought of as the safe stop distance before the intersection i.e., all vehicles that decide to stop must do so before entering the safety zone.

For the scope of this paper, we make the following simplifying assumptions:

- A single vehicle can be in the intersection at any time.
- A single lane in each direction amounting to a total of 4 lanes: eastbound, westbound, northbound, and southbound.

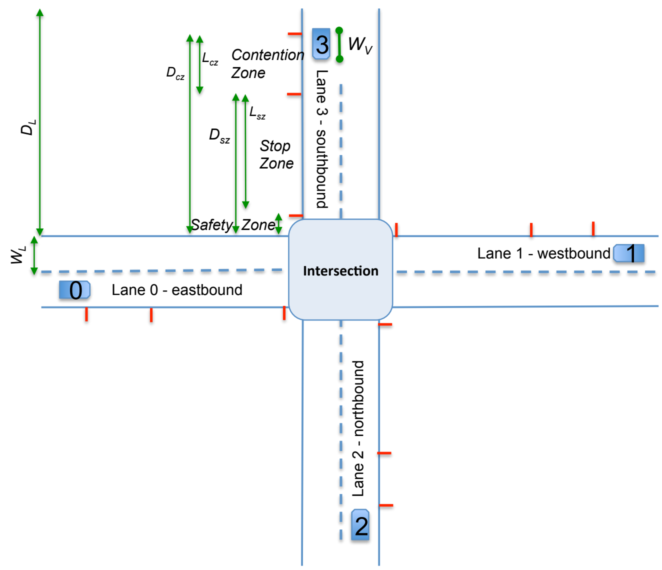


Fig. 1: Intersection Model

- Vehicles travel straight i.e. no turns are allowed. Note that our algorithms can support turns since we only allow a single vehicle in the intersection at any time. We only make this assumption to simplify the simulation.
- Vehicles in all 4 lanes travel at a constant speed of v m/sec, the lane's speed limit (being autonomous, it is logical to follow the speed limit).

We additionally assume that each autonomous vehicle is equipped with the following sensors:

- A Distance-to-Vehicle in front (D2V) sensor determines how far is the vehicle in front on the same lane (e.g., using laser rangefinder),
- A Distance-to-Intersection (D2I) sensor $dist2Intesection$ that determines how far the vehicle is from the intersection,
- A zone sensor that allows the vehicle to determine which zone it is currently in. Specifically, $inCtnZone$, $inStopZone$, $inIntersection$ return true if the vehicle is inside the contention zone, the stop zone, or the intersection respectively.

Finally, the intersection's infrastructure is equipped with presence sensors. Each lane has presence sensors that detect if vehicles are present within a certain zone. This information is made available to the vehicles and is used to detect contention (section III). The vehicles expect to receive the following information in every timestep:

- The current time,
- The lane ID that the vehicle is currently in,
- The per lane presence sensor values

There are different approaches to exposing this information to the vehicles. We describe two approaches: the first approach requires the intersection to expose this information while the second eliminates the intersection completely and relies on vehicle sensing.

In the first approach, the infrastructure sends a beacon, using short range Infrastructure-to-Vehicle (I2V) communication, to all vehicles located in any of the zones (contention, stop, safety) at every time step. The beacon contains all the information. This approach has the advantage of allowing both driverless and traditional driver-based vehicles to co-exist – the intersection can compute the signal and expose it. Vehicles use the beacon messages to synchronize their clocks. This clock synchronization technique is common to wireless networks, as described in the 802.11 standard [10] (see Time Synchronization Function (TSF) for Infrastructure Networks). Clock synchronization is important for consistent decision making as we shall see later in section III. It is important to highlight that the infrastructure in this case is still *stateless*. In other words, no state about any vehicle or sensor is required to be stored and/or processed centrally at the intersection for future computations. This renders the intersection infrastructure less costly and more resilient to failures.

Another approach does not require any infrastructure or communication (no beacon) to expose this information to the vehicles. Specifically, vehicles use the Global Positioning System (GPS) to synchronize their clocks. They can use imaging sensors to detect the lane ID and potentially other information which could be posted to the side of the road.

In terms of the vehicles, we use a single type of vehicle for our simulation. The vehicle has length $L_V = 4.5$ m and is 1.85 m in width, has a maximum acceleration of 3.5 m/sec^2 and maximum deceleration of 3.6 m/sec^2. The values are consistent with those used in standard traffic simulation tools.

Finally, we model vehicle arrival rates within a lane using a poisson random process with rate parameter λ vehicles/hour/lane (vphpl). In order to generate the poisson arrivals in our discrete event simulator, we split time into discrete time steps where each time step is τ sec. Then we further split each time step into k sub steps each of length τ/k sec. Within each sub time step, we generate a vehicle (an event) with uniform probability $p = \lambda\tau/k$. This results in poisson arrivals and allows generating up to k vehicles per time step [2]. The above process is repeated independently for all 4 lanes.

III. DAIAC Algorithms

Our DAIAC algorithms allow vehicles to make local uncoordinated decisions as to whether they can access the intersection based purely on the sensing and beacon information described in section II. Our distributed algorithms achieve three main tasks: (1) compute a consistent actuated signal (Algorithm 1), (2) determine whether there is contention (Algorithm 2), and (3) use the above to safely decide whether to stop or traverse the intersection (Algorithm 3) We shall detail the algorithms next. The algorithms presented hereafter are executed by each vehicle at every time step (i.e., whenever a vehicle receives a beacon).

[2]Note that an alternative approach is to compute the exact arrival time of the next vehicle using the exponential probability distribution function.

Recall that the infrastructure sends beacon messages to the vehicles every τ seconds where each beacon message contains the following information: a timestamp now indicating the current time in increasing ticks, the index (id) of the current lane $lane\_id$, the zone presence information per lane where $presence_s[lane\_id]$ is a bit indicating whether vehicles are present within the stop zone of the lane indexed by $lane\_id$, similarly $presence_c[lane\_id]$ indicates whether vehicles are present within the contention zone of $lane\_id$, a configured $green$ duration indicating the duration of a green interval, and similarly all_red duration indicating the duration of the all-red interval as described shortly.

First, we present a distributed algorithm for vehicles to consistently simulate an actuated signal. Algorithm 1 allows each vehicle to locally compute a signal phase for its current lane simulating an actuated signal with two phases: a GREEN phase assigns to a single lane a green signal for a fixed duration of $green$ sec (while assigning red to other lanes), and a RED phase assigns a red signal to all lanes for a duration of all_red seconds. Since all vehicles synchronize their clocks, the result is a consistent actuated signal phase i.e. the output of the algorithm at each time step is a signal phase (GREEN or RED) which is used by Algorithm 3 (described shortly) to determine whether to stop or proceed. The all_red phase can include a guard to protect against small clock offsets during the vehicle's ride (or wait) through the stop zone.

The algorithm is executed by each vehicle as soon as it enters the contention zone. This gives each vehicle enough time (at least L_{cz}/v) to synchronize its clock before entering the stop zone at which point the value of the signal will be used to determine whether to stop or not as explained shortly in Algorithm 3. Note that $lane[x]$ returns the lane index for a specific phase i.e., $lane[0]$ is westbound, $lane[1]$ is southbound, $lane[2]$ is eastbound, and $lane[3]$ is north-bound. Finally, it is important to note that our model could simultaneously allow for mixing driverless and traditional driver-based vehicles– in the latter case the infrastructure simply exposes the traffic light to the drivers.

Algorithm 2 next describes how a vehicle determines whether there is contention. Only while the vehicle is in the contention zone, it checks whether vehicles are present in the contention or stop zones of other lanes using $presense_c[lane\_id]$ and $presense_s[lane\_id]$ and if so sets the contention flag.

Based on this computed state, a simple local algorithm (Algorithm 3) is executed by a vehicle to determine whether to stop or to proceed. Note that $slowedDownBeforeIntersection$ returns true if at any point in the stop zone the brakes are applied.

IV. Evaluation

In order to evaluate our algorithms, we use the following two metrics:

- Average $delay$ is defined as $\frac{\sum_{i \in V}(T_i - T_i^*)}{|V|}$ where V is the set of all vehicles that completed the journey, T_i is the actual travel time of vehicle i and T_i^* is

Algorithm 1 Algorithm executed by a vehicle for simulating an actuated signal

```
function GETTRAFFICSIGNAL(now, lane_id, presence_s, green, all_red, τ)
    new_period = false
    period = green + all_red
    epoch = ⌊now / period⌋
    prev_epoch = ⌊(now − τ) / period⌋
    if epoch > prev_epoch then
        new_period = true
    end if
    phase_id = epoch%4                                ▷ 4 lanes
    if new_period and                                 ▷ actuated logic
       not presence_s[lane[phase_id]] then
        for i = 1 → 3 do
            next_id = (phase_id + i)%4
            if presence_s[lane[next_id]] then
                phase_id = next_id
                break
            end if
        end for
    end if
    if lane[phase_id] = lane_id then
        t = now − epoch * period
        if t ≤ green then
            return GREEN                    ▷ green signal for lane_id
        else
            return RED                      ▷ all-red signal, for all lanes
        end if
    else
        return RED                          ▷ red signal for other lanes
    end if
end function
```

Algorithm 2 Algorithm executed by a vehicle for deciding whether there is contention

```
var contending              ▷ this state is kept throughout the journey
function ISCONTENDING(lane_id, presence_c)
    if inCtnZone then
        for i = 0 → 3, i ≠ lane_id do
            if presence_c[i] or presence_s[i] then
                contending = true
            else
                contending = false
            end if
        end for
    else if inIntersection then
        contending = false                          ▷ clear it
    end if
    return contending
end function
```

Algorithm 3 DAIAC algorithm

```
if inStopZone then
    if (not ISCONTENDING() and
      not slowedDownBeforeIntersection)
      or not GETTRAFFICSIGNAL() == RED
      or TOOLATETOSTOP() then
        Proceed safely              ▷ don't hit vehicle in front
    end if
else if inIntersection then
    Traverse intersection safely
else
    stop        ▷ Proceed safely and stop before safety zone
end if

function TOOLATETOSTOP(v, a) ▷ v is vehicle's current velocity and a is its max deceleration
    dist2Stop = − v^2 / (2*a)
    if dist2tIntersection < dist2Stop then
        return true
    else
        return false
    end if
end function
```

the optimal travel time assuming the vehicle travels at constant speed of v m/sec and does not slow down (i.e., $T_i^* = D/v$ when the travel distance is D).

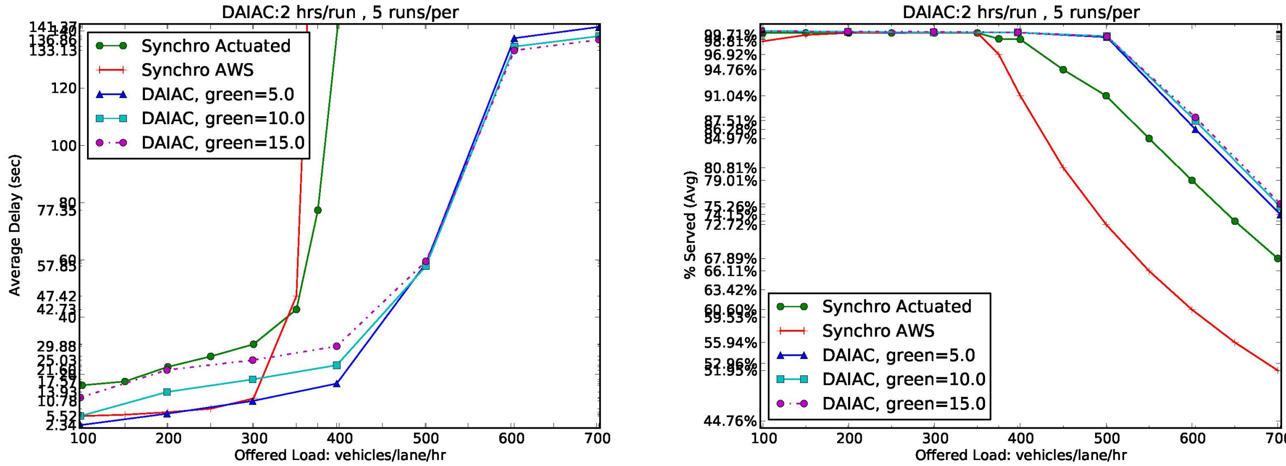
- Average *percent served* is defined as $\frac{\sum_{j=1}^{N} \frac{|C_j|}{|S_j|}}{N}$ where C_j is the number of vehicles that were served (completed) and S_j is the number of vehicles that were spawned in time period j. This is a sampled ratio of vehicles completed to vehicles spawned. The ratio/sample is computed every period (in our case 5min) and all samples are averaged.

We made significant extensions to the open source custom AIM simulator described in [5] to support decentralized operations. We used the simulator to measure the performance of our DAIAC algorithm. A sketch of the simulator in action is shown in Figure 2.

The following parameters were fixed during the simulation:

1) The time step $\tau = 0.02$ sec
2) $D_{cz} = 40.5$ m, $L_{cz} = 8$ m, $D_{sz} = 32.5$ m, $L_{sz} = 28$ m, $D_{safe} = 4.5$ m, $L_{safe} = 4.5$ m, $W_L = 4$ m
3) The lane speed limit is $v = 10$ m/s
4) The *all_red* safety phase was set to 1.94 sec
5) The *green* duration is increased from 5 to 15 sec

Different values for the *green* time were simulated corresponding with the increased demands per lane. Obviously, at higher demands, queues will form and increasing the *green* time would be more efficient. The *all_red* phase time was calculated to account for the worst case scenario. The worst case scenario is defined when two vehicles from consecutive lanes (perpendicular) are spawned with one time-step differ-



(a) Average Delay

(b) Percent Served

Fig. 3: Performance of DAIAC relative to actuated signal and All Way Stop (AWS). DAIAC was simulated using the custom AIM simulator [5] while the actuated and AWS were simulated using Synchro

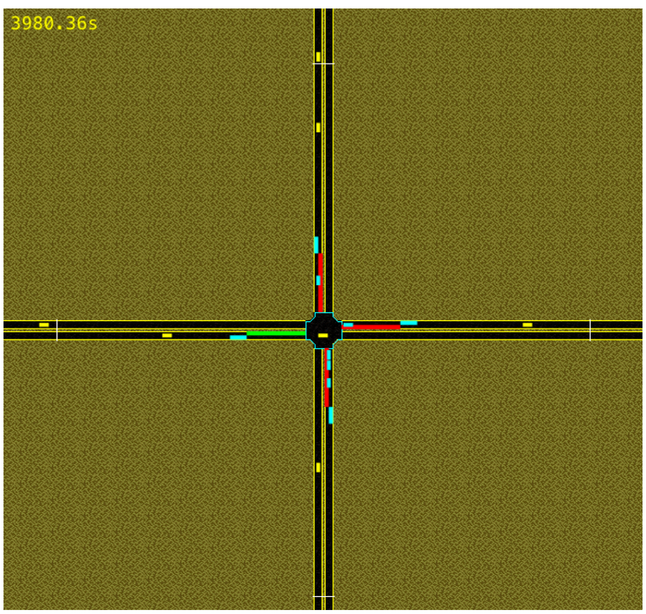


Fig. 2: Simulator Snapshot: vehicles in yellow are not contending (*contending* flag not set); vehicles in cyan are contending; the stop zone is colored to show the vehicles' view of the traffic signal (RED or GREEN); the contention zone is colored cyan when any of the active vehicles is contending.

ence in time. Since vehicles are autonomous and driving at constant speed, then the two vehicles will reach the point of decision (POD), within the stop zone, with one time-step difference in time. We define the POD as the point at which any vehicle will have to decide to stop or continue at constant speed, when contention is declared. It is located at a known distance from the safety zone, to allow for safe stopping. Once a vehicle passes the POD, the vehicle is bound to continue irrespective of the signal indication (similar to the dilemma zone in current traffic signal operations). Note that all vehicles that lose contention and are supposed to stop, will start decelerating at the POD. This will minimize vehicle delay and maximize the time vehicles are driving at the speed limit (v m/sec). Consider two vehicles generated with a time-step difference and reached the POD. Since they are in contention, one will get the green while the other will get the red phase. The worst case scenario occurs when the green for vehicle 1 turns red right when it passed the POD. Then, vehicle 1 is bound to continue at its speed limit. The all-red phase is in effect and vehicle 2 is decelerating at constant rate (d m/sec^2). Once the all-red is over, vehicle 2 gets the green phase and starts accelerating (also at constant acceleration rate of a m/sec^2) back to the speed limit. Assuming vehicle 1 is in a lane and vehicle 2 is in the next counterclockwise lane. Vehicle 1 will need to traverse 12.5 meters more than vehicle 2 to avoid a collision, which is the distance through the intersection ($2 \times W_L$) and its own length (L_V). Then, the all-red phase will need to delay vehicle 2 enough to create a gap between the two cars equal to the additional distance the first car has to traverse.

We define the following new parameters (all others are defined earlier):

- v_{min} = minimum speed vehicle 2 reaches at the end of the all-red (m/sec)
- v_c = speed of vehicle 2 at the collision point (m/sec)
- R = all_red time (sec)
- t_c = time from end of all-red phase to collision (sec)
- L_I = distance through the intersection ($2 \times W_L$) (m)

From the basics of vehicle motion (knowing constant acceleration and deceleration apply), we define the positions and speeds of vehicles' 1 and 2. The distances traversed by vehicles' 1 and 2 are presented in equations 1 and 2, respectively:

- $v(R+t_c) = [\frac{v^2}{2d} + L_V + L_{safe} + L_I]$......(1)
- $\frac{(v+v_{min})}{2} \times R + \frac{(v_{min}+v_c)}{2} \times t_c = [\frac{v^2}{2d} + L_{safe}]$......(2)

The motion of vehicle 2 can be explained using the following equations:

- $v_{min} = v - dR$......(3)
- $v_c = v_{min} + at_c$......(4)

Replacing equations 1, 3 and 4 in 2 and rearranging results in a second degree equation of R. Using the defined parameters, R was calculated to be 1.94 sec, which insures zero collision rate.

The simulation results of the proposed DAIAC scheme are presented in Figures 3a and 3b. Various demands are simulated ranging from 100 to 700 vphpl. The average delay per vehicle (seconds per vehicle) entering the intersection and the percent of vehicles served are the two main metrics measured at every time step. The two measures were used to compare the performance of the DAIAC scheme to a standard actuated signal controller and to an all-way stop (AWS) control. The standard actuated signal and the AWS are simulated using the well-known market software [11]. Two models (actuated signal control and an AWS) were created using Synchro to replicate the single-lane approaches to an isolated intersection shown in Figure 2. Multiple replications of the two models were created to account for the varying demands from 100 to 700 vphpl. Parameters used in the DAIAC and Synchro simulators were set to be consistent. Those parameters include geometric characteristics of the intersection, vehicle length and width, vehicle type, acceleration, deceleration, headway and speed factor. For example, only passenger vehicles were simulated in both models having characteristics as discussed in Section II.

The comparison differentiates between operations under low contention (approach demands less than 400 vphpl) and operations at higher demands. For demands between 100 and 300 vphpl, the DAIAC scheme performed, on average, 4.5 times better than the standard actuated signal (a reduction of 75% in delay) and 1.5 times better than the AWS (25% reduction in delay). For low contention conditions, the AWS, actuated signal and the DAIAC simulation are able to serve 100% of the demand.

As for the higher approach demands (starting at 400 vphpl), the DAIAC scheme performed 7 times better than the standard actuated signal (more than 85% reduction in delay) and 18 times better than the AWS (95% reduction in delay). In addition to significant reduction in delay, the DAIAC scheme out-served both the actuated signal and the AWS. For high contention scenarios, we see about 10% increase in service rate relative to the actuated signal, and about 30% increase relative to the AWS. Actually, the DAIAC scheme continued to serve 100% of the demand even as the flow reached 500 vphpl. The actuated signal and the AWS both reached capacity at demands around 350 vphpl.

V. Conclusion and Future Work

Throughout this paper, we presented a different approach to autonomous intersection management. The proposed DAIAC scheme permits autonomous vehicles approaching an intersection to make localized collision-free access decisions based purely on sensing and beacon information. An operational comparison between the DAIAC and the standard actuated signal controller has been conducted. Results show that the average delay per vehicle using the DAIAC scheme is 3 to 9 times lower than a standard signal operation (an average of 75% reduction in delay) and 1 to 20 times lower than an AWS. The results were consistent for all tested traffic demands per approach. The DAIAC scheme also proved much more flexible in serving the higher demand levels.

We acknowledge that a central automated controller outperforms the DAIAC scheme, but we traded-off optimality for security, privacy and practicality of the system. The system we present is passive requiring no V2V or V2I communications and is solely based on sensing information. With a decentralized approach, the single point of failure associated with centralized controllers is eliminated. In addition, the cost of the extensive communications infrastructure needed for centralized operations is significantly reduced.

We presented the initial step for the development of a fully decentralized and autonomous access scheme, which is practical and feasible. Future work will include expanding the simulation capabilities to account for turns at the intersection, a larger set of vehicle classes/characteristics, various intersection geometries and a wider range of approach demands, more than one vehicle per intersection, and support for multiple intersections.

References

[1] "Traffic safety facts 2012," National Center for Statistics and Analysis (NCSA), U.S. DOT Washington, DC, 2013.

[2] T. L. David Schrank, Bill Eisele, "Ttis 2012 urban mobility report," Texas A&M Transportation Institute, http://mobility.tamu.edu, December 2012.

[3] S. Thrun, M. Montemerlo, and et al., "Stanley: The robot that won the darpa grand challenge: Research articles," *J. Robot. Syst.*, vol. 23, no. 9, pp. 661–692, Sep. 2006.

[4] K. Dresner and P. Stone, "Multiagent traffic management: a reservation-based intersection control mechanism," in *Proceedings of AAMAS '04*, July 2004, pp. 530–537.

[5] ——, "Multiagent traffic management: An improved intersection control mechanism," Proceedings of AAMAS '05, pp. 471–477, July 2005.

[6] I. Zohdy and H. Rakha, "Optimizing driverless vehicles at intersections," In 10th ITS World Congress Vienna, Austria, October 2012.

[7] M. VanMiddlesworth, K. Dresner, and P. Stone, "Replacing the stop sign: Unmanaged intersection control for autonomous vehicles," in *Proceedings of AAMAS '08*, 2008, pp. 1413–1416.

[8] D. Carlino, S. D. Boyles, and P. Stone, "Auction-based autonomous intersection management," in *Proceedings of the 16th IEEE Intelligent Transportation Systems Conference (ITSC)*, 2013.

[9] US Government Accountability Office, "Vehicle-to-vehicle technologies expected to offer safety benefits, but a variety of deployment challenges exist," Intelligent Transportation Systems, [online] http://www.gao.gov/assets/660/658709.pdf, November 2013.

[10] IEEE Computer Society, "IEEE 802.11 standard part 11: Wireless lan medium access control (MAC) and physical layer (PHY) specifications," 2012.

[11] TrafficWare, "Synchro/simtraffic product," [online] http://www.trafficware.com.